

*Jean-Pierre Depétris*

# **Méthode raisonnée pour éditer en ligne**

et pour écrire avec un ordinateur

2009

## Historique :

Version 1.1, décembre 2009, corrections mineures.

Version 1.0, 13 novembre 2009

Première mise en ligne, octobre 2009

© Jean-Pierre Depetris, octobre 2009

Copyleft : cette œuvre est libre, vous pouvez la redistribuer et/ou la modifier selon les termes de la Licence Art Libre. Vous trouverez un exemplaire de cette Licence sur le site Copyleft Attitude <http://www.artlibre.org/> ainsi que sur d'autres sites.

## Table des matières

<b>Introduction</b> .....	7
<b>La page web</b> .....	11
Qu'est-ce qu'une page ?.....	11
Comment faire une page HTML ?.....	11
Texte et HTML.....	12
Qu'est-ce que le HTML ?.....	12
De quoi est composée une page HTML.....	13
Une propédeutique.....	14
<b>Le web</b> .....	15
Les trois internets.....	15
Le bac à sable.....	15
La nouvelle économie numérique.....	15
L'outil de travail.....	15
L'essence de l'internet.....	16
La nouvelle bibliothèque de Babel.....	16
Le changement.....	16
La difficulté de cerner le changement.....	17
L'architecture du net.....	18
Les protocoles du net.....	18
Protocoles de couche 2.....	19
Protocoles de couche 3.....	19
Protocoles de couches 3 et 4.....	20
Protocoles de couche 5 et plus.....	20
Les serveurs.....	20
Systèmes d'exploitation et logiciels.....	20
Principales utilisations d'un serveur.....	21
Serveur de fichiers.....	21
Serveur d'impression.....	21
Serveur de base de données.....	21
Serveur de courrier.....	21
Serveur web.....	22
Autres.....	22
Les hébergeurs.....	22
Les diverses offres d'hébergement.....	23
L'époque numérique et prénumérique.....	24
L'usage du net.....	24
Les limitations du net.....	24
La personne et le web.....	25
L'autre côté de l'écran.....	26
Principes de base sur les sites et le courriel.....	26
Le courriel.....	26
La complémentarité des sites et des courriels.....	27

Le papier et l'écran.....	28
Le document et le programme.....	28
L'affichage et l'impression.....	28
La typographie.....	29
Les conventions de mise en page.....	30
Proscrire les sauts de ligne.....	31
Utiliser les styles.....	31
<b>Écrire avec un ordinateur.....</b>	<b>33</b>
Le texte.....	33
Les jeux de caractères.....	33
L'US-ASCII.....	33
Jeu de caractères Macintosh.....	34
Jeu de caractères ISO-LATIN-1.....	35
Les jeux de caractères Unicode.....	36
Les formats.....	36
Le format DOC.....	37
Les formats RTF et RTFD.....	37
Les formats XML et ODT.....	38
Les formats TEX et LATEX.....	38
Les formats PDF, PS et EPS.....	39
Les formats PostScript et EPS.....	39
Le format PDF.....	39
Les formats HTML et XHTML.....	41
Le format TXT.....	42
Le SVG.....	42
Les systèmes et la notion de transparence.....	43
Window et Unix.....	43
Transparence et opacité.....	43
Les systèmes et les programmes.....	44
Mac OS.....	45
Les programmes.....	46
Programmes, formats et langages.....	46
Les outils.....	46
Le choix des programmes.....	47
La portabilité.....	47
L'ergonomie.....	47
Les fonctionnalités.....	48
La révolution de l'écrit.....	49
L'œuvre en procès.....	49
Concilier écriture édition et lecture.....	50
Éditabilité, lisibilité.....	50
Le marbre liquide.....	50
La réédition permanente.....	52
Écrire dans la durée.....	52
La puissance du texte.....	52

<b>Ce qu'est un site</b> .....	55
Comment crée-t-on un site ?.....	55
De quoi il est question.....	55
Deux méthodes.....	56
L'éditeur de site.....	56
L'exportation.....	58
Un cas concret.....	59
L'exportation.....	60
Le modèle de page.....	60
La création de la page.....	61
Structure.....	62
Le fichier du texte.....	65
La balise BODY.....	65
La CSS.....	67
La CSS externe.....	68
Une approche plus rigoureuse du même cas.....	69
Remarque sur le code source et les programmes.....	69
Remarques sur l'organisation des répertoires.....	69
Le formatage de la page avec des CSS.....	70
La navigation.....	73
Ce qu'il importe de retenir.....	74
La syntaxe des CSS.....	75
Les sélecteurs et leurs possibles attributs.....	75
Les unités de mesure.....	76
Les couleurs.....	77
La compacité du code.....	77
 <b>Écrire et éditer</b> .....	 79
Aperçu.....	79
Essai de synthèse.....	79
La communication facile.....	79
L'évolution du traitement numérique du texte.....	80
Quelques autres conclusions.....	81
Trois états.....	81
Du point de vue de l'écriture.....	82
Écrire et publier.....	83
Pourquoi un site.....	83
Les outils collaboratifs.....	84
L'usage.....	85
Le plus simple est le mieux.....	85
Le travail en cours.....	85
L'idée fallacieuse de perfection.....	86
L'édition en temps-réel.....	86
Le nettoyage du code.....	87
Évaluer le plus simple.....	88
La collaboration.....	88
Le courriel.....	88
Pièces-jointes et URL.....	88

La correspondance électronique reste de la correspondance.....	89
Les envois multiples.....	89
Les outils de coopération.....	90
Liste de diffusion et forum.....	90
Le blog.....	90
Le wiki.....	91
Le système de gestion de contenu ou CMS.....	91
Le dossier de travail.....	92
Le site.....	93
Comment ne pas rendre aisément accessible tout ce qu'on publie en ligne.....	93
Pourquoi ne pas rendre aisément accessible tout ce qu'on publie en ligne.....	93
Comment rendre intelligemment son site visible.....	93
Les liens.....	94
Qu'est-ce, fondamentalement, qu'un lien ?.....	94
Les liens et l'écriture.....	95
Les liens et la sauvegarde.....	95
Sur la propriété des auteurs.....	95
Conclusions provisoires.....	96
<b>Bibliographie.....</b>	<b>99</b>

# Introduction

Je rencontre rarement des utilisateurs personnels ou des sociétés qui ne sous-emploient pas leurs machines et leurs programmes, tout particulièrement pour se contenter de faire de la « bureautique », et ce terme en dit déjà assez de ce sous-emploi. Je découvre tout aussi rarement des sites ou des blogs faits simplement et intelligemment ; de même je reçois très peu de courriels écrits et adressés correctement et qui ne soient pas accompagnés de pièces jointes trop lourdes et dans des formats fantaisistes. Je me suis parfois demandé comment on pouvait apporter des solutions à ces trois problèmes, et j'ai fini par conclure qu'on ne pouvait répondre qu'aux trois en même temps.

Et d'abord, y a-t-il bien problème ? Sans-doute, puisqu'il y a des solutions. La nature des solutions proposées, et leurs relations avec les problèmes me laisse toutefois songeur. Depuis plus de vingt ans que j'utilise un ordinateur pour écrire, j'ai rencontré beaucoup de problèmes, et j'ai trouvé beaucoup de solutions. Je suis aussi toujours surpris de lire ou d'entendre qu'il y en aurait d'accessibles sans peine ni sans rien connaître. Il me semble plutôt que ce sont ces solutions pour réaliser certaines choses sans peine ni sans connaissances particulières qui génèrent le plus de problèmes.

Je ne crois pas qu'il soit possible de faire quoi que ce soit avec des programmes sans *se casser la tête*, ni même, pour être précis, sans éprouver ce plaisir particulier qui accompagne la sensation de ne pas s'être *cassé la tête* pour rien. Cela tient à la nature de la programmation. Si l'on dit à quelqu'un de sortir, on n'a généralement pas à lui commander d'avancer d'abord un pied devant l'autre dans la direction de la porte, puis de recommencer en déplaçant le centre de gravité de son corps vers l'avant jusqu'à ce qu'il soit assez près pour atteindre de sa main la poignée sur laquelle il exerce une pression vers le bas, etc. C'est ce qu'on doit faire en programmant.

Rien n'est plus bête en réalité que *l'intelligence artificielle*. Un programme exécute scrupuleusement ce qu'on lui commande sans aucune initiative. Cela impose de lui spécifier très exactement tout ce qu'il doit faire, et par conséquent de le savoir soi-même. L'avantage est qu'on n'a plus à le lui redire ; il ne l'oublie plus. Le désavantage est qu'on tend alors soi-même à l'oublier.

D'accord, la plupart du temps, nous ne programmons rien, nous ne sommes pour rien dans les programmes qui s'exécutent. C'est donc que quelqu'un d'autre les aura programmés à notre place. Inévitablement, un autre aura dû le faire pour nous. C'est encore un avantage de la programmation : ce que quelqu'un peut faire, il peut le faire une fois pour toutes pour beaucoup de monde.

Le désavantage est que le programme qu'un autre a écrit pour nous, n'exécutera pas toujours exactement ce qu'on aurait souhaité lui voir faire. Plus un programmeur aura tout prévu pour nous permettre de travailler sans peine ni sans rien connaître, plus il y aura de chances pour que son programme ne corresponde pas en tous points à ce que nous souhaitons, et plus nous aurons de peine et de choses à connaître, soit pour modifier le programme, soit, plus probablement, pour modifier les résultats. Dans tous les cas, nous ne devons pas seulement savoir ce que nous voulons, mais être capable de concevoir comment y parvenir par le menu détail.

Autant dire qu'il y a peu d'espoir de voir un programme faire à notre place ce que nous ne savons pas faire sans lui ; mais ce que nous savons faire, il pourra l'exécuter très rapidement et en nous épargnant toutes les tâches répétitives.

Il y a dans tout cela, si l'on y songe bien, quelque chose de contradictoire, et peut-être même de tragique. Il y a même quelque chose qui va à l'encontre de ce que l'homme depuis la nuit des temps tenait pour acquis.

Quoi donc ? — Que l'apprentissage consistait à cultiver en soi des automatismes, c'est-à-dire à chasser tous les menus détails hors de la conscience pour, en quelque sorte, les incorporer, alors que la programmation consiste bien au contraire à inscrire ces automatismes hors de soi, c'est-à-dire à éclairer d'abord ces menus détails par la conscience la plus attentive, et cela pour aussi vite les vider de sa mémoire.

Ce sont donc de toutes nouvelles aptitudes que nous serions appelés à cultiver. Nous devrions développer l'aptitude à passer le plus rapidement possible d'une vue analytique à une vision intuitive. Nous devrions apprendre à nous arrêter et à considérer le détail des opérations que nous nous efforcions avant de faire automatiquement à force de répétitions, et apprendre aussi à les oublier au plus vite pour n'en conserver qu'une intuition synthétique. En somme, nous devons avant tout cultiver notre principale aptitude : celle d'acquérir des aptitudes.

Tout cela n'est peut-être pas si nouveau qu'il pourrait y paraître. La programmation n'est jamais qu'un pas supplémentaire dans l'usage du signe écrit ; un pas peut-être décisif — on verra bien — mais un pas dans un long parcours. Le signe écrit permet d'oublier « deux fois deux » quand on a obtenu quatre. Ce n'est pas d'hier qu'on a appris à associer des opérations à des dispositifs matériels, comme l'antique boulier. Ce n'est donc pas non plus d'hier que le signe écrit est la véritable interface entre l'esprit et la mécanique, ce qui les sépare, comme le jaune et le blanc de l'œuf, pour permettre à l'un de commander l'autre.

Voilà un impressionnant détour pour aborder mon propos et le mettre en perspective. Trop souvent, les réponses à la question « comment » supposent que celles à la question « pourquoi » soient déjà connues. D'autre part, les réponses à la question « pourquoi » sont souvent incompréhensibles, inconcevables, ou dans le meilleur des cas stériles, sans celles à la question « comment ». À tout prendre, nous sommes bien capables de répondre seuls à « pourquoi » quand nous savons « comment ». Mais nous avons alors souvent besoin de savoir « comment quoi ».

Bref, ce que je tente d'expliquer dans les pages qui suivent n'est pas facile. Disons que j'ai cherché à combler un vide entre, d'une part, la quantité d'informations techniques et ciblées qu'on trouve sans peine (mais où l'on peine quand même à s'y retrouver), et, d'autre part, des connaissances plus théoriques, mais qui finissent par se confondre avec des généralités stériles, voire du discours militant et publicitaire.

Bien souvent, l'ordinateur semble être moins un outil pour simplifier la vie, accroître la capacité de concentration ou d'intuition, l'autonomie et l'inventivité... que l'inverse : une source de soucis et de contraintes qui conduit plutôt à la dispersion de l'esprit. C'est à quoi il peut bien conduire si l'on n'en possède pas assez les techniques. C'est à quoi il est même probable qu'il nous contraigne au moins momentanément, pendant le temps nécessaire à les acquérir. Comme ces techniques sont en perpétuelles mutations, il importe aussi d'apprendre à les acquérir de manière à ne pas se retrouver en situation d'éternel débutant.

Y parvenir suppose bien sûr de faire une part du feu, et d'accepter des efforts momentanés de mise-à-jour de nos connaissances, de nos logiciels et de notre matériel, et des efforts d'entretien continu. Il n'y a là rien de bien nouveau ni de bien particulier à l'informatique. Ce qui l'est davantage, c'est de s'entendre perpétuellement affirmer qu'on pourrait parvenir au même résultat sans peine ni sans rien connaître.

En local ou en réseau, tout se fait avec des programmes. Il n'est en effet pas très difficile d'apprendre à les utiliser. On trouve sans peine des méthodes et des tutoriels, mais ils n'apprennent

guère autre chose qu'utiliser un et un seul programme. Or, toute la difficulté est là : tout ce qu'on fait à l'aide d'un programme doit tôt ou tard passer par un autre, ou du moins s'en émanciper. En particulier, tout ce qu'on publie en ligne à l'aide de n'importe quel logiciel est déjà fait à l'aide d'un ou de nombreux autres.

Par conséquent, ce travail ne prétend pas se substituer aux manuels et aux tutoriels des différents programmes. Il suppose au contraire une connaissance minimale des logiciels qu'on utilise. Il se contente d'ouvrir des pistes pour en tirer davantage, ou éventuellement, changer d'outils.

À quels utilisateurs s'adresse-t-il ? En principe, à tous. Le complet débutant fera mieux de se familiariser d'abord avec les arborescences de ses disques durs, avant de se lancer à enregistrer des scripts, mais il n'aura pas à attendre de devenir un expert. L'expert lui-même trouvera certainement matière à s'instruire sur des questions qu'il ignore ou néglige. J'en ignore et néglige moi-même, à commencer par éditer des équations et des formules en ligne — sujet que j'ai préféré éliminer carrément.

Cet ouvrage s'adresse surtout à des utilisateurs qui veulent cultiver leur autonomie. Peu importe que leurs objectifs soient professionnels ou ludiques, ambitieux ou modestes, collectifs ou privés. On a souvent confondu l'autonomie et l'autarcie, et pour les opposer à la coopération et à la dépendance, que l'on confond aussi. (Si « libre » veut la plupart du temps dire « gratuit », « coopératif » ne signifie souvent rien d'autre que « captif ».) En réalité, l'autonomie est nécessaire à la coopération ; elle est même la condition *sine qua non* de la libre coopération.



# La page web

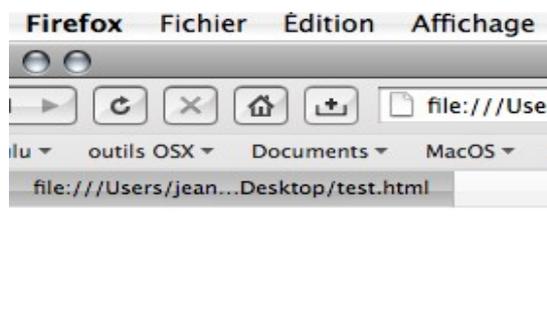
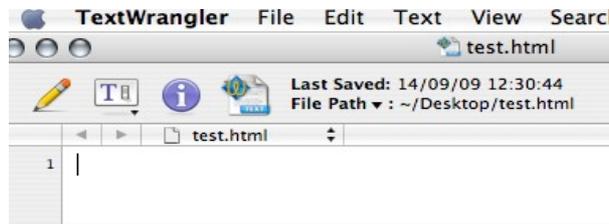
Contrairement à une idée reçue, il n'est pas plus facile de faire un site avec un logiciel qui permet d'ignorer complètement le code, que d'apprendre à manipuler ce code. Il vaut mieux accepter immédiatement l'idée qu'il est impossible de faire quoi que ce soit en ligne avec les langages et les codes standards du web sans y rien connaître.

## ***QU'EST-CE QU'UNE PAGE ?***

### ***Comment faire une page HTML ?***

Avant tout, on doit comprendre ce qu'est un site, et donc d'abord ce qu'est une page web, puisqu'un site est un ensemble de pages liées dans un même répertoire.

Une page web, ce n'est, pour ainsi dire rien. Il suffit de créer un fichier texte avec le suffixe « .html » ou « .htm », et l'on obtient une page web blanche.



Que peut-on mettre sur une page blanche ? Le plus simple est d'y mettre du texte. Sinon on peut y mettre à peu près n'importe quoi, des images, des tableaux, du son, de la vidéo, des programmes... On comprend aisément alors que ces différents objets ne se font pas avec un éditeur de page web, mais avec des logiciels spécialisés : traitement de texte, éditeur d'image, etc.

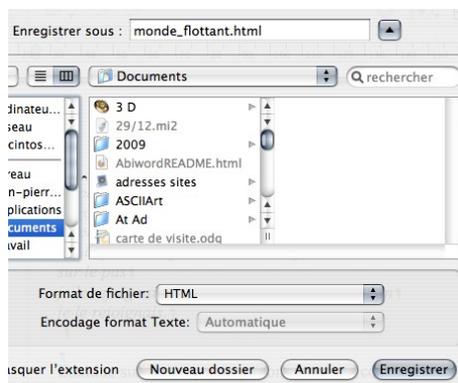
À supposer que l'on sache se servir de tels logiciels, la difficulté consistera à faire passer ce qu'ils nous ont permis de produire au sein d'une page web, avec les langages et le code qui sont les

standards du net. La difficulté consiste à faire passer son travail d'un logiciel à l'autre, et de le continuer dans d'autres formats.

### Texte et HTML

Le texte a cependant une nature différente des autres contenus possibles d'une page web : la page web est déjà du texte. Le format HTML (*Hyper Text Marked up Language*) est lui-même un format de texte. La meilleure et la plus simple façon de créer une page HTML est alors d'exporter son contenu à partir de son traitement de texte dans le format HTML.

La procédure est très simple dans son principe. On construit son fichier dans son traitement de texte, on le corrige, on le retravaille, on le met en page, on le structure, on y insère éventuellement les images, les tableaux, les sons, les liens externes, les couleurs de fond, etc. et l'on cherche dans les menus (généralement Fichier) celui qui correspond à la commande « Enregistrer sous » ou « Exporter », et l'on choisit dans le menu déroulant ce qui ressemble à « HTML ».



### Qu'est-ce que le HTML ?

Quand on lit du HTML dans un navigateur, ou encore dans un traitement de texte, puisque presque tous savent lire ce format, le texte s'affiche normalement. Quand on l'ouvre sur un éditeur de texte, on en voit le code.

Voici ce qu'on peut lire d'un passage de cette page lorsqu'on l'affiche en HTML :

```

37 faire quoi que ce soit en ligne avec les langages et les codes
38 standards du web sans y rien connaître.
39 <H2 CLASS="western">Qu'est-ce qu'une page&nbsp;?</H2>
40 <H3 CLASS="western">Comment faire une page HTML ?</H3>
41 <P>Avant tout, on doit comprendre ce qu'est un site, et donc d'abord
42 ce qu'est une page web, puisqu'un site est un ensemble de pages liées;es
43 dans un répertoire.</P>
44 <P>Une page web, ce n'est, pour ainsi dire rien. Il suffit de créer
45 un fichier texte avec le suffixe "html" ou ".htm",
46 et l'on obtient une page web blanche.</P>
47 <P><FONT COLOR="#000000"><IMG SRC="Method_html_474844ae.png" NAME="images2"
ALIGN=LEFT VSPACE=24 WIDTH=283 HEIGHT=162 BORDER=1><BR CLEAR=LEFT></FONT><FONT
COLOR="#000000"><IMG SRC="Method_html_6c70d617.png" NAME="images1" ALIGN=LEFT
VSPACE=24 WIDTH=315 HEIGHT=112 BORDER=1><BR CLEAR=LEFT></FONT><BR>

```

Une part du texte s'affiche pour être lue par un être humain, une autre part n'est lue que par le programme. Un éditeur de texte sert à lire cette seconde part afin de la modifier. Ne pas confondre un éditeur de texte avec un traitement de texte.

La qualité du code qu'exporte un traitement de texte est essentielle. D'elle dépend la facilité avec laquelle il est possible le modifier, le corriger, l'améliorer. En dépend aussi d'abord la plus ou moins grande nécessité de le retoucher. Il est surtout important que le code soit clair et comporte le moins possible de balises inutiles. Le code ci-dessus est relativement bon. Les balises de paragraphe `<P></P>` sont on ne peut plus simples, mais une part du code est inutile, comme les paires de balises `<FONT COLOR="#000000"></FONT>`.

### De quoi est composée une page HTML

Le texte HTML se divise donc en une part lisible par l'homme, et d'une autre lisible par le programme, et seulement visible à l'aide d'un éditeur de texte. La page elle-même se divise en deux parties, chacune enfermée dans sa paire de balises `<head></head>` et `<body></body>`. Les deux sont enfermées dans la paire de balises `<html></html>`.

La première ligne, le "doctype", indique qu'il s'agit d'une page HTML et précise sa version.

Voilà comment est structurée une page vide dès qu'on ouvre un nouveau fichier dans un éditeur HTML :

```

1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
  TRANSITIONAL//EN">
2 <html>
3
4 <head>
5 <title></title>
6 </head>
7 <body>
8
9
10 </body>
11 </html>

```

Entre les balises `<title></title>`, on inscrit le titre qui doit s'afficher dans la barre du navigateur. Elles se trouvent dans le HEAD et concernent le programme. Entre les balises `<body></body>`, on trouve tout ce qui s'affiche dans la page du navigateur, mais avec encore du code qui demeurera invisible.

<pre> 13 sources toutes distinctes.&lt;/P&gt; 14 &lt;P CLASS="western"&gt;Si vous regardez quoi&amp;nbsp;? Allez-vous me 15 demander. Si vous regardez ce qu'il y a de plus r&amp;eacute;el   dans la 16 r&amp;eacute;alit&amp;eacute;:&amp;nbsp;: vos impressions.&lt;/P&gt; 17 &lt;P CLASS="western"&gt;Tout le monde a fait l'exp&amp;eacute;rience que </pre>	<pre> distinctes.   Si vous regardez quoi ? Allez-vous me demander. Si vous   regardez ce qu'il y a de plus réel dans la réalité : vos impressions.   Tout le monde a fait l'expérience que des souvenirs qui sont   pourtant lointains peuvent paraître proches comme si l'on y était.   Je me souviens moi-même comme si j'y étais du goût des fruits   verts que je cueillis sur le poivrier près du bassin dans le jardin </pre>
---	--

On voit ci-dessus, côte-à-côte, le texte tel qu'il s'affiche dans un éditeur de texte et dans un navigateur.

Cette autre copie d'écran qui colorise en bleu tout le code met en évidence, d'une part les balises enfermées dans une paire de chevrons (<P>), et les caractères spéciaux, comme "&eacute;" qui se lit "é", ou "&nbsp;" qui est une espace insécable.

```

1034 <P><I>&laquo;&nbsp;La mer capture le dernier train, et les derniers-
1035 pr&eacute;sages en partance se blottissant les uns contre les autres-
1036 dans le fusible et dans la plume d'une dentelli&egrave;re presque-
1037 transparente qui ne r&eacute;v&egrave;le jamais sa peur ou son-
1038 plaisir, dans le ghetto difforme des remords et de l&rsquo;&eacute;rotisme-
1039 sans foi&hellip; Il y a des bijoux dans la chair, et c&rsquo;est le-
1040 cr&eacute;puscule dans ses crocs, et toi seul<!-- Masculin jusqu'à la fin du
1041 chapitre -->

```

On peut remarquer aussi la balise en gris <!-- Texte -->, qui permet d'entrer des notes personnelles ne concernant ni le programme ni le lecteur.

## ***UNE PROPÉDEUTIQUE***

Voilà succinctement ce qu'est une page web et comment on la produit. Ce n'est pas si difficile en principe, mais ce n'est quand même pas si simple au point qu'on puisse y parvenir sans ne rien savoir ni apprendre. Avant de mieux découvrir comment s'y prendre, il est nécessaire de connaître quelques autres aspects de la question.

- (i) avoir une idée assez claire de ce qu'est l'internet ;
- (ii) savoir produire du texte, c'est-à-dire écrire avec un ordinateur ;
- (iii) comprendre ce qu'est un site, comment il est structuré entre les pages qui le composent, leurs liens entre elles, entre les différents objets qu'elles contiennent, internes ou externes, et avec le reste de l'internet.
- (iv) comprendre comment tout ceci fonctionne ensemble.

# Le web

## ***LES TROIS INTERNETS***

On peut distinguer trois façons d'utiliser l'internet, qui produisent quasiment trois internets différents, ou du moins qui donnent l'apparence de trois niveaux, trois étages.

### ***Le bac à sable***

Le premier est plutôt ludique et consommateur. C'est sur quoi l'on tombe dès qu'on déballe une machine neuve : musiques en ligne, grands médias, offres extraordinaires, abonnez-vous, faites-vous des amis, tout est libre et gratuit, carte visa, et autres sornettes. Voilà ce que l'on trouve en poussant la porte, dans un style *salut des copains* des années soixante.

Si l'on entre par là, on peut passer sa vie à ne jamais en sortir. Il semble que ce soit d'abord de cet étage que se préoccupent les différentes lois récentes sur l'internet, et leur contestation aussi par ailleurs (« marre de payer », etc.)

Cette description est plutôt négative, mais on peut le voir aussi comme une sorte de bac-à-sable où l'utilisateur se dégrossit et s'amuse. C'est comme tout, il suffit de savoir passer à autre chose. Naturellement, en voyant cette grande nasse où règne le paiement sécurisé en ligne, chacun finit par être tenté de vouloir y pêcher un peu, ce qui nous conduit au deuxième niveau.

### ***La nouvelle économie numérique***

Ce second niveau est celui où l'on peut tirer un peu d'argent à entretenir le premier. En dehors de grandes structures, il concerne des utilisateurs déjà expérimentés, de petites entreprises ou des associations. Il est celui où l'on cherche à tirer quelques revenus de toute cette agitation *néocommunicationnelle*.

C'est le niveau que l'on appelle *la nouvelle économie numérique*. C'est en réalité la même économie qui ne change pas beaucoup : version améliorée de la vente par correspondance, ou vente d'équipements et de services propres aux nouvelles technologies. Sinon, le numérique et l'internet ne sont pas propices à un commerce spécifique, ni à l'échange marchand, principalement parce qu'ils favorisent la relation et l'échange directs.

### ***L'outil de travail***

Et puis il y a le troisième niveau, celui de l'outil de travail, de production. Le net est l'outil par excellence, le super-outil. On ne fait rien sans. C'est celui d'un outil de travail très efficace, mais c'est un outil de collaboration, pas d'échange, ni même de don.

C'est l'étage du web toujours égal à lui-même, et filtré par les deux premiers niveaux. C'est pour fonctionner ainsi que le web a été conçu, c'est donc le principe fondamental de son fonctionnement, et celui qui domine tous les autres.

La proportion des usages, les quantités d'internautes concernés, ne sont ici d'aucune importance. Elles ne sont pas déterminantes. Elles indiquent seulement le succès d'une nouvelle invention à faire évoluer les usages, ou le retard d'une population pour l'intégrer. Elles ne changent rien au fond à ce qu'est ce troisième niveau, et qui demeure par sa nature dominant.

## ***L'ESSENCE DE L'INTERNET***

Ces trois usages de l'internet ne devraient en principe pas se gêner et, du moins, personne ne devrait avoir à se gêner pour passer de l'un à l'autre. Il est cependant vrai que les deux premiers niveaux entraînent souvent des gênes importantes dans le troisième, et que les lois viennent encore renforcer.

Mon découpage est cependant utile pour concevoir sérieusement ce que l'on cherche à faire. Si l'on cherche seulement à se distraire, à se faire des amis, ou à échanger quelques photos, il n'est pas nécessaire de se casser la tête. Il ne manque pas de site pour offrir des blogs clé en main et qui ne demandent aucune connaissance préalable. On trouve aussi des sites dits « sociaux » où l'on peut héberger tout les documents en ligne que l'on souhaite. Si l'on cherche à vendre ou à tirer quelques profits de l'internet, on aura intérêt à acquérir d'autres techniques dont je ne parlerai pas. Ce qui suit concerne donc essentiellement le troisième étage.

### ***La nouvelle bibliothèque de Babel***

On ne doit quand même pas prendre trop au sérieux ce troisième étage, ni la séparation que je propose. Dès qu'on commence à la percevoir et à se faire une idée de la topique que j'ai dessinée, il vaut mieux s'empresse de la remettre en cause en brouillant les cartes. Les cartes ne cessent d'être battues, et c'est la nature même de l'internet de battre son jeu.

On pourrait comparer l'internet à une immense bibliothèque. Elle est si immense qu'elle n'a plus alors les caractères communs aux autres bibliothèques. Dans une bibliothèque normale, lorsqu'un livre n'est plus à sa place, il est perdu. Sur l'internet, aucun livre n'a de place, et rien n'y est jamais perdu. Rien n'est jamais très loin d'être sous la main, sous le doigt, même ce qui échappe aux moteurs de recherche.

Une bibliothèque suppose d'être classée, rangée, ordonnée, c'est-à-dire hiérarchisée. La base de donnée d'une bibliothèque peut avoir de nombreuses entrées, mais leur nombre est par nature limité. L'internet n'a besoin ni d'ordre ni de hiérarchie, et sa base de données est illimitée. Elle est composée non seulement de tous ses contenus, mais aussi de toutes ses combinaisons de contenus.

Un des plus célèbres logiciels de FTP s'appelle Interarchie, après avoir porté le non d'Anarchie. L'évolution du nom est instructive. L'internet est bien moins anarchique qu'interarchique.

### ***Le changement***

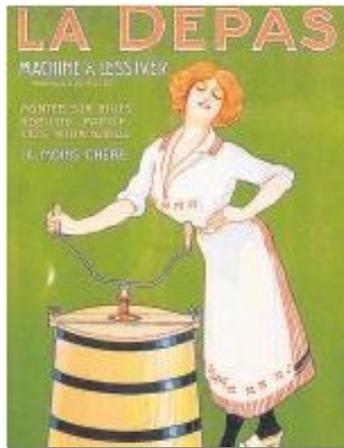
L'internet, et avec lui les deux techniques sur lesquelles il repose — la numérisation des données et l'ordinateur personnel, qui s'enchâssent les unes dans les autres comme des poupées russes — sont des inventions toutes nouvelles et qui n'ont pas encore modifié le monde où elles sont nées.

Comme toutes les nouvelles technologies qui réussissent, elles doivent d'abord prouver leur capacité de servir les usages anciens avant d'imposer les leurs.



Les premières publicités d'électroménager montraient des domestiques ravis devant une machine à laver ou un aspirateur. Le succès de l'électroménager a montré qu'il ne servait pas à rendre la vie plus facile au personnel domestique, mais à le rendre inutile et le faire disparaître. De la même façon, l'internet se présente souvent aujourd'hui comme s'il était au service de ce qu'il rend inutile.

Par exemple, l'internet se présente comme un média de masse, alors qu'il est à peu près aux masses ce que l'électroménager a été au personnel domestique.



### ***La difficulté de cerner le changement***

Le web apparaît comme ce qu'il est en principe : un outil permettant à chacun de s'adresser à tout le monde. Cependant, en mettant ce « tout le monde » à portée de clic, il en dissipe l'illusion. Il fait naître au contraire la prégnante question que les médias sur-développés du vingtième siècle avaient tendance à éluder : qui s'adresse à qui ?

Cette possibilité virtuelle de s'adresse au monde entier — qui pouvait d'ailleurs saisir avant n'importe quel journaliste, professeur, prêcheur, orateur ou auteur — renvoie immédiatement, quand on est sur le net, à la réalité de la relation actuelle, où l'interlocuteur peut se manifester immédiatement, sur le site, le blog ou le forum, ou par courriel, et laisser sinon au moins une trace sur le serveur.

Ce croisement inattendu entre la publication et le téléphone met à mal cette division qui paraissait aller tellement de soi entre public et privé. Public et privé apparaissent alors comme deux horizons dont l'internet occupe l'entre-deux. Bien d'autres séparations tombent avec celle-ci : célèbre et anonyme, producteur et consommateur, privé et public, central et marginal, etc.

Comme bien d'autres inventions antérieures, l'internet sert et sape à la fois les mœurs qui existaient avant elles. Ceci, la sociologie contribue plus à nous le cacher qu'à nous l'apprendre, pour la principale raison qu'elle part des mœurs pour étudier comment de nouvelles techniques les servent, et qu'elle ne saurait partir d'ailleurs, car là se trouvent ses outils conceptuels.

La sociologie va d'abord observer des comportements (humains) plutôt que des fonctionnements (techniques), puis les mesurer, et pour cela les échantillonner. Or il est bien difficile d'échantillonner une entité qui n'a ni limites ni centre. Quelle que soit la façon dont on s'y prenne pour étudier des comportements sur le net, on risque bien de n'avoir à faire qu'à ses propres comportements, usages et conceptions.

Les études sociologiques sont indigentes, et à leur suite les initiatives hasardeuses des législateurs ou des entrepreneurs qui s'appuient sur elles. Les limites entre le discours publicitaire et celui à prétention scientifique tendent aussi à se confondre et s'embrouiller. Les publicitaires s'appuient logiquement sur ces derniers, mais ceux-ci prennent trop souvent pour argent comptant les slogans qui leur sont renvoyés, confondant technologies et arguments de vente, et à leur suite, formateurs, enseignants et presse spécialisée.

Les propos avisés sur l'internet demeurent essentiellement du côté des ingénieurs. L'internet a son fonctionnement et sa logique. On peut, et l'on doit bien sûr, s'en servir à ce qu'on vise. On n'y parvient pas si on les ignore.

## ***L'ARCHITECTURE DU NET***

Une autre difficulté pour comprendre l'internet est qu'il ouvre d'un côté sur des questions proprement philosophiques (bien plus que sociologiques), des questions de philosophie des langages, des techniques, des mathématiques..., et de l'autre, sur des connaissances abscondes, avec une profusion de sigles sibyllins, désignant pêle-mêle des langages, des protocoles, des programmes, des appareils, des marques... comme s'il s'agissait de choses comparables. Il manque à l'évidence quelques paliers entre les deux. Le passage qui suit ne prétend pas les remplacer. Il n'est qu'un condensé de ce qu'on trouve sur Wikipedia, dans les rubriques protocoles, serveurs, hébergeurs, et qui est plus complet en anglais. Il y est seulement question de sérier quelques notions qu'on ne peut pas complètement ignorer, même s'il n'est pas indispensable de tout en retenir pour tirer profit de cet ouvrage.

### ***Les protocoles du net***

L'internet repose sur des protocoles de transfert de données numériques :

Le **FTP**, *File Transfer Protocol*. Il concerne le téléversement sur le disque dur de serveur des fichiers numériques pour les rendre accessibles à tous les ordinateurs connectés.

Le **HTTP**, *Hyper Text Transfer Protocol*, qui concerne la lecture et ou le téléchargement de tous les contenus téléversés sur les serveurs.

Le **SMTP**, *Simple Mail Transfer Protocol*, qui concerne le courrier électronique.

Il existe bien d'autres protocoles réseaux, plus ou moins bien connus. Il n'est pas très utile de tous les connaître, mais comme on ne manquera pas de retrouver leurs noms dès qu'on connecte le moindre appareil électronique, il n'est pas plus mal de comprendre de quoi il s'agit.

Un protocole réseau est un protocole de communication mis en œuvre sur un réseau informatique. Il est possible que plusieurs protocoles réseau forment des couches de protocoles.

Afin de s'y retrouver au milieu des protocoles, et même de pouvoir en changer, on les hiérarchise par exemple en quatre couches dans le modèle TCP/IP. D'autres modèles plus complexes, comme le SNA d'IBM ou le modèle de l'OSI, tous deux à sept couches, ont rencontré moins de succès pratique et sont plutôt évoqués aujourd'hui à des fins de complétude de la théorie.

Considérée dans son ensemble, une suite de protocoles entre diverses couches forme ce qu'on nomme une pile de protocoles. Les termes « protocole » et « pile de protocoles » désignent également les logiciels qui implémentent un protocole.

The image shows a screenshot of an FTP client's login and configuration window. At the top, there is a 'Protocol' dropdown menu set to 'FTP' and a 'Site List' dropdown. Below this is a 'Host/URL' text field containing 'fttperso.free.fr', with 'Port' and 'Get' buttons to its right. A note below reads: 'For Public (anonymous) FTP servers, leave these fields blank'. Underneath are 'Username' and 'Password' text fields; the username is 'concubit' and the password is masked with dots. A 'Login Options' section contains an 'Initial Path' field with '/documents', a 'Show Transcript' checkbox (unchecked), a 'Show Home Folder Only' checkbox (checked), a 'Use SSL' checkbox (unchecked), and a 'Passive On' dropdown menu. At the bottom, there are two buttons: 'Site Preferences' and 'Connect'.

Les protocoles les plus récents sont standardisés par l'IETF dans le cas des communications sur Internet, et par l'IEEE ou l'ISO pour les autres types de communication. L'UIT-T prend en charge les protocoles et les formats des télécommunications.

Dans le milieu du P2P, on considère l'ensemble des algorithmes d'un réseau comme son protocole.

### *Protocoles de couche 2*

Cette couche correspond à la couche de liaison de données dans le modèle OSI. Beaucoup de périphériques utilisent de tels protocoles : Ethernet ; Fiber Distributed Data Interface (FDDI) ; HDLC ; Logical Link Control (LLC) ; Media Access Control (MAC) ; Point-to-point protocol (PPP) ; etc.

### *Protocoles de couche 3*

Cette couche correspond à la couche de réseau dans le modèle OSI. À ce niveau, on trouve par exemple les protocoles suivants : Internet Protocol (IP) ; contenant Border Gateway Protocol (BGP) ; Routing information protocol (RIP) ; Open shortest path first (OSPF) ; Integrated Intermediate System to Intermediate System (Integrated IS-IS) ; Internet Control Message Protocol (ICMP) avec Address Resolution Protocol (ARP) ; IPX ; etc.

### *Protocoles de couches 3 et 4*

Xerox Network Services (XNS).

### *Protocoles de couche 5 et plus*

Cette couche correspond à la couche de session dans le modèle OSI. À ce niveau, on trouve par exemple : FTP ; SSH ; SFTP ; HTTP ; IMAP ; NFS ; POP3 ; protocole Samba ou SMB/CIFS ; RSerPool ; SNMP ; SMT ; Telnet ; etc. (*Sources Wikipedia.*)

C'est à la fois simple et déconcertant. C'est simple, car les connaissances mécaniques qui sont mises en œuvre ne vont pas bien au-delà de celles du dix-neuvième siècle. Seule une « science numérique », logico-mathématique, permet de tirer un tel parti de la mécanique et d'aller si loin au-delà de l'invention de Marconi.

Le secret consiste à moduler et démoduler des suites binaires, et surtout à les encapsuler, c'est-à-dire à les enfermer entre un signal entrant et un signal sortant. À partir de là, un nombre incalculable de signaux les plus divers peuvent circuler par les canaux les plus variés sans se mêler ni se corrompre, et sans nécessiter de hiérarchie ni d'ordre.

### ***Les serveurs***

Dans un réseau informatique, un serveur est un ordinateur et un logiciel qui répondent automatiquement aux demandes des *clients* (ordinateur personnel, poste de travail, terminal et logiciel) à travers le réseau. Ils sont utilisés simultanément pour stocker, partager et échanger des informations. Le *World wide web*, la messagerie électronique et le partage de fichiers sont des applications de serveurs.

Dans une application distribuée, les traitements sont répartis entre plusieurs ordinateurs d'un réseau informatique. Un protocole de communication établit les règles selon lesquelles les ordinateurs communiquent et coopèrent. Dans une architecture client/serveur les traitements sont effectués par l'exécution conjointe de deux logiciels différents et complémentaires placés sur des ordinateurs différents: le client et le serveur.

Le client formule des requêtes, puis les transmet au serveur. Le serveur reçoit et traite les demandes, puis envoie sa réponse au client. Un protocole de communication établit le format des requêtes envoyées au serveur, et des réponses de celui-ci.

Le mode pair-à-pair (anglais *peer-to-peer* ou P2P) est l'opposé du mode client-serveur.

### *Systèmes d'exploitation et logiciels*

Il existe de nombreux systèmes d'exploitation optimisés pour le travail de serveur, notamment dans les familles Unix, Mac OS X et Windows NT. Les serveurs étant utilisés en réseau, le système d'exploitation est équipé de divers logiciels utilisés avec de nombreux protocoles, et sont utilisés comme relais ou pour du routage. Divers logiciels serveur sont intégrés dans le système d'exploitation.

Est aussi appelé serveur un logiciel qui traite des demandes envoyées à travers un réseau. Un logiciel serveur donné peut coopérer avec n'importe quel client qui utilise le même protocole. De même un logiciel client donné peut coopérer avec n'importe quel serveur du même protocole.



## ***Principales utilisations d'un serveur***

### *Serveur de fichiers*

Le serveur stocke et partage les fichiers. Les fichiers dans le serveur peuvent être utilisés simultanément par plusieurs clients. Les serveurs de fichiers sont souvent inclus dans les systèmes d'exploitation. CIFS est le protocole du serveur de fichier et d'impression inclus dans les systèmes d'exploitation Windows. NFS est un protocole courant des serveurs de fichiers des systèmes d'exploitation Unix. Samba est un serveur en protocole CIFS pour les systèmes d'exploitation Unix. FTP est un protocole de serveur de fichier très utilisé sur le réseau internet.

### *Serveur d'impression*

Le serveur est utilisé entre des clients, et chacun peut envoyer des documents à une des imprimantes reliée au serveur. Le serveur envoie les documents en file l'imprimante. Les documents peuvent subir des changements de formats pour les adapter aux caractéristiques de l'imprimante. PostScript est le plus courant formats de données en impression numérique. CIFS est le protocole des serveurs d'impression et de fichiers des systèmes d'exploitation Windows. CUPS est un serveur d'impression courant sur les systèmes d'exploitation Unix..

### *Serveur de base de données*

Le serveur est utilisé pour stocker, et utiliser des données dans une ou plusieurs bases de données et partagées entre les clients. Le serveur de base de données fait partie d'un système de gestion de base de données (abréviation SGBD) qui comporte un logiciel client et un logiciel serveur. Les demandes de manipulation de données sont souvent créées par un logiciel de gestion sous forme de requêtes en langage SQL, puis le client les transmet au serveur en utilisant un protocole propre au SGBD. Oracle, Microsoft SQL Server ou MySQL sont des SGBD qui comportent un serveur de base de données.

Une application informatique en architecture trois tiers comporte typiquement un serveur de base de données (tiers 3) avec un serveur web (tiers 2) et un navigateur web (tiers 1). L'application hébergée par le serveur web est le client du serveur de base de données.

### *Serveur de courrier*

Le serveur est utilisé pour stocker et transmettre du courrier électronique. Le courrier partant est transmis au serveur du destinataire. Le courrier arrivé est stocké sur le serveur et est consultable par le client. Un Serveur de courrier réponds à des demandes d'acheminement de messages

électroniques. Le serveur stocke les messages arrivés, et transmet les messages en partance au destinataire (un autre serveur de courrier).

SMTP est un protocole utilisé pour acheminer un message électronique entre un client et le serveur, aussi bien qu'entre un serveur et un autre serveur. POP est un protocole utilisé pour récupérer les messages stockés et les transmettre au client. IMAP est un protocole utilisé pour examiner et manipuler les messages stockés à partir d'un client. Les messages en partance sont souvent placés dans des files d'attente, et envoyés groupés à intervalles réguliers (voir spooling).

Un webmail est une application placée sur un serveur web, qui agit comme un client sur un serveur de courrier.

### *Serveur web*

Le serveur stocke et manipule les pages d'un site Web, et les transmet sur demande au client (navigateur web). Un serveur web répond à des demandes de consultation de documents tels que ceux du *World wide web*. Chaque passage d'un document hypertexte à l'autre par parcours d'un hyperlien dans le logiciel client (la plupart du temps un navigateur web) provoque l'envoi d'une nouvelle demande. Chaque demande peut être envoyée à un serveur différent. HTTP et HTTPS sont les protocoles des serveurs web.

Apache est un serveur web pour les systèmes d'exploitation Unix et Windows. Internet Information Services (abrégé IIS) est un serveur web inclus dans les systèmes d'exploitation Windows. Une application web est un logiciel composé d'un ensemble de traitements qui créent des documents instantanés suite à la réception de demandes de consultation. Le webmail, le wiki, et le weblog sont des applications web.

### *Autres*

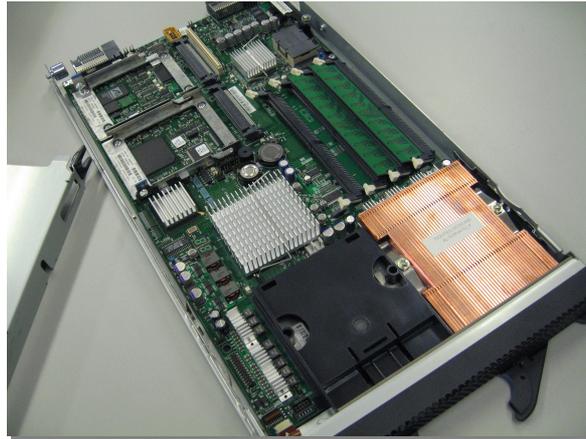
Serveur de jeu : Le serveur arbitre et suit l'évolution d'un jeu. Il met en communication les différents joueurs d'un jeu en ligne multijoueur. Le serveur fait partie d'un logiciel de jeu en ligne multijoueur en architecture client-serveur. Il répond à des demandes de diffusion des opérations effectuées par chaque joueur. Le serveur valide les opérations de chaque joueur, arbitre le jeu, puis transmet les évolutions du jeu aux autres joueurs.

Serveur d'applications : Désigne un serveur qui effectue les traitements d'un ou plusieurs logiciels d'architecture client/serveur.

Serveur mandataire (*proxy*) : Le serveur reçoit des demandes, les contrôle, puis les transmet à d'autres serveurs. Il peut être utilisé pour accélérer le traitement des demandes, ou faire appliquer des règlements (censure, confidentialité, etc.)

### ***Les hébergeurs***

Un hébergeur offre, ou loue, des espaces sur des disques ou des sites peuvent être hébergés et mis à la disposition des visiteurs. Il maintient des ordinateurs connectés 24 heures sur 24 à l'Internet, sur lesquels sont installés des logiciels : serveur HTTP (souvent Apache), serveur de messagerie, de base de données...



Lorsqu'un visiteur demande une page à son navigateur Web, celui-ci interroge des serveurs DNS pour connaître l'adresse IP du serveur hébergeant ce site. Dès qu'il obtient la réponse, le navigateur va interroger ce serveur et lui demander cette page. Le serveur web va alors chercher la page sur son ou ses disques durs (s'il s'agit d'une page statique), ou la fabriquer à l'aide d'un script (s'il s'agit d'une page dynamique), puis l'envoyer au navigateur, qui l'affiche sur l'écran du navigateur.

#### *Les diverses offres d'hébergement*

**Les hébergements partagés ou mutualisés** : Chaque serveur héberge plusieurs sites, jusqu'à plusieurs milliers, dans le but de mutualiser les coûts. Le principal avantage est le prix, le principal inconvénient est que le client mutualisé n'est pas l'administrateur du serveur, il est donc souvent tributaire du bon vouloir de l'hébergeur s'il souhaite une technologie particulière. Dans certains cas, l'utilisateur peut être administrateur d'un serveur virtuel sur lequel son site est installé. Il partage cependant les ressources système avec les autres clients mutualisés.

**Les hébergements dédiés** : Le client dispose de son propre serveur, et peut en général l'administrer comme il le souhaite, ce qui est le principal avantage de ce type d'offre. Le fournisseur du serveur reste propriétaire du serveur. Les inconvénients sont le prix beaucoup plus élevé que les hébergements mutualisés, et le besoin de compétences pour administrer la machine correctement.

**L'hébergement virtuel dédié via un hyperviseur** qui offre au client la souplesse d'un dédié (le client administre sa machine comme il le souhaite) en lui fournissant une machine virtuelle qui utilise une partie des ressources d'un serveur (physique) par des techniques de virtualisation.

**Les hébergements dédiés dits « managés », avec « serveur dédié infogéré » ou « clés en main »** : Le client dispose de son propre serveur mais les techniciens de l'hébergeur s'occupent de sa gestion système. Cette solution est parfaite si l'on n'a aucune connaissance technique d'administration. Il suffit de déposer son site.

**La colocation** : L'hébergeur met, dans son centre de traitement de données, un espace à disposition du client, de sorte qu'il puisse placer son propre serveur à l'intérieur (La plupart du temps dans des armoires spéciales nommées racks ou « baies »). L'hébergeur met également à disposition du client un câble d'alimentation électrique et un câble ethernet pour qu'il puisse alimenter et connecter son serveur à l'internet. Ce système coûte en principe moins cher, mais les systèmes de sécurité et des badges dans les centres de traitement de données peuvent coûter plus cher que la location.

La plupart des hébergeurs sont payants, il en reste quelques gratuits, le plus souvent il s'agit d'offres d'appel pour leurs prestations payantes, plus riches en fonctionnalités et plus performantes.

## ***L'ÉPOQUE NUMÉRIQUE ET PRÉNUMÉRIQUE***

La spécificité du web est donc de permettre une relation directe, d'égal à égal et sur un même plan, tout en lui permettant de demeurer ouverte, de se faire publique. La simplicité de cet énoncé ne doit pas cacher que c'est là quelque chose de bien difficile à concevoir.

Un telle réalité serait inconcevable sans ces trois inventions intriquées que sont le numérique, l'ordinateur personnel et l'internet. Elle est tout simplement inconcevable pour, disons, l'homme prénumérique (comme on dit préhistorique).

### ***L'usage du net***

Le web ne sert pas plus à faire des masses disciplinées marchant d'un même pas dans la même direction, admirant les mêmes célébrités, gobant les mêmes nouvelles, ressassant les mêmes idées... qu'il ne sert à enfermer des individus sur eux-mêmes, à les isoler, les nombriliser. Ce ne sont là que les limites de ce que peut concevoir l'homme prénumérique. Ces deux extrêmes, on doit bien s'en rendre compte, reviennent d'ailleurs au même, et décrivent assez bien ce que faisait déjà la consommation spectaculaire.

Naturellement, le web sert cela aussi, et c'est ce qu'on lui demande. Les ordinateurs personnels, les connexions haut débit, les hébergements, les logiciels... ne seraient pas si accessibles si le web ne se prêtait pas à ces usages dominants, qui sont les seuls commercialement viables, et qui constituent les deux premiers niveaux que j'ai décrits.

### ***Les limitations du net***

Un tel usage est cependant un usage détourné et limité. Par limité, on doit comprendre plusieurs choses :

— La première est qu'il s'agit d'un sous-emploi des moyens réels, comme si l'on utilisait une clé-à-griffe pour tourner une poignée de porte.

— La deuxième est que ce sous-emploi exige de se retenir, de limiter soi-même ses moyens. (Sans précautions, une clé-à-griffe risquerait d'arracher la poignée plutôt que d'ouvrir la porte.) Se retenir concerne alors aussi bien l'utilisateur individuel, les pouvoirs publics et tous les fournisseurs d'équipements et de moyens.



En somme, le troisième niveau est plus simple à utiliser que les deux premiers, mais les chemins pour y parvenir sont brouillés. Pour dire les choses paradoxalement, le type d'usage auquel introduit cet ouvrage est tout à la fois le plus expert et le plus basique.

— La troisième est que cette auto-limitation de l'usage est une source de complications. Il est en somme plus difficile de sous-employer le web, plus techniquement complexe, que d'en tirer toutes les possibilités.

La sorte d'usage qu'implique ce troisième niveau est le plus simple et le plus naturel du web, et le plus aisé à mettre en œuvre. Il ne nécessite même pas les outils de ce que l'on appelle le Web.2, le PHP ou MySQL.

### *La personne et le web*

J'ai dit que le web n'avait pas de centre, et c'est relativement faux. Dès que vous vous connectez, il en a un, vous. Certes, on n'a pas besoin de l'internet pour se croire le centre du monde, mais là ce n'est pas qu'une impression. Il suffirait que tout le monde se déconnecte, et il n'y aurait plus de web.

Ce n'est pas une plaisanterie ; la moindre trace que vous laissez sur le net devient un nœud de croisements, le centre d'un réseau rayonnant sans fin. Vous pouvez chercher à voir sans être vu, prendre sans rien donner, vous faire invisible, indécélable, vous cacher sous un avatar et un pseudo, masquer votre IP, porter une fausse barbe, effacer vos traces : peine perdue, vous êtes tracés. Alors autant tomber le masque, montrer votre vrai visage et offrir une main franchement tendue.

Mais à qui et pour quoi ? Voilà la grande innovation de l'internet, et que l'on sent confusément dès qu'on s'y montre aussi peu que ce soit. C'est même très exactement le premier problème qui se pose quand on fait un site, perso ou pas. Dans presque tous les aspects de la vie, il nous est relativement facile de donner une image unidimensionnelle, unipolaire, univalente. Nous nous construisons une image dans notre famille, dans notre travail, avec nos amis, nos voisins, dans nos associations et nos clubs. Nous changeons de costumes, et personne ne nous voit jamais dans la totalité des rôles que nous interprétons. De même, une entreprise ne présente pas le même visage à ses collaborateurs, ses partenaires, ses fournisseurs, ses clients.

Sur le web, cela paraît d'abord plus facile encore. On sent tous les mensonges possibles, et en effet ils le sont, mais d'un autre côté, toutes les données sont recoupables.

Il est tentant de se construire une image dans un réseau, et une seconde dans un autre, mais l'internet est « le réseau des réseaux ». Il peut être gênant de tenir ici des propos excessifs, et là, se donner l'image d'un personnage respectable et mesuré.

Nous avons immédiatement sur le web les soucis d'un personnage public, mais sans susciter les mêmes intérêts. De prime abord, les autres ne se soucient pas de nos différentes facettes. Si nous les intéressons professionnellement, ils ne cherchent pas nos goûts musicaux (sauf si nous sommes musiciens). Si quelqu'un attend de nous des aides techniques parce que nous utilisons le même système ou le même logiciel, il ne s'intéresse pas à nos idées politiques (sauf si elles concernent ce programme). Comment gérer cela ?

L'image qu'on se construit en ligne est faite pour quiconque, pas pour un groupe précis qui en imposerait assurément une lui-même, souvent sans même connaître quelqu'un autrement que par un CV, un diplôme, une recommandation, voire un costume ; mais « quiconque » n'existe pas.

Nous devons bien nous reconnaître sur le net comme ce point de jonction de plusieurs réseaux qui convergent sur nous et ne sont plus étanches.

### *L'autre côté de l'écran*

Comme il mêle les cartes entre public et privé, le web brouille le rapport entre la fugacité de la parole et la consistance de l'écrit. Les écrits restent sur le net, bien plus sûrement que sur le papier. Le moindre commentaire des plus désinvoltes ne s'efface pas tout seul, ne s'égare même pas.

On ne se doute pas à quel point tout ce qu'on peut dire, penser ou faire un jour, on ne le dirait, ne le penserait plus, ne le ferait plus dix ans plus tard. On y songe davantage quand on écrit un livre, même si l'on sait qu'il sera archivé, oublié et pilonné en trois ou quatre ans. Tout, au contraire, reste en ligne sur le net, comme si l'encre était encore fraîche, d'autant plus que, même si nous comprenons l'utilité de dater visiblement nos pages, nous négligeons presque toujours de le faire.

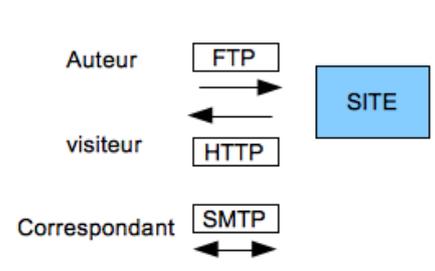
De telles idées auraient de quoi nous rendre très nerveux quand nous publions en ligne, et nous donner au moins le trac que nous ressentirions sur une tribune. Mais nous nous trouvons généralement dans un espace familier, dans la quiétude de notre bureau, du fond d'un « chez soi » (*home*) qui se confond presque avec une « vie intérieure », et qui ouvrirait sur un « autre côté de l'écran », comme l'autre côté du miroir carollien.

L'homme prénumérique que nous demeurons tend à voir dans le web un autre côté du miroir, un monde autre et fondamentalement opposé au réel. L'homme prénumérique peine à concevoir que « l'autre côté de l'écran » n'est jamais qu'un autre écran que son utilisateur regarde en face.

À proprement parler, il n'y a pas de contenu sur le web, le web n'est qu'un réseau qui met en relation des personnes réelles, dans le monde réel, qui font des choses réelles, et dont il est sans doute plus facile de s'assurer de l'authenticité devant son écran que partout ailleurs.

### *Principes de base sur les sites et le courriel*

L'internet est constitué de deux sortes de choses qui fonctionnent en complément : les sites, avec leurs adresses qui commencent pas « http://... », et le courrier électronique. Les sites sont mis sur des serveurs distants par le protocole FTP, et consultés par le protocole HTTP. Le courrier circule par le protocole SMTP.



C'est aussi simple que ça, et l'on fera bien de garder ce modèle à l'esprit, quelles que soient les complications souvent inutiles qui pourront s'y agréger.

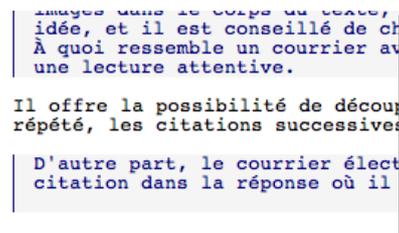
### *Le courriel*

En principe, un site est fait en HTML, et le HTML peut y suffire. En principe, le courrier s'écrit en texte brut, et le texte brut lui suffit.

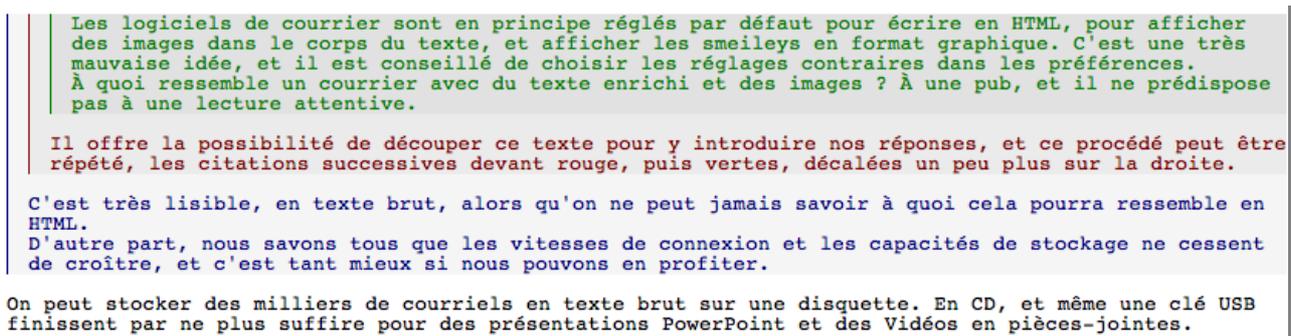
Les logiciels de courrier sont en principe réglés par défaut pour écrire en HTML, pour afficher des images dans le corps du texte, et afficher les smileys en format graphique. C'est une très mauvaise idée, et il est conseillé de choisir les réglages contraires dans les préférences.

À quoi ressemble un courrier avec du texte enrichi et des images ? À une pub, et il ne prédispose pas à une lecture attentive.

Le courrier électronique a cet avantage que le texte reçu est immédiatement recopié comme citation dans la réponse où il s'affiche en bleu, décalé par une ligne bleue verticale.



Il offre la possibilité de découper ce texte pour y introduire nos réponses, et ce procédé peut être répété, les citations successives devenant rouge, puis vertes, décalées un peu plus sur la droite.



C'est très lisible, en texte brut, alors qu'on ne peut jamais savoir à quoi cela pourra ressembler en HTML.

D'autre part, nous savons tous que les vitesses de connexion et les capacités de stockage ne cessent de croître, et c'est tant mieux si nous pouvons en profiter. Cependant, même une connexion ADSL peut se partager dans des structures assez grandes, réduisant en proportion la vitesse de connexion de chacun. Quant aux toujours plus grandes capacités de stockage, elles n'obligent pas de jeter les anciens supports qui rendent encore des services. On peut stocker des milliers de courriels en texte brut sur une disquette. Un CD, et même une clé USB finissent par ne plus suffire pour des présentations PowerPoint et des Vidéos en pièces-jointes. Certes, cela concerne généralement ces lourds courriels que l'on jette dès qu'on en a pris connaissance. Qu'on en tienne donc compte en postant son courrier.

### ***La complémentarité des sites et des courriels***

Les sites et les courriels sont d'ailleurs complémentaires principalement pour cette raison-là. Il ne sert à rien d'envoyer des images, de longs textes ou toute sorte de documents qui se trouvent en ligne. Il suffit d'en envoyer l'adresse, l'URL (*Uniform Resource Locator*). Collée sur un courriel, elle deviendra immédiatement un lien actif où il suffira de cliquer.

Si ces documents ne sont pas déjà en ligne, on n'a qu'à les placer sur son site. C'est à quoi sert le web, et tous les sites qui le composent : à partager des documents.

Les documents sont partagés en ligne avec quiconque. Mais quiconque, on ne le connaît pas. On envisage toujours de partager des documents avec des gens précis. On peut opter alors pour les rendre accessibles à d'autres ou non.

Arriver à faire tout cela n'est pas particulièrement compliqué, mais demande quand même un peu d'attention et d'apprentissage. Pour permettre de parvenir malgré tout aux mêmes résultats sans rien comprendre ni rien apprendre, on a inventé beaucoup de complications dont on pourra faire ici l'économie.

## ***LE PAPIER ET L'ÉCRAN***

### ***Le document et le programme***

Papier et écran ne sont pas dans une concurrence réelle. Le fichier numérique est au contraire voué à passer à tout moment de l'un à l'autre. C'est tout l'intérêt du fichier numérique de n'être jamais tout à fait, ni tout entier, sur l'écran ni sur le papier. Il n'est jamais définitivement imprimé, ni définitivement affiché, ni non plus, et ce dernier point a une importance considérable, tout entier dans le programme.

En effet, l'opposition entre le papier et l'écran ne doit pas en masquer une autre plus fondamentale entre le document et le programme. Un document doit être autonome de tout programme particulier. Il pourrait être en effet préjudiciable que notre fichier ne soit plus utilisable si nous faisons appel à un autre, ou qu'il soit du moins partiellement corrompu.

Même un format aussi standard que le PDF peut réserver de mauvaises surprises. Il suffit de changer d'imprimeur, ou encore de version d'un logiciel, pour que le fichier PDF produit exactement de la même façon qu'on s'y est toujours pris, laisse imprimer un texte sans illustration, des illustrations sans texte, ou donne des feuilles de papier immaculées. Et le HTML, pourtant si souple, avec lequel on avait soigné la mise en page qui s'affichait si bien dans Explorer, devient chaotique dans un autre navigateur. Quant à rouvrir dans un autre programme un fichier XPress ou InDesign, il ne faut même pas y songer.

Naturellement, tout document doit être plus autonome encore de tout système informatique ou de tout dispositif matériel, d'affichage, de programmation ou d'impression. Imaginons un peu qu'un texte écrit au stylo Bic ne soit plus modifiable avec de l'encre Waterman. Si l'on n'a pas cela en tête, tout ce dont il est question dans cet ouvrage se réduirait à une vaine « prise-de-tête ».

### ***L'affichage et l'impression***

L'affichage et l'impression ne sont pas en concurrence, et justement parce qu'ils ne le sont pas, et que tout ce qui est affiché peut devoir être imprimé, et bien souvent industriellement, comme tout ce qui est imprimable doit bien être affiché, au moins pour l'éditer. Leur compatibilité, donc, est un problème permanent. Il est d'autant plus un problème qu'il inclut le précédent, celui de l'autonomie du document et du programme.

La difficulté dépend en grande partie de nos exigences. Si l'on en a peu, et qu'on se contente d'aligner du texte aussi pauvrement que sur une vieille machine à écrire, tout est un peu plus simple, mais à peine un peu plus : les principaux problèmes demeurent.

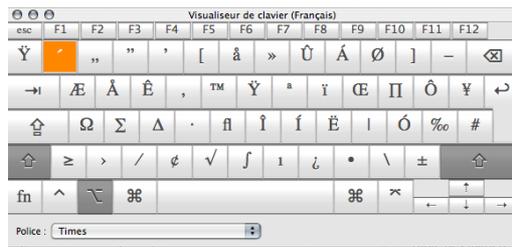
Le premier est de rendre correctement certains caractères qui ne font pas partie du jeu standard, les accents notamment. On ne peut quand même pas renoncer aux caractères spécifiques à certaines langues, au français tout particulièrement. Le second, est celui de la séparation des paragraphes, qui cesse d'être visible sans justification ni alinéas.

### La typographie

Les machines à écrire n'ont jamais respecté les règles de la typographie française. On peut continuer de les ignorer, renoncer aux guillemets à la française, ou à certains espaces, comme devant le point d'interrogation, ou après un guillemet ouvrant. Si l'on n'y renonce pas, on devra s'assurer que ces espaces deviennent insécables, car il serait pire qu'un point d'interrogation se retrouve en début de ligne, ou un guillemet ouvrant à la fin d'une autre.

Beaucoup de traitements de texte gèrent eux-mêmes la typographie française, mais on peut parfois souhaiter s'en passer (si l'on écrit un ouvrage sur du code par exemple). Il est donc utile de savoir comment les écrire au clavier.

L'espace insécable s'obtient à l'aide de la touche contrôle (option sur Mac) et espace. Pour les autres combinaisons de touches, le mieux est d'afficher son clavier.



Ci-dessus, les touches enfoncées sont en gris. On voit donc aussi comment composer les lettres liées ou certains symboles.

Dans tous les cas, on devra s'assurer que ces caractères et ces espaces sont bien exportés et importés.

	espace avant	espace après
,	non	oui
;	insécable	oui
:	insécable	oui
?	insécable	oui
!	insécable	oui
—	oui	insécable
...	non	oui
'	non	non
-	non	non
" "	oui - non	non - oui
()	oui - non	non - oui
{}	oui - non	non - oui
[]	oui - non	non - oui
« »	oui - insécable	insécable - oui

Les puristes retiendront que les espaces peuvent exiger 1, 1/2 et 1/4 de cadratins, mais seulement quelques logiciels voués à la typographie peuvent le permettre. On trouvera d'ailleurs difficilement des livres respectant ces critères en librairie.

*Les conventions de mise en page*

Il est aujourd'hui possible, et même relativement facile avec un peu de pratique, d'appliquer en ligne les mêmes règles de mise en page que dans l'imprimerie. Il est préférable cependant de tenir compte de certaines spécificités de l'écran. Tout d'abord, l'éclairage venant de l'intérieur de l'écran, le contraste entre le blanc et le noir fatigue les yeux. Une couleur de fond que ne diminue pas la lisibilité est donc bien venue.

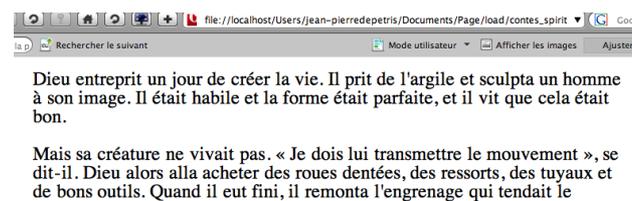
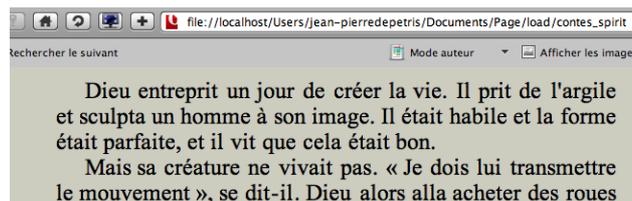
Certaines conventions de mise en page se sont au début imposées sur le net, sensiblement différentes de celles de l'imprimerie.

PAGES IMPRIMÉES	PAGES WEB
Pages Justifiées	Fer à gauche
Alinéas en début de paragraphes	Espace après chaque paragraphe
Marges (intérieure pour la reliure, extérieure pour tenir le livre ou le cahier en main)	Pas de marges (rendues inutiles par la barre de défilement)
Sauts de pages	Lignes horizontales (Convertis automatiquement par certains logiciels lors d'une exportation)

Comme il était impossible au début de justifier le texte et de placer des alinéas, la distinction entre les paragraphes n'était pas toujours visible. On laissa donc un espace entre eux. On peut choisir de conserver cette coutume, ou revenir à celles de l'imprimerie, puisque c'est maintenant possible. Ces deux options sont parfaitement justifiables.

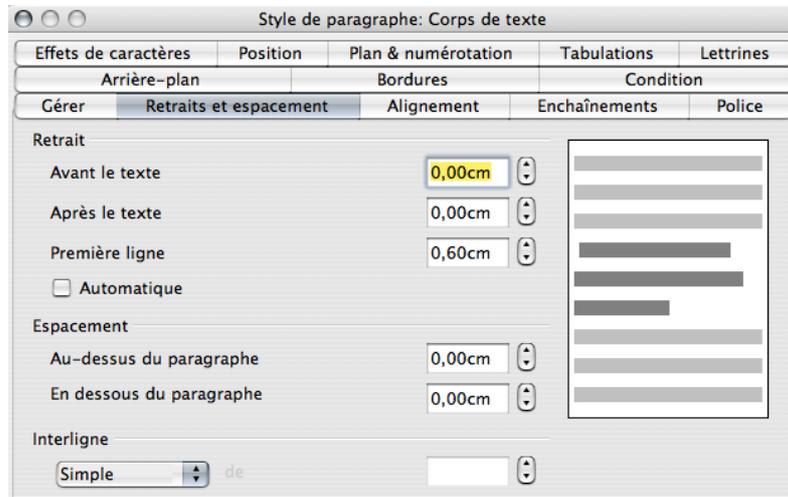
Comme tout texte est destiné à être imprimé et à être affiché, on devra seulement prendre garde à ne pas mélanger les deux. Si le texte est bien saisi, il est facile de basculer d'une mise en page à l'autre. Pour cela, on doit éviter les sauts de lignes vides.

Il est d'ailleurs très facile de passer de l'une à l'autre. Il suffit d'ouvrir la page avec un navigateur qui, comme Opéra, permet facilement de décocher la lecture des CSS en passant du « mode auteur » au « mode utilisateur ». Le texte programmé selon les conventions de l'édition s'affichera par défaut selon celles du net.



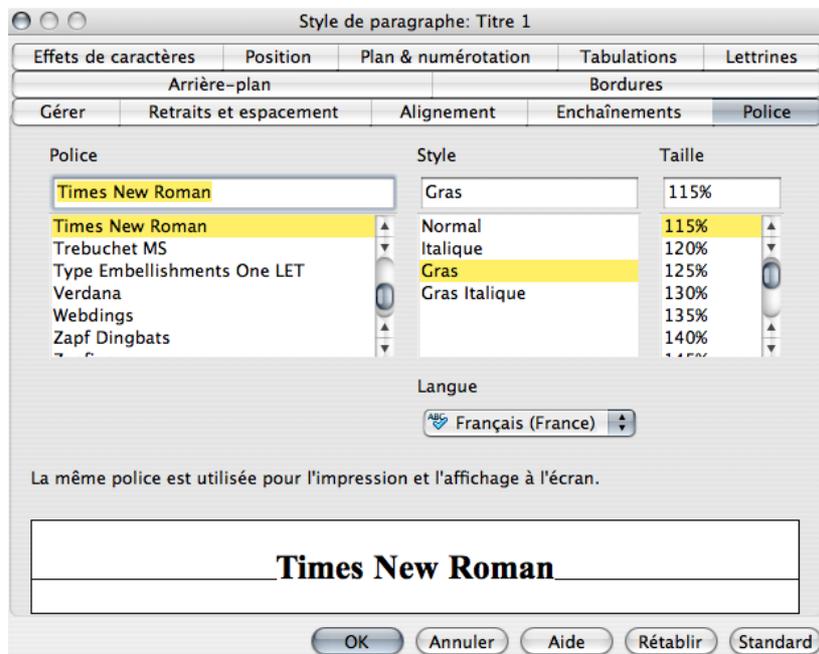
*Proscrire les sauts de ligne.*

Il est préférable de gérer les espaces autour du paragraphe dans l'éditeur de style, plutôt que de sauter des lignes avant et après.



*Utiliser les styles*

Pour être capable de modifier en bloc l'apparence et la structure du texte entier, on doit systématiquement utiliser les styles pour les titres et les sous-titres plutôt que modifier chaque fois la taille, l'alignement, l'alinéa...



Si tout est bien fait, les balises exportées seront claires : H1, H2, H3... P, LI, etc. et non pas un nombre ingérable de CLASS.



Tout cela et tout ce qui va suivre peut ressembler à des casses-têtes inutiles dont on n'a pas envie de s'occuper quand on écrit, où tout encourage plutôt à aller au plus vite. Il semble évidemment plus simple de glisser un curseur à la souris pour créer un alinéa, changer la police, le corps ou l'alignement d'un paragraphe, d'enfoncer la touche retour pour créer un espace... Ces commodités dans l'immédiat génèrent pourtant de pires casses-têtes dans la durée.

Tout ce dont il est question ici et par la suite, est à la fois complexe et simple contradictoirement. La contradiction est qu'en principe toutes ces techniques devraient être parfaitement maîtrisées avant d'entreprendre d'écrire, alors que le meilleur moyen de les acquérir serait de les prendre en main les unes après les autres et d'améliorer perpétuellement par étapes ses méthodes de travail. C'est comme si l'on exigeait de savoir d'abord avant d'apprendre.

Le mieux, pour trancher cette contradiction, est de faire d'abord au plus simple, d'éviter toutes les complications inutiles, les exigences qui ne sont pas indispensables ; et à chaque occasion, chaque fois qu'on recommence quelque chose, ne pas craindre de parfaire un peu plus.

Tout ce qui suit pourra paraître compliqué, mais, d'une part, quand on l'aura déjà fait plusieurs fois, tout deviendra plus simple ; et d'autre part, il sera rarement nécessaire de tout reprendre à zéro. L'une des principales utilités de la programmation est de ne pas devoir refaire inutilement ce qu'on a fait une fois.

Par exemple, tous les réglages de pages et de styles pourront être enregistrés dans des modèles, et des opérations plus complexes pourront l'être sous forme de scripts exécutable ou de petites applications.

On ne doit pas hésiter à prendre parfois des heures pour comprendre et enregistrer des procédures automatiques qui épargneront sans doute d'innombrables fractions de secondes, qui avant tout libèreront l'esprit pour des activités qui le requièrent tout entier, et qui enfin épargneront bien d'autres efforts, vains cette fois, pour émanciper son ouvrage des outils qui auront servi à le produire.

# Écrire avec un ordinateur

## ***LE TEXTE***

Écrire avec un ordinateur n'est pas fondamentalement différent d'écrire avec la plume ou le pinceau. Tout le monde a d'ailleurs commencé par utiliser un ordinateur comme une machine à écrire perfectionnée ; une machine à écrire qui facilitait la correction, le couper et le coller, la copie des documents, la correction orthographique ou grammaticale, l'enrichissement du texte, etc.

L'ordinateur a aussi servi d'outil de publication. C'était justement le même outil, ou presque. C'était le même ordinateur, le même système, mais des logiciels souvent différents. C'était l'époque où Word et XPress ont envahi le marché, et il était de toute façon possible de publier avec Word et d'écrire avec XPress.

Au début, le texte a continué à être ce qui sortait de l'imprimante et qu'on appelait toujours « manuscrit » ; et la publication, ce qui sortait de l'imprimerie. Dans tous les esprits, la place du texte était encore sur le papier, et cette façon de voir commençait à être corrigé par des expériences cuisantes : la perte des données et l'incompatibilité des formats.

On s'aperçut vite qu'il était bien plus simple d'imprimer un fichier numérique que de numériser un texte imprimé. Alors le texte, le « vrai texte », a commencé dans nos esprits à devenir le contenu du fichier numérique, plutôt que l'encre sur le papier.

L'internet a renforcé cette évolution. Rien n'est plus simple que de communiquer des documents par courriels ou de les publier en ligne.

On doit d'abord comprendre qu'avec le numérique tout est texte. Et qu'est-ce que le texte ? Des suites de caractères qui constituent des énoncés, des expressions, des commandes. Les caractères sont aussi bien des lettres, des chiffres, ou divers connecteurs comme les signes de ponctuation. Tout est texte, pas seulement les écrits en langues naturelles, mais aussi le code, les algorithmes.

Le texte est proprement le plancher, sous lequel sont des suites hexadécimales et binaires, et finalement le dispositif mécanique lui-même. Au-dessus, en passant par la langue naturelle, est tout le côté humain de l'interface. Le texte est proprement l'interface homme-machine : celui des caractères sur le clavier, celui qui s'affiche à l'écran, et l'on doit bien en passer par lui.

## ***LES JEUX DE CARACTÈRES***

### ***L'US-ASCII***

On peut le nommer plus simplement ASCII.

Ce jeu est très incomplet pour la plupart des langues européennes, mais il est bien suffisant pour écrire du code.

## ÉCRIRE AVEC UN ORDINATEUR

0	16	32	space	48	0	64	@	80	P	96	`	112	p
1	17	33	!	49	1	65	A	81	Q	97	a	113	q
2	18	34	"	50	2	66	B	82	R	98	b	114	r
3	19	35	#	51	3	67	C	83	S	99	c	115	s
4	20	36	\$	52	4	68	D	84	T	100	d	116	t
5	21	37	%	53	5	69	E	85	U	101	e	117	u
6	22	38	&	54	6	70	F	86	V	102	f	118	v
7	23	39		55	7	71	G	87	W	103	g	119	w
8	24	40	(	56	8	72	H	88	X	104	h	120	x
9	tab	25	)	57	9	73	I	89	Y	105	i	121	y
10	lf	26	*	58	:	74	J	90	Z	106	j	122	z
11		27	+	59	;	75	K	91	[	107	k	123	{
12		28	,	60	<	76	L	92	\	108	l	124	
13	or	29	-	61	=	77	M	93	]	109	m	125	}
14		30	.	62	>	78	N	94	^	110	n	126	~
15		31	/	63	?	79	O	95	_	111	o	127	

### *Jeu de caractères Macintosh*

Mac OS a été le premier système à se doter d'un jeu de caractères plus complet. Il comprenait les 127 caractères de l'ASCII et 128 nouveaux.

Il aurait été plus simple que tous l'adoptent, sans doute trop simple. Les jeux de caractère des différents systèmes varient, bien que tous utilisent les 127 de l'ASCII.

0	16	32	space	48	0	64	@	80	P	96	`	112	p
1	17	33	!	49	1	65	A	81	Q	97	a	113	q
2	18	34	"	50	2	66	B	82	R	98	b	114	r
3	19	35	#	51	3	67	C	83	S	99	c	115	s
4	20	36	\$	52	4	68	D	84	T	100	d	116	t
5	21	37	%	53	5	69	E	85	U	101	e	117	u
6	22	38	&	54	6	70	F	86	V	102	f	118	v
7	23	39		55	7	71	G	87	W	103	g	119	w
8	24	40	(	56	8	72	H	88	X	104	h	120	x
9	tab	25	)	57	9	73	I	89	Y	105	i	121	y
10	lf	26	*	58	:	74	J	90	Z	106	j	122	z
11		27	+	59	;	75	K	91	[	107	k	123	{
12		28	,	60	<	76	L	92	\	108	l	124	
13	or	29	-	61	=	77	M	93	]	109	m	125	}
14		30	.	62	>	78	N	94	^	110	n	126	~
15		31	/	63	?	79	O	95	_	111	o	127	

128	Ä	144	ê	160	†	176	⊗	192	¿	208	-	224	‡	240	✦
129	Å	145	ë	161	°	177	±	193	¡	209	-	225	•	241	◊
130	Ç	146	í	162	¢	178	±	194	¬	210	“	226	„	242	Ó
131	É	147	ì	163	£	179	±	195	¬	211	”	227	„	243	Ô
132	Ë	148	î	164	§	180	¥	196	f	212	‘	228	‰	244	Û
133	Ï	149	ï	165	•	181	µ	197	≈	213	’	229	‰	245	Ü
134	Û	150	ñ	166	¶	182	ð	198	Δ	214	÷	230	È	246	ˆ
135	á	151	ó	167	ß	183	Σ	199	«	215	◊	231	Á	247	˜
136	à	152	ò	168	©	184	Π	200	»	216	ÿ	232	È	248	-
137	â	153	ô	169	®	185	π	201	…	217	ÿ	233	È	249	˘
138	ä	154	ö	170	™	186	∫	202	∞	218	/	234	í	250	˙
139	ã	155	õ	171	ˆ	187	∫	203	Å	219	◊	235	î	251	˚
140	å	156	ú	172	˜	188	∫	204	Å	220	<	236	ï	252	¸
141	ç	157	ù	173	∫	189	Ω	205	Û	221	>	237	î	253	˝
142	é	158	û	174	Æ	190	æ	206	œ	222	fi	238	ó	254	¸
143	è	159	ü	175	ø	191	ø	207	œ	223	fi	239	ô	255	¸

**Jeu de caractères ISO-LATIN-1**

Afin de résoudre les problèmes de conversion, un nouveau jeu de caractères s'est imposé comme standard : l'ISO-LATIN-1, ou ISO-8859-1

0	16	32	space	48	0	64	@	80	P	96	`	112	p
1	17	33	!	49	1	65	A	81	Q	97	a	113	q
2	18	34	"	50	2	66	B	82	R	98	b	114	r
3	19	35	#	51	3	67	C	83	S	99	c	115	s
4	20	36	\$	52	4	68	D	84	T	100	d	116	t
5	21	37	%	53	5	69	E	85	U	101	e	117	u
6	22	38	&	54	6	70	F	86	V	102	f	118	v
7	23	39		55	7	71	G	87	W	103	g	119	w
8	24	40	(	56	8	72	H	88	X	104	h	120	x
9	tab	41	)	57	9	73	I	89	Y	105	i	121	y
10	lf	42	+	58	:	74	J	90	Z	106	j	122	z
11		43	+	59	;	75	K	91	[	107	k	123	{
12		44	,	60	<	76	L	92	\	108	l	124	
13	or	45	-	61	=	77	M	93	]	109	m	125	}
14		46	.	62	>	78	N	94	^	110	n	126	~
15		47	/	63	?	79	O	95	_	111	o	127	

128	144	160	nbsp	176	∞	192	¿	208	-	224	‡	240	•
129	145	161	°	177	±	193	¡	209	—	225	‡	241	◊
130	146	162	¢	178	≤	194	¬	210	"	226	,	242	◊
131	147	163	£	179	≥	195	√	211	"	227	,	243	◊
132	148	164	§	180	¥	196	f	212	"	228	‰	244	◊
133	149	165	•	181	µ	197	≡	213	'	229	À	245	◊
134	150	166	¶	182	∂	198	Δ	214	+	230	Ê	246	◊
135	151	167	ß	183	Σ	199	«	215	◇	231	Ë	247	◊
136	152	168	®	184	Π	200	»	216	ÿ	232	Ë	248	◊
137	153	169	©	185	Π	201	...	217	ÿ	233	Ë	249	◊
138	154	170	™	186	∫	202		218	/	234	Ë	250	◊
139	155	171	ˆ	187	ˆ	203	À	219	◻	235	Ë	251	◊
140	156	172	˜	188	◊	204	Á	220	<	236	Ë	252	◊
141	157	173	≠	189	Ω	205	Ï	221	>	237	Ë	253	◊
142	158	174	Æ	190	æ	206	œ	222	fi	238	Ë	254	◊
143	159	175	Ø	191	ø	207	œ	223	fl	239	Ë	255	◊

Il existe naturellement d'autres jeux de caractères pour les autres familles de langues

ISO-8859-1 (Latin-1)	Europe Occidentale, Amériques
ISO-8859-2 (Latin-2)	Europe Centrale et Orientale
ISO-8859-3 (Latin-3)	Europe du sud-est
ISO-8859-4 (Latin-4)	Scandinavie, pays baltes
ISO-8859-5	Cyrillique
ISO-8859-6	Arabe
ISO-8859-7	Grec
ISO-8859-8	Hébreux
ISO-8859-9	Turc
EUC-JP ou Shift_JIS	Japonais

Deux ordinateurs utilisant deux systèmes différents peuvent donc s'entendre pour utiliser l'ISO-Latin-1 afin de communiquer. Ils peuvent le faire, mais encore parfois doit-on le leur commander.

Il est parfois plus sûr d'utiliser l'ASCII pour communiquer entre des systèmes différents. Les caractères qui n'en font pas partie (caractères spéciaux) peuvent être écrit en code à l'aide de l'ASCII. « é » s'écrit « &eacute; ». Le code s'ouvre toujours par « & » et se ferme par « ; ». Entre « & » et « ; » se trouvent des lettres, ou des chiffres précédés d'un dièse : le tiret long s'écrit « &#151; ».

Il n'est bien sûr pas question d'apprendre ce code par cœur. Des programmes se chargent très bien de la conversion. Tous les navigateurs et tous les traitements de texte sont capables d'afficher en ISO le code ASCII.

### ***Les jeux de caractères Unicode***

On a enfin les jeux de caractères Unicodes qui se déclinent en UFT-8, et UFT-16 pour les langues asiatiques à 16 bits. Le but de l'Unicode est d'offrir un ensemble de jeux de caractères pour toutes les langues du monde, y compris les plus inusitées sur le web.

L'Unicode simplifie beaucoup de choses, mais il faut bien reconnaître qu'il en complique aussi, puisqu'il s'agit bien d'un système d'encodage supplémentaire, et même double.

Tout fichier de texte contient dans son entête l'indication de son jeu de caractères sous le nom de « charset », et que le programme lira pour afficher la page.

Exemple d'une page web :

```
<meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
```

Que ce *charset* indique de l'ISO, comme ci-dessus, ou de l'UFT, il sera capable de lire du code ASCII, mais le charset ISO lira mal l'UTF, et inversement. Aussi, est-il toujours plus sûr d'encoder en ASCII.

## **LES FORMATS**

Ce qu'on appelle format est un concept spécifique de l'informatique et n'a rien à voir avec un format d'impression. Il désigne un système d'encodage des documents. Pour le texte, c'est tout l'encodage des attributs du texte, enrichissement, mise en page, etc.

Pour le seul texte, il existe une quantité impressionnante de formats.

On distingue les formats propriétaires des formats ouverts. Un format propriétaire est breveté, et il n'est interprétable que dans le logiciel qui le génère (en principe). Un format ouvert peut être connu de tous, et il peut être utilisé dans des logiciels différents (en principe).

Au début, avant qu'on n'utilise l'ordinateur autrement que comme une machine à écrire ou à classer des fiches, chaque traitement de texte avait un peu son propre format. Le programme était cependant fourni avec un certain nombre de filtres permettant de les convertir. Les programmes et leurs versions se multipliant, le nombre des filtres devenait si important que seuls les imprimeurs achetaient encore les mises-à-jour régulières des jeux de filtres.

Ces formats propriétaires sont aujourd'hui bien moins nombreux qu'il y a encore dix ans, et se limitent presque au seul fossile vivant qu'est la suite Microsoft Office avec son format au suffixe « .doc ».

Tous les textes se traitent aujourd'hui dans des formats relativement compatibles avec tous les traitements de texte.

- DOC
- RTF et RTFD
- XML et ODT
- TEX et LATEX
- PDF, PS et EPS
- HTML et XHTML
- TXT

Voici à peu près les seuls types de formats auxquels on peut avoir à faire, ce qui n'est déjà pas mal, mais relativement facile à embrasser quand même.

### ***Le format DOC***

Word, le traitement de texte de la suite Microsoft Office, ou son petit frère Works, sont des logiciels tellement utilisés que leur format fait figure de standard universel du texte. Rares sont ceux qui se demandent si ce format est bien compatible avec les autres, mais plutôt si les autres sont compatibles avec « .doc ».

En réalité, ce « .doc » cache plusieurs formats qui ne sont même pas si compatibles entre eux. Les dernières versions sont à base de XML entrecoupé de code Microsoft.

La plupart des ordinateurs sont vendus avec un système Microsoft, et donc l'un de ces deux programmes. Leur avantage est qu'on les trouve prêts à l'emploi dès la machine déballée, et que tout le monde aura le moyen de lire les documents qu'ils auront permis de produire. Les pages web s'afficheront bien dans Explorer, le navigateur le plus utilisé pour les mêmes raisons, et les courriels s'exporteront aisément dans Outlook, etc.

Tant qu'on ne sort pas de là, c'est en effet la solution de facilité. Les rares personnes qui n'utilisent pas Microsoft n'auront qu'à s'y adapter, et naturellement, elles ont bien eu le temps de le faire. C'est donc la solution la plus simple et la plus économique si l'on n'a pas à acheter le programme, et elle ne posera pas de problèmes particuliers dans les utilisations les plus communes. Elle pourra cependant montrer rapidement ses limites si on lui demande plus, et même si l'on s'avise qu'avec Word on peut aussi exporter dans d'autres formats.

### ***Les formats RTF et RTFD***

Le format de texte enrichi (*Rich Text Format*) est un format standard qui conserve (en principe) tous les enrichissements sur tous les logiciels qui traitent le texte. Le RTFD (*Rich Text with Graphic Documents*) en est une version améliorée pour gérer les objets graphiques.

Il pourrait donc être recommandé pour échanger le texte entre divers logiciels et divers systèmes. Le résultat sera acceptable pour peu que les deux traitements de texte soient capables de gérer les mêmes enrichissements. Même dans ce cas, il faudra compter aussi avec l'encodage des caractères si l'on passe d'un système à l'autre. Dans la pratique, l'usage du RTF pour collaborer ou pour sauvegarder peut se révéler décevant.

### Les formats XML et ODT

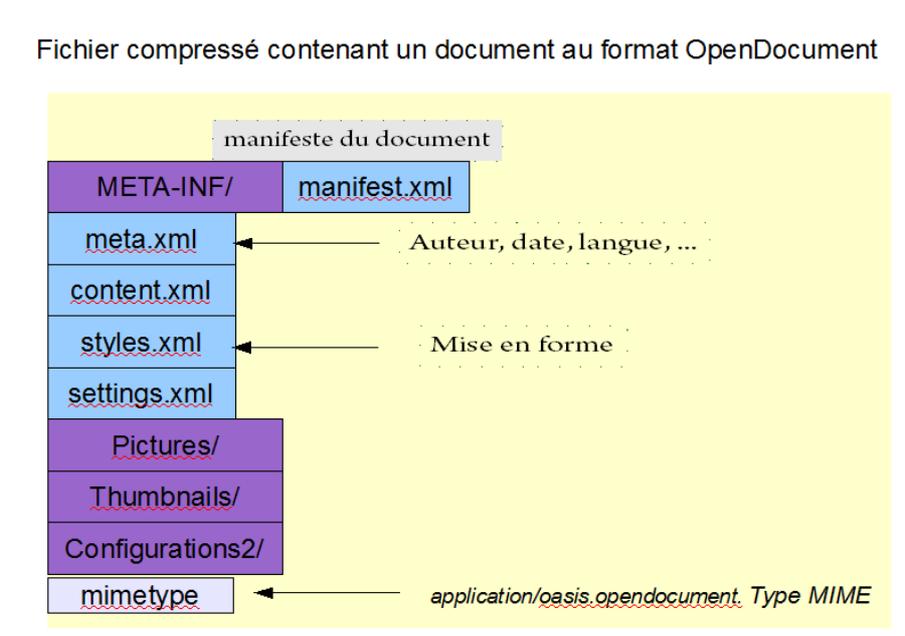
L'ODT (*Open Document Format*) n'est que du XML compressé. Si l'on remplace le suffixe « .odt » d'un fichier par « .zip », il se transforme en archive que l'on peut décompresser avec un logiciel de décompression, et il en sortira un fichier avec le suffixe « .xml ».

Le XML est un peu le successeur du SGML. Le SGML est un très ancien format de texte qui se voulait universel, mais n'a jamais rencontré beaucoup de succès, en partie pour des raisons de concurrence commerciale entre les traitements de texte de l'époque où l'on prenait les ordinateurs pour des machines à écrire, en partie parce qu'il était trop complexe à utiliser.

Le XML, après des débuts difficiles depuis qu'il a commencé à être développé en 1996 et rendu public en 1998, est en passe de devenir le format le plus standard de texte enrichi. Sous sa forme ODT, il est déjà le format par défaut d'Open Office et d'Abiword, et de plus en plus de logiciels l'importent ou l'exportent.

Comme le HTML fonctionne en association avec d'autres langages, CSS, Javascript, PHP... le XML s'associe avec une famille plus nombreuse encore : XSL, XSLT, DOM, SAX, PHP, ASP... Il existe aussi une forme hybride entre HTML ET XML, le XHTML, dont la vocation est de favoriser le passage de l'un à l'autre.

Le XML parvient à tenir ses promesses en passant d'un système à l'autre.



### Les formats TEX et LATEX

Le langage TeX permet de satisfaire à une typo et une mise en page très exigeante, sans se mettre sous la dépendance d'un langage propriétaire. Il existe des éditeurs TeX pour tous les systèmes, gratuits ou à des prix très modiques. Borges pour Linux, Texture pour Mac...

TeX a été développé par Donald Knuth dès 1979, principalement pour éditer des équations dans un format compatible avec tous les systèmes. Il travaille en effet à partir du format ASCII. Il est, naturellement, capable de gérer les jeux de caractères de toutes les langues, et des outils de correction linguistiques sont disponibles. Il est à noter que certains de ces outils, comme Excalibur,

fonctionnent aussi avec de nombreux traitements de textes sur tous les systèmes. TeX a été étendu et modifié par LaTeX dès 1980, développé par Leslie Lamport.

Bien sûr, le format TeX est orienté vers l'impression. Il serait naturellement voué à être converti en PDF ou en EPS, ou encore en HTML, ou en XML. Il n'est pas un format dédié à la diffusion.

### *Les formats PDF, PS et EPS*

L'EPS et le PDF sont des formats très particuliers. On s'en aperçoit d'abord en cherchant en vain sur la plupart des logiciels la possibilité d'enregistrer ou d'exporter en de tels formats, alors que la plupart des autres, en principe, sont proposés.

La conversion en PDF ou EPS se fait généralement par le gestionnaire d'impression. Pour simplifier, les fichiers sont enregistrés à partir des informations envoyées à l'imprimante.

C'est pourquoi de tels fichiers sont comparables à des clichés de la page. Aussi ce sont les meilleurs formats pour enregistrer les pages telles qu'elles sont composées afin de les archiver, les imprimer ou les diffuser, mais, comme avec un ouvrage imprimé, on ne pourra plus continuer à travailler sur de tels documents, ou très difficilement.

### *Les formats PostScript et EPS*

Le format PostScript est généré et lu par un pilote d'impression équipé d'un dispositif de rasterisation (transformation des primitives graphiques et bitmap à la résolution du dispositif d'impression : RIP, de l'anglais *ripping*). Il concerne essentiellement le transport des données vers l'imprimerie, et, de ce fait, commence à sortir des limites de mon propos.

L'EPS (*Encapsulated PostScript*) est une variante du PS qui contient un en-tête permettant l'interprétation du fichier par un logiciel de mise en page. De nombreux fichiers EPS sont disponibles sur le web, notamment en provenance des utilisateurs Linux. C'est une solution de transfert entre plates-formes. Les moyens ne manquent pas de convertir les fichiers EPS en PDF.

### *Le format PDF*

Le PDF (*Portable Document Format*) est un langage de description de pages créé par Adobe Systems pour l'impression. La spécificité du PDF est de préserver la mise en forme (polices, images, objets graphiques...) telle qu'elle a été définie par son auteur, et ce quelles que soient l'application et la plate-forme utilisées pour imprimer ou visualiser le fichier. Depuis plusieurs générations, les navigateurs sont capables d'ouvrir du PDF en ligne. On peut préférer le télécharger, et dans ce cas, il suffit de l'enregistrer. Le fait qu'on ne puisse que très difficilement retravailler un fichier PDF peut se révéler avantageux, ne serait-ce que pour prévenir les erreurs de manipulation. Le PDF est ainsi le meilleur format pour diffuser du texte sur l'internet, ou le transmettre pour l'impression.

Il existe beaucoup de méthodes et de logiciels pour exporter des fichiers en PDF. Toutefois, la quantité d'usages possibles et la quantité d'options pour optimiser les fichiers à ces usages peuvent transformer de telles opérations apparemment simple en casse-tête.

Le PDF permet soit d'incorporer, soit de substituer les polices par des polices PostScript ou Adobe. L'avantage de cette dernière solution est que le fichier peut devenir sensiblement plus léger, surtout s'il ne comporte que du texte. On doit dans ce cas choisir des polices qui ont leur équivalent pour les remplacer. Voici la liste des polices de substitution :

- AvantGarde, remplaçant pour Avant Garde.
- Bookman

- Courier, remplaçant pour Courier, Courier New et Monaco
- Helvetica, remplaçant pour Geneva, Helvetica, Arial et Verdana
- Helvetica-Condensed, remplaçant pour Helvetica Compressed
- Helvetica-Narrow, remplaçant pour Helvetica Narrow
- NewCenturySchlbk, remplaçant pour New Century Schlbk
- Palatino, remplaçant pour Palatino
- Symbol, remplaçant pour Symbol
- Times, remplaçant pour New York, Times et Times New Roman
- ZapfChancery-MediumItalic, remplaçant pour Zapf Chancery
- ZapfDingbats, remplaçant pour Zapf Dingbats
- HeiseiMin-W3 va remplacer la plupart des polices japonaises
- STSong-Light remplace toute police de script chinois simplifié
- MSung-Light remplace toute police de script chinois traditionnel
- HYSMyeongJo-Medium remplace les polices coréennes

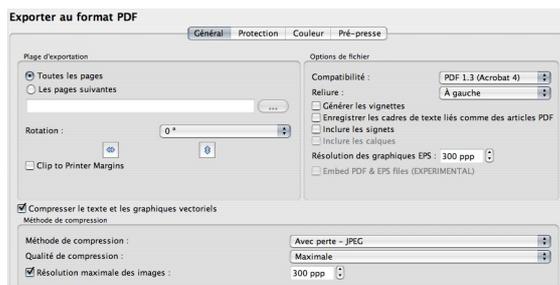
Cependant, celui qui ouvrira ce fichier devra lui-même posséder ces jeux de polices, ce qui est très probable, mais jamais certain. Il vaut donc mieux se limiter aux polices les plus courantes (Courier, Helvetica, Symbol, Times...) Cependant, même si les polices sont remplacées, le texte conservera la même mise en page, les mêmes enrichissements, la même disposition.

Si la vitesse de connexion et l'espace de stockage ne posent pas de problème, l'incorporation des polices est préférable, elle est même indispensable si le fichier est destiné à l'impression. C'est la solution qu'adoptent la plupart des logiciels pour exporter en PDF. Il est cependant utile de s'en assurer.

Au début, Adobe Acrobat, était le seul programme capable de générer du PDF, et Acrobat Reader, le seul capable de le lire. Les premières versions de PDF n'intégraient pas le support des hyperliens externes, réduisant son usage sur le web. Le poids du document PDF représentait un temps de téléchargement bien long, et le temps d'affichage des documents était lent sur les machines les moins puissantes. Les fichiers PDF étaient surtout utilisés dans les processus de publication assistée par ordinateur. Le PDF est devenu depuis le standard pour les documents imprimables sur l'internet, et pour l'imprimerie.

Bien d'autres programmes sont apparus aussi qui permettent d'afficher, de générer, de modifier, d'annoter ou de marquer, de découper ou de recomposer des fichiers PDF. Le PDF est un format ouvert, c'est-à-dire que ses spécifications sont connues et que son créateur Adobe Systems autorise des programmes tiers à réutiliser son format. Adobe détient un certain nombre de brevets, ils accordent une licence gratuite à tous les développeurs pour implémenter les fonctions de lecture et de création de PDF dans des logiciels tiers.

Plusieurs traitements de texte, comme Open Office, et d'autres logiciels, comme le programme d'images vectorielles Inkscape, ou celui de PAO, Scribus, permettent d'enregistrer au format PDF, avec de plus ou moins nombreuses options.



Le PDF continue d'évoluer, avec la sortie de nouvelles versions d'Adobe Acrobat. Il y a eu neuf versions de PDF correspondantes à celles du logiciel :

- (1993) - PDF 1.0 / Acrobat 1.0
- (1994) - PDF 1.1 / Acrobat 2.0
- (1996) - PDF 1.2 / Acrobat 3.0
- (1999) - PDF 1.3 / Acrobat 4.0
- (2001) - PDF 1.4 / Acrobat 5.0
- (2003) - PDF 1.5 / Acrobat 6.0
- (2005) - PDF 1.6 / Acrobat 7.0
- (2006) - PDF 1.7 / Acrobat 8.0
- (2008) - PDF 1.7, Adobe Extension Level 3 / Acrobat 9.0

Le format PDF est normalisé par l'organisation internationale de normalisation depuis 2008. Deux sous-ensembles du format PDF ont également été normalisés par l'ISO, il s'agit des formats PDF/A-1 (PDF for Archive, référencé par la norme ISO 19005-1) et PDF/X (PDF for eXchange). De plus, deux autres sous-ensembles sont actuellement en proposition pour devenir des normes ISO : PDF/E (PDF for Engineering) et PDF/UA (PDF for Universal Access).

Le format PDF est toujours plus orienté vers différents usages peu compatibles qui l'entraînent à bifurquer en plusieurs déclinaisons :

— Les documents destinés à être lus à l'écran, en ligne ou en local, et qui demandent au moins une barre de navigation et des liens actifs.

— L'imprimerie, ou encore des manuels livrés avec des applications et qui sont plus ou moins susceptibles d'être imprimés. Là, la taille des fichiers n'a pas ou peu d'importance, mais tous les enrichissements qui ne sont pas nécessaires à l'impression gagnent à être évités pour parer aux mauvaises surprises.

— Des formulaires à remplir ou encore des documents interactifs, gérant des vidéos, des liens internes et externes, des couches. Ils deviennent alors inexploitable pour l'imprimerie.

— Des documents protégés en lecture et en copies par des mots-de-passe et des filigranes pour servir le projet d'un marché de livres numérique.

Les options prédéfinies des différents programmes qui génèrent du PDF proposent généralement un ou des compromis raisonnables, mais ils sont souvent insuffisants pour des usages bien définis. Cette multiplicité d'usages pourrait à court-terme menacer le succès du PDF si elle n'est pas accompagnée de moyens simples d'en décider. Il commence en effet à devenir aujourd'hui relativement difficile à faire du PDF l'usage le plus simple, préparer du texte pour l'impression, au point qu'il est tentant de revenir à l'EPS.

### ***Les formats HTML et XHTML***

L'*Hyper Text Markup Language* est le format standard de l'internet. Il est le plus universel qui soit.

Lorsqu'il fut conçu, le HTML servait seulement à présenter du texte sur l'internet sans souci particulier de mise en page. Sous sa forme la plus simple, le HTML sert à présenter du texte brut indépendamment du codage particulier des caractères spéciaux propres aux différents systèmes.

Il sert aussi à créer des liens hypertexte qui permettent de naviguer dans une page ou d'une page à l'autre, et c'est pourquoi on appelle les logiciels destinés à le lire des navigateurs.

Le HTML permet de plus en plus des mises en page fines, et des enrichissements les plus divers. Pour cela, il s'associe à d'autres langages : CSS, Javascript, PHP...

Le XHTML, *eXtensible HyperText Markup Language*, est une sorte de croisement du HTML et du XML, qui se voudrait le futur langage du web. Mais le HTML 5 est déjà opérationnel au moment où j'écris ces lignes, et le développement du XHLK 2 a été abandonné.

### Le format TXT

Ou le format texte brut, est en tous points semblable à celui qu'on dactylographiait sur les anciennes machines à écrire (moins le texte souligné) : tout aligné à gauche, pas de justification, pas de changement de police, de taille ou de style, pas non plus d'espaces insécables. C'est bien suffisant pour écrire du courrier.

On s'en sert cependant surtout pour écrire du code. Tous les fichiers de code sont écrits en texte brut, mais leurs noms doivent être suivis du suffixe qui désigne le code pour être exécuté (« .jnlp » pour Java, « .py » pour Python, etc.)

Inversement, si l'on efface ce suffixe pour le remplacer par « .txt », le fichier s'ouvrira dans un éditeur de texte et le code s'affichera. On obtiendra les mêmes résultats avec des fichiers « .odt », « .doc » ou « .pdf ». Pour lire et écrire du code, on utilise un éditeur de texte, qu'on ne doit pas confondre avec un traitement de texte.

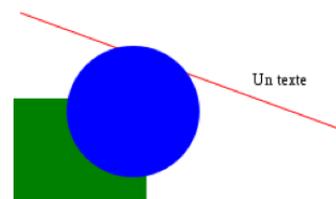
On peut donc en conclure que tous ces formats sont déjà des langages de programmation. C'est tout particulièrement vrai du HTML, dont l'usage est souvent reconnu comme la meilleure introduction à la programmation. C'est plus vrai encore pour le XML.

### Le SVG

*Scalable Vector Graphics* n'est pas un format de texte, mais d'image vectorielle. Il est conçu pour décrire des ensembles de graphiques vectoriels et basé sur le XML. Les coordonnées, dimensions et structures des objets vectoriels sont indiquées sous forme numérique dans le document XML. Un système spécifique de style (CSS ou XSL) permet d'indiquer les couleurs et les polices de caractères à utiliser.

```
<?xml version="1.0" encoding="utf-8"?>
<svg
  xmlns="http://www.w3.org/2000/svg"
  version="1.1"
  width="300"
  height="200">
  <title>Exemple simple de figure SVG</title>
  <desc>
    Cette figure est constituée d'un rectangle,
    d'un segment de droite et d'un cercle.
  </desc>
  <rect
    width="100" height="80"
    x="0" y="70"
    fill="green" />
  <line
    x1="5" y1="5"
    x2="250" y2="95"
    stroke="red" />
  <circle
    cx="90" cy="80"
    r="50"
    fill="blue" />
  <text x="180" y="60">
    Un texte
  </text>
</svg>
```

L'affichage graphique du code ci-contre donne ce résultat :



On distingue le rectangle vert de dimensions 100 x 80 pixels, le segment rouge, le cercle bleu et le texte « Un texte ».

Ce format gère des formes géométriques de base et des chemins (paths), qui utilisent les courbes de Bézier et permettent ainsi d'obtenir presque n'importe quelle forme.

Le SVG est très capable de numériser du texte, soit en incorporant les polices, soit en les transformant en chemins. Il n'est pas impensable qu'il devienne une alternative au PDF pour l'impression.



## ***LES SYSTÈMES ET LA NOTION DE TRANSPARENCE***

### ***Window et Unix***

Il existe plusieurs systèmes d'exploitation. On distingue traditionnellement Mac et PC, ce qui ne veut rien dire. Les Macintosh sont des PC, des ordinateurs personnels, comme les autres, si ce n'est qu'ils sont vendus avec un système d'exploitation Mac OS. Les autres PC sont quasiment toujours livrés avec un système Window. On ferait donc mieux de dire Mac ou Window, mais il existe encore bien d'autres systèmes, notamment Linux. Ces autres systèmes sont tous des déclinaisons d'Unix, et Mac OS aussi. On ferait donc mieux de dire Window et Unix.

En principe, tous les systèmes peuvent fonctionner sur toutes les machines. Ils leur sont indépendants, sauf Mac OS, qui est spécifiquement conçu pour les ordinateurs Macintosh et réciproquement.

L'idée de faire des ordinateurs personnels (PC) n'était pas évidente à l'origine. L'ordinateur était une affaire de spécialistes, d'ingénieurs, inutilisables par le premier venu. Faire des ordinateurs personnels impliquait de leur donner une interface intuitive, avec des fenêtres, des boutons et des icônes, permettant à Monsieur Jourdain de faire de la programmation sans le savoir. Les premiers PC furent des Apple où matériel et logiciel étaient conçus l'un pour l'autre. Quand d'autres fabricants se mirent à produire des PC, l'entreprise Microsoft obtint le monopole des systèmes d'exploitation avec Window.

L'interface Window a toujours eu une tête de retard sur l'interface Mac OS d'Apple. On s'en aperçoit en comparant de simples manuels d'installation : les instructions sont toujours plus courtes de quelques lignes dans la rubrique Macintosh, mais Window s'adapte à toutes les autres machines.

### ***Transparence et opacité***

Les interfaces Mac et Window sont conçues pour des utilisateurs individuels. Elles doivent en principe épargner toute aide extérieure au premier venu. Cet exploit repose sur une première couche

de l'interface très intuitive mais relativement opaque. Elle doit non seulement éviter à l'utilisateur de devoir passer à des couches plus profondes, mais aussi l'empêcher d'y tomber malgré lui, pour qu'il n'y fasse pas des catastrophes. Cet avantage pour le plus grand nombre devient un handicap pour l'informaticien éclairé, et surtout pour des structures de travail qui disposent de leur propre service informatique interne ou externe. Dans ce cas, le système doit être plus transparent, et comme les machines sont plus ou moins en réseau et accessibles à plusieurs, les programmes et les données doivent être protégées par des mots-de-passe et des droits d'administration. Les systèmes Window sont donc plus faciles à utiliser, mais en contrepartie plus fermés, plus opaques. Il en va de même avec les logiciels commerciaux qui leur sont compatibles. Cette fermeture favorise évidemment la vente de licences d'utilisation.

Les systèmes Unix favorisent plutôt l'offre de services. Il ne suffit pas d'avoir le droit d'explorer et de modifier des programmes ; on doit encore en être capable, et surtout en avoir le temps. Il peut être alors tout aussi commercial et bien plus profitable pour tous de laisser à quiconque le droit d'améliorer, de corriger et d'innover. Les deux types de systèmes demeurent concurrents puisqu'ils ont chacun leurs avantages et leurs inconvénients opposés. Chacun essaye bien sûr de minimiser les inconvénients et d'accroître les avantages. En fait, c'est plutôt du côté Unix qu'on y réussit. Les systèmes Linux deviennent toujours plus simples à utiliser, ainsi que les logiciels Open Source. Toutefois, une certaine difficulté est inhérente aux avantages d'Unix. Plus de possibilités et plus de simplicité d'utilisation sont des objectifs difficilement conciliables.

Bien sûr, l'opacité protège la propriété ; on ne peut pas toucher à un code qu'on ne voit pas. Elle protège aussi surtout, on doit bien le dire, l'utilisateur de trop de complications. Nous avons de bonnes raisons d'accéder au code d'une page HTML, mais quand nous la lisons, nous ne souhaitons pas devoir le déchiffrer nous-même. Nous ne désirons la transparence dans la seule mesure où nous avons aussi la possibilité de la masquer. Elle ne serait sinon qu'une gêne.



*Nova, une distribution Linux sur une base Gentoo, est développée à Cuba par l'Université des sciences informatiques.*

### ***Les systèmes et les programmes***

Un système d'exploitation n'a d'intérêt que pour les programmes qu'il fait tourner. Un logiciel est d'abord programmé pour un système d'exploitation particulier, on le dit alors « natif ». On peut en faire des versions pour d'autres systèmes, on parle alors de « portage », et on le dit « porté ». La

plupart des logiciels natifs Linux sont portés sur Mac et sur Window. Il est rare qu'un logiciel commercial soit porté sur Linux. Cela tient largement à une philosophie différente de la programmation.

Le caractère principal de l'*open source* est souvent masqué par l'aspect juridique qui l'accompagne. Les programmes *open source* peuvent être librement modifiés et distribués. Par voie de conséquence, ils sont souvent gratuits, mais pas toujours. Le plus important n'est pas la gratuité, ni le droit de modifier. Le plus important est la possibilité réelle. On général, les trois vont ensemble pour des raisons assez logiques, mais pas toujours. Des logiciels commerciaux offrent de réelles possibilités d'adaptation bien documentées (comme BBEdit, Komodo, Nisus), et il n'est pas rare que les utilisateurs coopèrent avec les producteurs pour les améliorer, alors que des logiciels formellement libres demeurent parfois non-modifiés, et non-modifiables à cause de leur programmation opaque et de leur manque de documentation (comme HyperEngine).

C'est donc bien en réalité la façon de programmer qui fait la différence. C'est elle qui détermine en définitive la licence d'utilisation, et finalement le coût de développement. Le résultat paradoxal est qu'en matière de programmation le prix est souvent inversement proportionnel à la qualité du logiciel. Les logiciels, en tout cas, demeurent déterminés par le système pour lequel ils ont été programmés, leur environnement natif, Window ou Unix, Sun ou Mac.

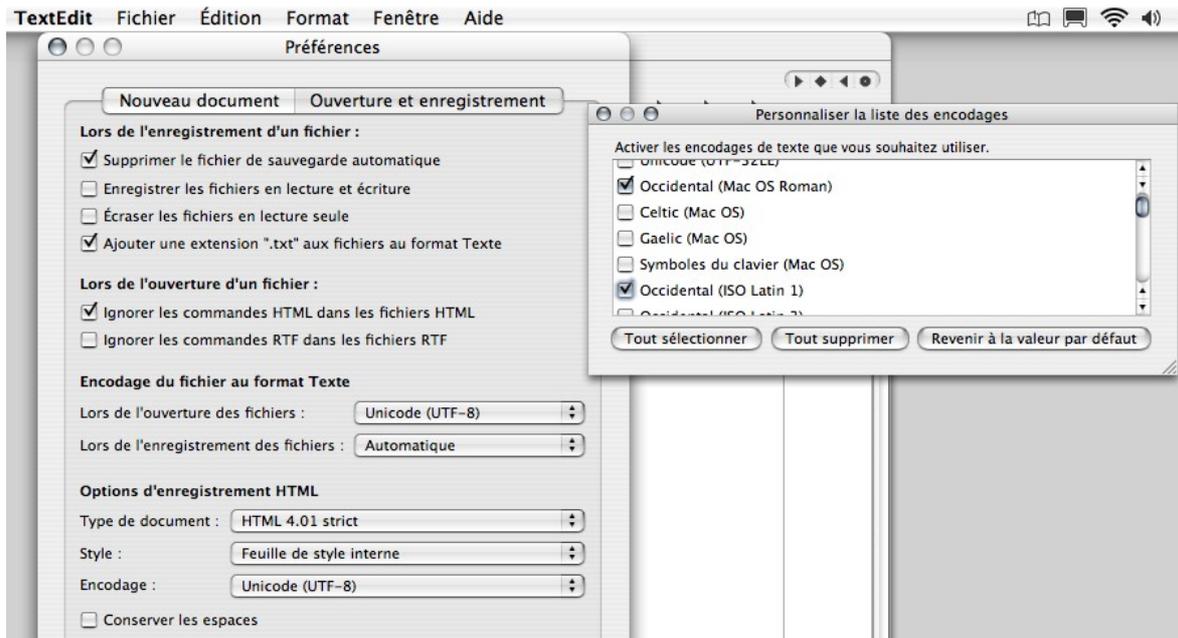
Les logiciels *open source* sont souvent plus puissants que leurs concurrents commerciaux, mais ils demandent aussi des aptitudes peu communes pour tirer tout le parti de cette puissance. Pour des usages standards, les produits commerciaux sont souvent plus simples à utiliser et mieux documentés. Ils sont faits par des professionnels pour des utilisateurs, alors que les premiers sont faits par des professionnels et des utilisateurs entre eux. Les logiciels *open source* fonctionnent très bien les uns avec les autres ; les logiciels commerciaux aussi. Il est parfois plus problématique de passer des uns aux autres, notamment pour des questions de formats et de conversion.

### Mac OS

Mac OS est un système à part dans la mesure où il est un Unix et où il appartient à la société Apple, comme Window à Microsoft. D'autre part, il est dédié aux seules machines Apple.

Il est plus simple et intuitif que Window, les principaux logiciels commerciaux ont des portages Mac, les logiciels *open source* s'y installent très simplement quand ils ont un portage, et ils peuvent sinon y fonctionner quand même au prix de quelques installations supplémentaires. Il ignore les virus, ce fléau de Window. Il est donc particulièrement adapté au néophyte. Il est à peu près impossible de quitter la première couche de l'interface sans le faire exprès, et rien n'y oblige, et il n'y a pas d'obstacle particulier pour y parvenir si on le désire. Il est donc aussi adapté au hacker. Les Mac sont souvent présentés comme des ordinateurs ludiques, mais comme ils gèrent facilement des réseaux et des droits d'administration, ils sont aussi d'excellentes machines pour une entreprise ou toute structure de travail.

Mac OS a aussi ses logiciels natifs. Certains sont livrés avec le système. Ils sont aussi extrêmement simples d'usage, mais ils ne sont pas pour autant des jouets. Ils peuvent permettre de créer des documents de qualité professionnelle simples. Certains autres sont payants et constituent des alternatives très économiques aux autres logiciels commerciaux. D'autres encore n'appartiennent pas à Apple et peuvent être en *open source*.



TextEdit, le petit traitement de texte offert avec Mac OS, aux fonctions les plus sommaires, peut se transformer en un éditeur de texte, très basique aussi, mais offrant des options de réglage très avancées.

Profitant des qualités du système, ce sont des outils très intuitifs, performants et très proprement programmés. De plus, Mac OS se tournant de plus en plus vers les formats libres, l'intégration avec les produits Linux est toujours meilleure. Ceux-ci ont aussi de plus en plus de versions natives exploitant toutes les ressources propres à Mac OS. C'est donc le système idéal pour travailler entre les deux environnements Unix et Window.

## ***LES PROGRAMMES***

### ***Programmes, formats et langages***

Si l'on va au bout de ce qu'implique la dernière conclusion, on admettra qu'il n'y a pas une limite bien nette entre programmer et utiliser un programme. Quand on met en page un texte, on écrit du code comme Monsieur Jourdan faisait de la prose sans le savoir, même si c'est en réalité le programme qui l'écrit pour nous. Selon le format (le langage) que nous employons, il nous est plus ou moins possible, facile, ou recommandé de le modifier, voire d'entrer directement le code. C'est plutôt recommandé avec le HTML, TEX ou LATEX, c'est possible et ce peut être utile avec le XML. C'est bien moins évident avec les autres, du moins sans conversion préalable, c'est-à-dire sans les traduire en de nouveaux langages.

### ***Les outils***

Les programmes, nous les appelons des outils, des outils logiciels, que nous ne confondons pas avec des outils matériels. Nous les appelons des outils pour la simple raison qu'ils nous servent à

accomplir des tâches différentes, ou encore à les accomplir différemment. Nous n'utiliserons par exemple jamais un traitement de texte pour éditer du code.

Certains de ces outils sont très simples et n'ont qu'un usage très limité, d'autres sont complexes, sont des agglomérations d'outils, de véritables machines, des usines, des « usines à gaz » comme on dit. Pour le texte, c'est le cas d'Open Office, qui peut à peu près tout faire, ou, pour l'image, du GIMP (*GNU Image Manipulation Program*), dont un seul de ses dizaines de filtres peut être un logiciel complexe.

### *Le choix des programmes*

Ces programmes, nous les choisirons et les penserons d'après les tâches qu'ils sont capables d'accomplir, mais aussi, et parce que c'est fondamentalement la même chose, selon les formats, les langages qu'ils sont capables de gérer. Nous songerons aux langages avec lesquels ils travaillent, dans lesquels ils exportent et ceux qu'ils sont capables d'ouvrir.

Lorsqu'on voudra éditer du code en ASCII, on utilisera un éditeur de texte, et rien d'autre. Pour du son, un éditeur de son ; de l'image, un éditeur d'image ; pour naviguer, un navigateur bien sûr, et pour héberger son courrier, un client FTP, etc.

Le plus dur, ce n'est pas d'apprendre à utiliser un programme. Le plus dur, c'est de passer d'un programme à l'autre, et donc d'un format à l'autre, d'un protocole à l'autre, car il est totalement impossible de réaliser la moindre chose à peu près complètement avec un seul logiciel, un seul format, un seul langage, un seul protocole — et tout d'abord de faire un site.

### *La portabilité*

Lorsqu'on écrit avec un ordinateur, on utilise un traitement de texte. Si l'on a bien suivi ce qui précède, on comprendra qu'on travaillera alors avec les formats XML ou ODT, RTF ou RTFD, TeX ou LaTeX, ou DOC (et avec des jeux de caractère ISO-8859-x ou Unicode). Oublions maintenant TeX et LaTeX qui sont un peu compliqués et ne retenons que les trois ou cinq autres.

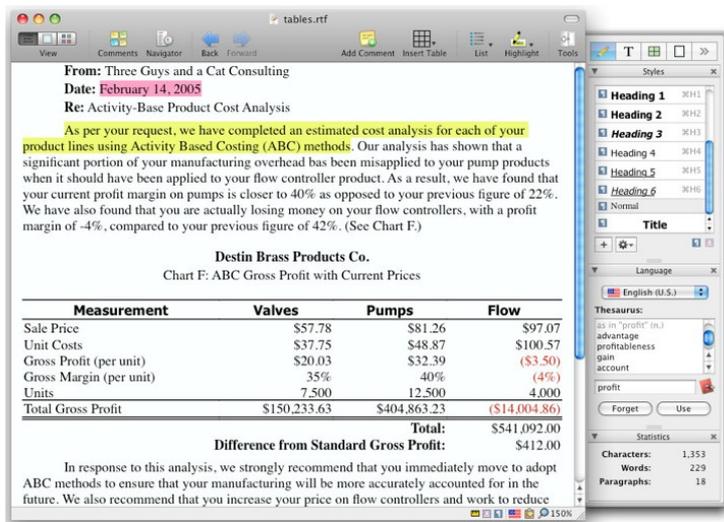
On aura compris aussi qu'il est plus judicieux de choisir d'abord le format, et ensuite le traitement de texte en conséquence, car ce traitement de texte nous imposera son format ou nous en laissera le choix. Chaque traitement de texte a un format natif dans lequel il est plus performant.

Il y a encore bien d'autres critères pour choisir un traitement de texte. Comme tout ce qu'on fait dans un format devra tôt ou tard être exporté dans un autre, on sera attentif aussi à la façon dont l'application gère d'autres formats, comment il les ouvre, les importe, les enregistre et les exporte. On appelle cela la portabilité. On pourra vérifier s'il permet le choix entre ISO-8859-x ou Unicode. On se souciera aussi de toutes ses fonctionnalités et de son ergonomie.

### *L'ergonomie*

L'importance des fonctions dépend de ce qu'on fait ; l'ergonomie, elle, n'est pas relative ; elle est importante quel que soit l'usage. Qu'on écrive de très longs textes ou beaucoup de courrier, qu'on vise l'impression ou l'édition en ligne, les fractions de secondes qu'on gagnera à utiliser un bouton accessible ou un raccourci clavier plutôt que d'ouvrir des boîtes de dialogue et dérouler des menus, finiront par jouer un grand rôle sur notre temps de travail, notre confort et notre concentration.

Les logiciels les plus ergonomiques ont la mauvaise habitude de ne fonctionner que sur Mac OS. Ils vont du simplicisme TextEdit au très complet Nisus Writer Pro, en passant par l'élégant Bean.



Nisus Writer Pro est très complet pour écrire, mais il ne fait que ça, et l'on aura assurément besoin d'autres outils pour des mises-en-page destinées à l'impression, pour éditer le texte en ligne, pour faire des PDF interactifs, etc. mais il exporte et importe très bien le format ODT.

### Les fonctionnalités

L'importance des fonctionnalités varie avec l'usage qu'on a d'un traitement de texte. Si l'on n'utilise que des langues occidentales, on n'aura peut-être pas besoin d'un logiciel qui gère des écritures de droite à gauche à la volée. Si l'on n'écrit que du courrier ou de brefs articles, on n'aura pas la nécessité de créer des tables des matières, des index ou des sections. Si l'on écrit des romans, on n'aura peut-être pas besoin d'insérer des graphiques ou des tableaux, etc.

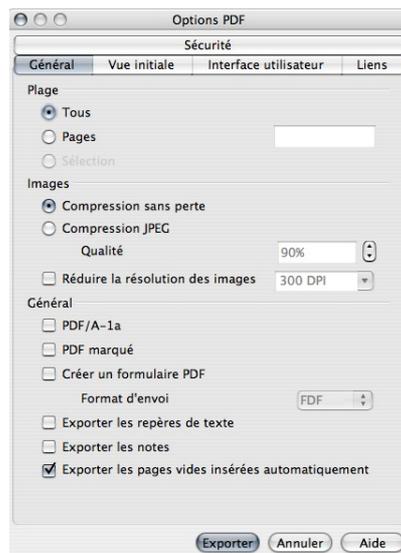
Un certain nombre de fonctions peuvent être fournies par le système et des logiciels tiers : correction orthographique et grammaticale, prononciation du texte, exportation en PDF, ou en HTML, décompte des mots et des caractères, etc.

Le minimum qu'on puisse attendre d'un traitement de texte est : la gestion de feuilles de styles (avec changement de polices et de tailles, alignements, espaces entre les lignes et entre les paragraphes), l'insertion de sections ou de styles de page, l'insertion de notes, des fonctions de recherche et de remplacement, la gestion d'en-têtes et de bas de page, la numérotation automatique, le saut de page...

D'un autre côté, moins on exige de fonctions pour son traitement de texte, plus on doit être attentif à ses capacités d'exporter en d'autres formats, puisqu'on sera conduit à utiliser d'autres logiciels.

Le logiciel le plus complet est Open Office. On peut presque tout faire avec lui, de la PAO à l'édition en ligne complète. Son module d'exportation en PDF est très riche, et son code HTML correct si la mise en page a été faite dans les règles de l'art.

Il est multilingue, crée des tables des matières et des index, édite des formules, gère les images, les retouches, transforme le bitmap en vectoriel etc. Il est possible de le compléter avec des modules : correction grammaticale, éditeur de formules perfectionné, ouverture et modification de PDF, etc. Il offre un choix complet de formats.



Sur les formats et les fonctions, c'est le meilleur, mais pour ce qui est de l'ergonomie, il est pour le moins perfectible.

Dans une moindre mesure, Microsoft Office offre aussi énormément de fonctions, mais, sans être supérieur pour l'ergonomie, il est déplorable pour les formats.

## ***LA RÉVOLUTION DE L'ÉCRIT***

On pourra tirer de ce qui précède la conclusion qu'écrire avec un ordinateur change profondément la nature de l'écrit. L'ordinateur personnel est bien plus qu'une machine à écrire perfectionnée, enchâssé entre ces deux techniques que sont, en amont, la numérisation, et en aval, l'internet. Cette triple invention a des conséquences à très longue portée.

### ***L'œuvre en procès***

Elle balaie les étapes successives dans lesquelles l'imprimerie avait enfermé l'écriture. Elle met d'abord à mal, comme on l'a déjà vu, la stricte distinction entre la correspondance privée et l'édition publique.

Dans sa généralité, « le public » est renvoyé à une pure abstraction par la réalité en acte de la connexion. Le courriel, lui, peut aisément être partagé dans des listes privées ou publiques. Il peut aussi très simplement être publié en ligne. Le courriel porte des liens actifs, qui ouvrent immédiatement des publications ; les publications comportent des liens pour correspondre tout aussi immédiatement et sans médiation avec leurs auteurs, ou éventuellement pour laisser des commentaires en ligne.

Les moments strictement séparés qu'avaient été l'écriture, la saisie, la lecture en comités, l'édition, l'impression, la lecture publique, la critique, etc. ne le sont plus. Si « le véritable texte » est devenu le fichier numérique, il est édité en même temps qu'écrit, et en conséquence immédiatement publiable, critiquable, modifiable...

Le point nodal de l'écrit qu'était *l'édition originale* est remplacé par ce nouveau qui est *la dernière version*. Tout travail est virtuellement *en cours* ; et un ouvrage, *collaboratif*.

### *Concilier écriture édition et lecture*

Écrire, éditer, publier, ce sont là trois termes distincts qui doivent d'abord être clarifiés. On le remarquera, c'est généralement sous le terme de « bureautique » qu'on présente les logiciels de traitement de texte. Or, chacun le sait, faire de la bureautique, ce n'est ni écrire, ni éditer, ni publier.

— Écrire, cela suppose d'abord de ne pas s'égarer à travers des dizaines de pages, voire des centaines, des milliers, qui bien souvent ne sont pas seulement celles de l'ouvrage que l'on écrit, mais aussi de ceux que l'on consulte, de ses notes, ses documentations. Cela se pratique généralement dans un espace relativement limité, et l'on n'y parvient pas sans méthode. Chacun sait combien on peut facilement se perdre parfois dans une seule phrase.

L'ordinateur peut nous aider dans ce travail. C'est même à quoi il sert. D'un autre côté, il nous complique aussi la tâche. Un écran normal ne permet pas d'afficher lisiblement une page A4 entière, alors qu'on peut déjà étaler plusieurs feuilles ou plusieurs livres sur une table de bar. D'autre part, il nous prive du sens du toucher, si précieux pour éprouver l'épaisseur de pages brochées.

Naturellement, on aurait tort de se priver d'écrire à la plume, et d'imprimer autant de fois qu'il est nécessaire. On découvre vite alors que le principal problème consiste à s'assurer qu'un même tirage correspond bien au même document numérique de travail. De tels problèmes ne se posaient pas à l'époque du manuscrit et du tapuscrit.

— Éditer a donc pris un sens nouveau. L'édition est devenue un moment de l'écriture. Nous rééditons perpétuellement notre ouvrage pendant que nous l'écrivons. Nous l'éditions, même si nous ne cherchons pas (encore) à le rendre public, le publier.

— Publier prend alors aussi un sens nouveau. Certes, il s'agit toujours de rendre public un écrit, mais la question prend ici le sens de l'éditer dans un format public. L'intérêt n'est pas alors seulement d'offrir un texte à la lecture publique, mais aussi, pour l'auteur seul, de s'assurer qu'il y accédera encore à l'aide d'une autre application, d'un autre système, d'une autre machine.

### *Éditabilité, lisibilité*

La première chose que chacun a découvert dans un traitement de texte a été la possibilité d'afficher ou de masquer les caractères invisibles. Peut-être en avons-nous été surpris la première fois, et peut-être aussi n'avons-nous vu d'abord dans cet affichage qu'une gêne.

Il est à priori contradictoire qu'un caractère soit invisible. L'invisibilité semble contradictoire avec l'essence d'un caractère qui existe pour être lu, donc vu. Ces caractères dits « invisibles » ne le sont d'ailleurs pas tant que ça. Ce sont des espaces ; des espaces entre des caractères, ceux que laisse un saut de ligne, un saut de paragraphe (avec son alinéa par exemple). L'espace vide a donc valeur de caractère, de ponctuation disons. Ces espaces peuvent donc s'afficher sous la forme de caractères éditables.

Ces caractères invisibles assurément gênent notre lecture. C'est pourquoi on peut les masquer, mais ils nous facilitent certainement le processus d'édition. Dès qu'on l'a compris, on ne peut plus s'en passer.

### *Le marbre liquide*

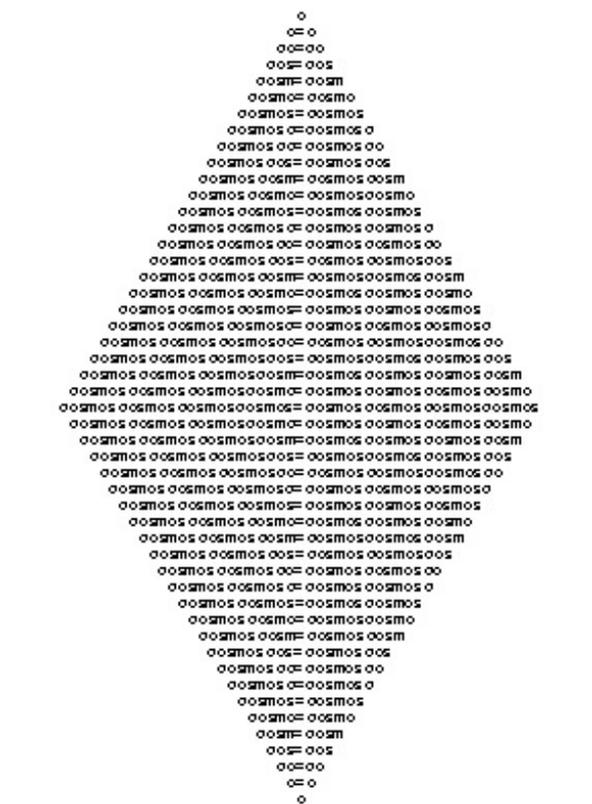
Cette fonction des traitements de texte nous montrent deux exigences contradictoires. Nous voulons d'un côté qu'un texte demeure immuable et aussi incorruptible que s'il était gravé dans le marbre. D'un autre côté, nous le voulons toujours modifiable, perfectible, rééditable. Nous ne voulons pas recommencer le travail que nous avons effectué sur un texte pour l'envoyer à l'impression, quand nous entreprenons ensuite de le mettre en ligne, ou inversement.

Nous voulons que nos textes demeurent aussi incorruptibles que s'ils avaient été gravés dans du marbre, et aussi fluides qu'un liquide qui épouse les formes du récipient dans lequel on le verse. Nous recherchons en somme du marbre liquide, et nous savons bien que ça n'existe pas.

Ces deux exigences sont diamétralement contradictoires, mais si nous les comprenons bien elles se rejoignent quand même sur certains points. Le plus important est de se faire déjà une idée plus précise de ce qui ne doit en aucun cas bouger, et de ce qu'on souhaite laisser fluide. On pourrait encore dire autrement : jusqu'où notre texte demeure le même ; à partir de quand il est corrompu ?

Il n'est pas essentiel, par exemple, que le texte que je suis en train d'écrire conserve ses polices, leurs tailles ou son format de page, tant du moins que les titres, les sous-titres, les paragraphes, les illustrations et les sauts de pages conservent leurs relations réciproques ; tant disons qu'il garde sa structure. Cela, un traitement de texte permet de le contrôler par des styles.

Il est au contraire des textes pour lesquels ce ne serait pas le cas. Un texte à chasse fixe (c'est-à-dire dont les caractères ont le même espacement) pourrait ne plus être lisible si l'on change les polices.



Tout ceci se contrôle par des feuilles de style, et nous verrons plus loin l'importance qu'elles ont avec le HTML et les navigateurs.

Malgré tout, nos désirs demeurent contradictoires. Si nous envoyons un fichier à l'imprimeur, ou si nous le mettons en ligne pour l'imprimer, nous exigeons que rien ne bouge. Nous connaissons déjà des formats qui le laissent espérer, comme le PDF ou l'EPS. Il serait alors logique que nous renoncions à le retoucher, et que nous admettions au moins que ce ne soit pas facile.

Si nous souhaitons au contraire publier un texte en ligne ou sur du papier, nous souhaitons plutôt le voir couler dans les cadres que nous lui réservons, changer éventuellement ses polices, leurs tailles, les espaces entre les lignes, entre les paragraphes, la tailles des alinéas, la longueur des

lignes, la largeur des marges, bref, tout ce qui peut changer quand on édite un texte sans qu'on puisse dire que le texte ne change. Et nous pouvons même souhaiter pouvoir le corriger, en modifier le contenu, sans que sa structure ne se corrompe. À cela aussi des feuilles de style nous aident, mais à la condition de résoudre les problèmes de portabilité.

### *La réédition permanente*

#### *Écrire dans la durée*

Nous avons déjà vu que les meilleurs formats pour écrire et éditer du texte sont le XML ou l'ODT; dans une moindre mesure le RTF et RTFD. Mais le format le plus portable est le HTML, tout particulièrement si nous lui associons des feuilles de style externes (CSS, *Cascading Style Sheet*) qui permettent de s'occuper séparément du contenu et de sa structure.

Si ce que nous recherchons est si contradictoire, il vaut donc mieux lui trouver des solutions distinctes. On sauvegardera son travail dans le format de son traitement de texte (XML ou ODT de préférence), on l'exportera en PDF pour qu'il soit possible de l'imprimer ou que ce soit sans risque de le corrompre, et on l'exportera en HTML pour être sûr qu'il soit parfaitement éditable d'où que ce soit.

Ces questions prennent une importance bien plus considérable qu'on peut l'imaginer sur la longue durée. La nouveauté du numérique et de l'internet nous font souvent oublier cette longue durée. Elle peut se rappeler à nous de façon cruelle ; bien des gens ont perdu une grande quantité de leur travail, et dans le meilleur des cas, beaucoup de temps.

L'internet nous offre une capacité de travail démesurée, il nous permet de produire et d'échanger des masses gigantesques de documents, de collaborer par-delà les distances dans les plus brefs délais et sans peine, il offre des réseaux de collaborations sans limites et des possibilités de réunions presque instantanées. Il donne au premier venu une puissance de travail intellectuel qu'aucun chercheur, aucun leader, n'avait jamais osé imaginer avant. Combien serait-il bête d'ignorer la durée et de laisser cette puissance s'évanouir dans la fugacité de moments disjoints !

Écrire est de toute façon une activité qui se place dans la durée.

#### *La puissance du texte*

Il importe de voir toutes ces possibilités nouvelles en même temps que les constantes de l'écriture. — Sur quoi repose la puissance du langage écrit ? — Il se conserve, et par voie de conséquence, il peut plus aisément être transmis. — Sans doute, mais avant même cela ? — Il peut être parcouru, navigué ; et il libère la pensée d'un cours à sens unique. On peut remonter ce cours, le reconstituer, le changer à tout moment.

Oui, c'est en effet ce dernier point qui fait toute la puissance du langage écrit, du texte.

Le texte en langue naturelle possède à peu près toutes les propriétés de la parole, il peut à chaque instant devenir parole, si on le prononce ou le fait prononcer par un programme. Écrite, la parole devient modifiable. Elle laisse à tout moment la possibilité de la corriger, la modifier, la recomposer, la traduire, la paraphraser, la développer, la compacter...

Cette propriété de l'écriture est plus visible encore dans le langage mathématique. Là, la puissance de l'écriture est plus saisissante, dans la mesure où le langage mathématique semble faire les raisonnements à notre place.

Le texte en langue naturelle décuple aussi nos capacités d'inférences, et cela d'une façon à peu près identique à celle de l'écriture mathématique, ou encore de la programmation : en donnant la possibilité, perpétuelle, de relire, de corriger, de modifier, de déplacer, de condenser, d'ajouter, de reformuler, de traduire... Il est probable qu'on ait souvent confondu cette puissance inhérente au langage et à l'écriture avec l'inspiration dont on ne saurait dire d'où elle vient.

Les conditions *sine qua non* de cette puissance de l'écriture sont, naturellement, que le texte d'abord se conserve suffisamment longtemps, et surtout qu'il demeure le plus aisément possible modifiable.

L'invention du papier, en concurrence à tout autre support, puis l'invention du livre, en concurrence au rouleau, furent des progrès incontestables ; mais l'imprimerie fut plutôt un recul envers cette possibilité de modifier et de rééditer à tout moment. Le numérique en lève les limites sans qu'il soit nécessaire de renoncer à ses avantages.

De tout ce qui précède, on peut déjà commencer à bien entrevoir les raisons pour lesquelles on fait un site.



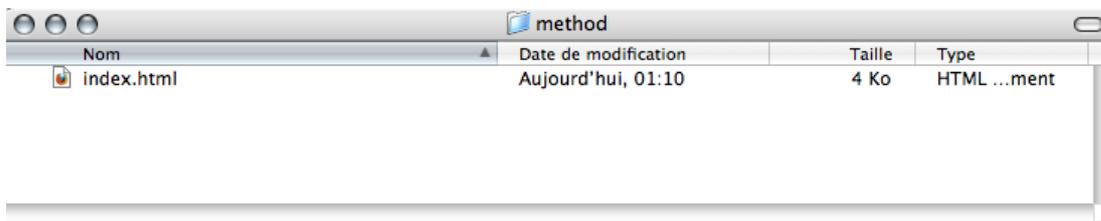
# Ce qu'est un site

## COMMENT CRÉE-T-ON UN SITE ?

### De quoi il est question

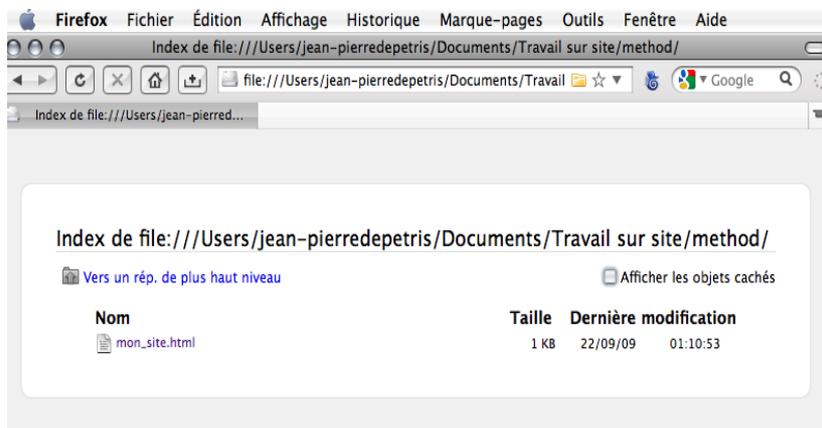
Observons tout de suite que la première question à se poser n'est peut-être pas comment on crée un site. On pourrait commencer par en soulever bien d'autres, notamment : Pourquoi ? Eh bien non, « comment » est plus simple, et aidera déjà à trouver des réponses aux autres questions.

Un site, c'est au moins une page, c'est-à-dire, on l'a vu, un fichier HTML. Ce fichier s'appellera « index.html » et pas autrement (sauf indication expresse de l'hébergeur), et ce fichier se trouvera à la racine du site, c'est-à-dire au premier niveau.



Ce fichier s'affichera dès qu'on appellera l'adresse du site dans le navigateur, par exemple « machin.fr ».

Si le fichier ne s'appelait pas « index.html », il ne s'ouvrirait pas. Il s'afficherait dans la fenêtre du navigateur avec son icône et son nom, comme dans la fenêtre d'un dossier (mais il s'affiche cependant si l'on y clique dessus).



Ce fichier a une certaine importance, puisqu'il est l'entrée du site, donc la première image qu'on en verra, si l'on entre toutefois par l'entrée, et dans tous les cas, celle où l'on aura tendance à aller quand on cherche quelque chose. Ce fichier devra donc conduire vers les autres pages de notre site — s'il en a d'autres — ou vers ses autres départements — si c'est un grand site.

Il est en quelque sorte l'équivalent de l'éditorial et du sommaire pour un journal, de la préface et de la table des matières pour un livre. Un site peut toutefois ne contenir qu'une page. Dans ce cas, elle débutera par l'éditorial et le sommaire de son contenu.

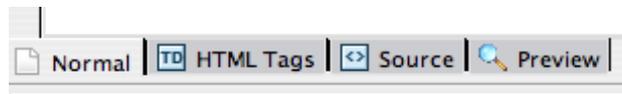
## Deux méthodes

### L'éditeur de site

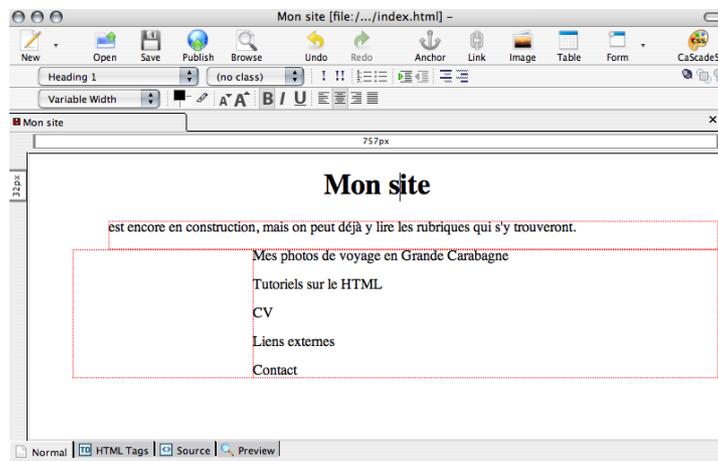
Comment s'y prend-on pour réaliser une telle page ? Il y a deux méthodes.

La première consiste à l'écrire directement dans un éditeur de site wysiwyg (abréviation de *what you see is what you get*) qui nous permet de travailler le HTML comme il s'affiche dans la page du navigateur. Un tel programme est pour l'édition en ligne l'équivalent d'un logiciel de PAO pour l'imprimé. Il est seulement moins précis et contient moins d'outils pour la mise en page, pour la raison évidente que la page s'affichera sur d'autres machines et d'autres écrans et qu'elle doit donc conserver une certaine élasticité. Il existe de nombreux programmes, libres ou commerciaux, dont les plus connus sont Kompozer pour les premiers et Deamweaver pour les seconds.

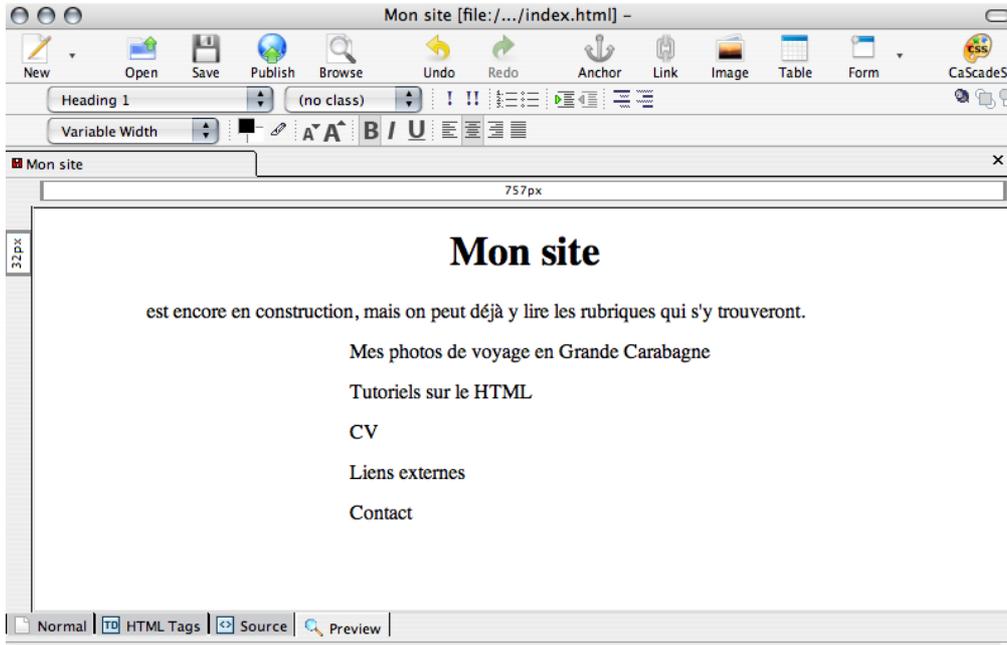
Cette méthode est la meilleure pour un débutant. En effet, ces logiciels permettent toujours d'afficher la page sous quatre formes : Normal (celle par défaut), Balises HTML, Code source et Aperçu. On peut naturellement travailler dans ces quatre formes d'affichage. Il est probable qu'au début seuls le premier et le dernier onglets seront utilisés.



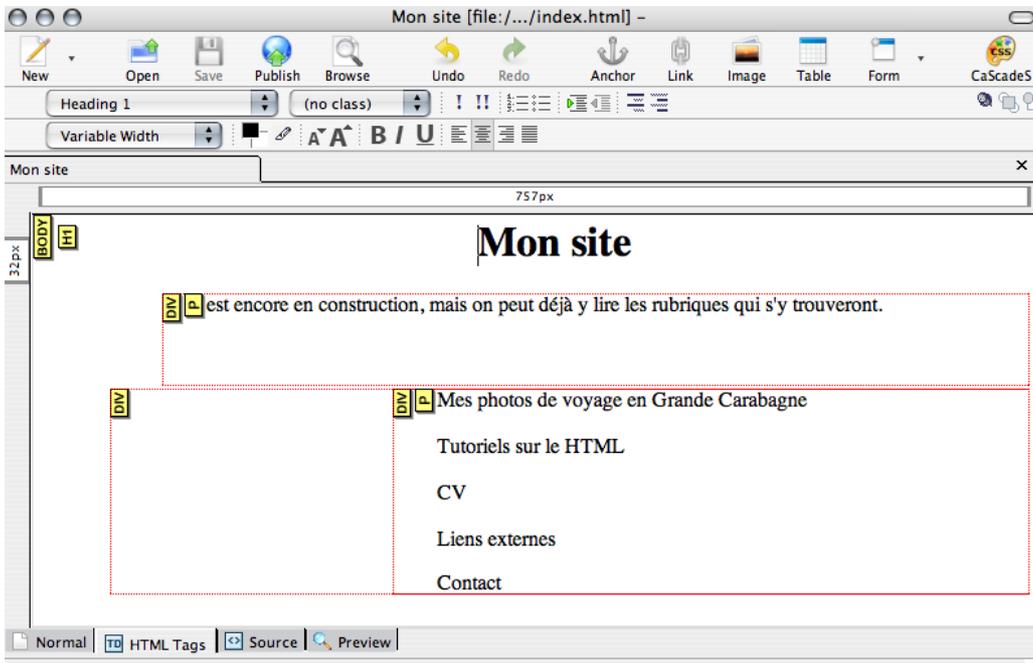
Le premier onglet affiche la page avec des cadres invisibles (non plus cette fois des caractères). On ne les appellera pas cadre pour éviter une confusion (ce qu'on appelait « cadre » est aujourd'hui déconseillé dans les pages web, et l'on n'en reparlera plus), on dit des divisions.



Le quatrième montre la page telle qu'elle apparaît dans un navigateur.



Les deux autres onglets sont quand même très intéressants pour le débutant. Le deuxième montre la page en affichant les balises qui la structurent.



Au titre correspond la balise « H1 », c'est-à-dire l'en-tête (*header*) de niveau 1. Aux paragraphes correspond « P ». On distingue aussi les divisions par la balise « DIV ». Nous voyons distinctement les éléments qui structurent notre page.

Dans le troisième onglet nous pouvons lire le code. C'est évidemment inextricable au premier abord, même si le code est colorisé. C'est cependant très instructif, nous avons commencé à le voir dès la première partie.

```

1. <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
2. |"http://www.w3.org/TR/html4/strict.dtd">
3. <html>
4. <head>
5.   <meta content="text/html; charset=ISO-8859-1"
6.   http-equiv="content-type">
7.   <title>Mon site</title>
8. </head>
9. <body>
10. <h1 style="text-align: center;">Mon site</h1>
11. <div style="text-align: justify; margin-left: 80px;">est
12. encore en construction, mais on peut déjà y lire les rubriques qui s'y
13. trouveront.<br>
14. <br>
15. </div>
16. <div style="text-align: justify; margin-left: 40px;">
17. <div style="margin-left: 200px;">Mes photos de voyage en
18. Grande Carabagne<br>
19. <br>
20. Tutoriels sur le HTML<br>
21. <br>
22. CV<br>
23. <br>
24. Liens externes<br>
25. <br>
26. Contact</div>
27. </div>
28. </body>
29. </html>

```

(On remarquera que la barre d'outil de formatage en haut est grisée, car elle est naturellement devenue inutile.)

Nous pouvons décorer un peu plus cette page. Les fonds légèrement gris sont plus reposants pour la lecture à l'écran. C'est très simple avec le menu « couleur de fond ». Nous pouvons aussi changer la couleur des polices. Tout cela est facile mais pas très important en réalité.

Tout ce qu'on rajoute à du texte pour le décorer est comme pour la parole si l'on criait, et plus on crie, moins on se fait comprendre. Le marchand de rue crie, bien sûr, mais seulement pour attirer les clients, capter leur attention et leur asséner des messages très simples. « Trois pour le prix d'un ! » Si quelqu'un voit votre page, c'est qu'il y a déjà été attiré, ce n'est plus la peine de crier. Maintenant, libre à vous de placer des publicités sur votre page qui, pendant que vous parlez, crieront pour attirer le lecteur ailleurs.

### *L'exportation*

Cette première méthode est la plus simple, la plus évidente, la plus souvent expliquée. Il suffit d'installer le logiciel et de se laisser guider par le menu d'aide. Elle a cependant un gros inconvénient si l'on veut faire davantage qu'une page d'accueil et quatre ou cinq autres petites : on doit tout prendre à zéro. Or dans la pratique ce que nous souhaitons mettre ne ligne est déjà tout fait et bien fait. Notre document est déjà structuré, illustré, et notre problème est de savoir seulement comment nous y prendre pour en faire le plus rapidement et le plus simplement possible une page web, voire un site entier.

Il suffit de l'exporter en HTML. L'exportation n'est pas non plus compliquée, comme nous avons commencé à le voir dès le début. On pourra s'en contenter si : on ne crée qu'un petit site de quelques

pages ; on a utilisé son traitement de texte dans les règles de l'art ; on n'est pas trop exigeant sur le résultat.

Les difficultés s'accumuleront de toute manière au fil des mises-à-jour et de la croissance du site — car un site croît toujours et ne décroît jamais.

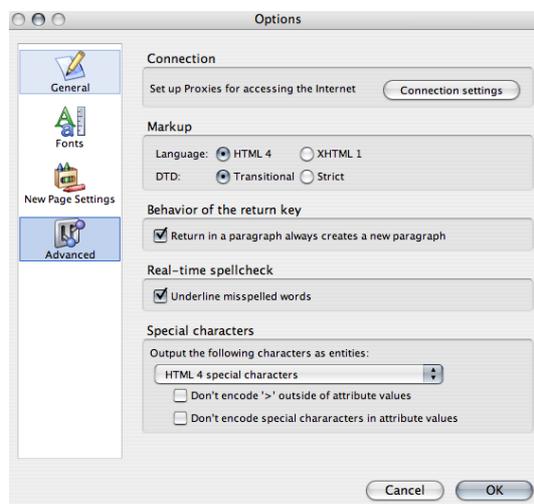
On doit savoir d'abord qu'il existe plusieurs versions du HTML (cinq plus les versions intermédiaires et des versions « transitionnelles »). Ceci ne doit quand même pas trop nous inquiéter. Les navigateurs sont très laxistes avec le code. C'est un peu comme l'orthographe du français, quelques fautes ne rendent pas un texte illisible. De plus, seul le programme lit le code, et non un lecteur humain ; en principe donc, personne ne le saura. Les pages sans faute sont extrêmement rares sur le web, même sur les sites où l'on s'attendrait à une autre rigueur, comme ceux des éditeurs de site. Toutefois, comme avec l'orthographe du français, on ne doit quand même pas exagérer.

Le plus grand nombre d'erreurs de code vient d'un mélange de balises propres à des versions différentes du HTML, et elles auront d'autant plus de chances de se produire qu'on se servira de plusieurs logiciels, ou encore qu'on fera des mises-à-jour de ses pages sur la longue durée, soit avec de nouveaux logiciels, soit avec de nouvelles versions successives des mêmes. Il est à peu près impossible de se contenter d'exporter à partir d'un traitement texte pour créer et gérer un site.

## UN CAS CONCRET

Pour être plus clair et plus précis, il est nécessaire d'aborder un cas concret. Le meilleur est d'éditer cet ouvrage lui même. Pour cela, il sera utile d'employer plusieurs logiciels. On exportera le texte à partir d'un traitement de texte. Open Office est une bonne solution. On utilisera aussi un éditeur de site. Kompozer, est un très bon choix. Ces deux programmes peuvent être suffisants, mais on peut en employer d'autres : un éditeur de texte, de préférence doté d'un correcteur de syntaxe ; peut-être un éditeur spécifique de CSS (feuilles de style) ; etc.

Il vaudra mieux commencer par accorder les deux programmes sur leur code. Par défaut, Open Office utilise du HTML 4 *transitionnal*. Ce n'est peut-être pas le choix optimal, mais il n'en laisse pas beaucoup d'autres et ce sera le meilleur dont on dispose, jusqu'à une prochaine version. Kompozer, lui, exporte par défaut en HTML 4 strict. Le plus simple serait alors de modifier les préférences et d'opter aussi pour du *transitionnal*.



Pourquoi va-t-on utiliser au moins ces deux logiciels. Le traitement de texte, c'est évident, sert à écrire et formater le texte, puis à l'exporter. L'éditeur de site sert, lui, à créer une matrice dans laquelle on exporte le texte, comme avec un logiciel de PAO pour l'édition sur papier.

Nous devons d'abord déterminer en combien de pages nous exporterons le texte. Nous pouvons en effet faire un seul gros fichier HTML. Les liens internes pour naviguer entre la table et les chapitres seront alors essentiels.

Ce n'est toutefois pas très pratique de naviguer dans un gros document. La seule bonne raison que nous aurions de le faire est de permettre le téléchargement en un seul fichier (plus un dossier image). C'est aussi un bon choix pour sauvegarder et échanger des documents dans un format très portable. C'est cependant une mauvaise solution pour la lecture en ligne.

Si l'on découpe le texte, en devra définir en combien de parties. La question est alors de décider du nombre et de la taille de ces parties. Choisira-t-on le premier niveau, par chapitre, ou le second, par sous-chapitre, ou le troisième... ?

Pour choisir, on doit se baser sur la difficulté du texte qu'on édite, l'effort et les capacités de lecture qu'on suppose du visiteur. Plus le texte s'adresse à un lecteur attentif, prêt à un effort et capable de le faire, plus il est préférable de présenter le texte sur des grandes pages. Plus le texte est facile à lire et contient des informations simples et peu liées entre elles, plus il est préférable de le découper en de nombreuses pages autonomes.

On se perd souvent plus dans un dédale de pages que dans la complexité intrinsèque d'un ouvrage, mais d'un autre côté, les pages trop longues découragent la lecture. L'évaluation n'est donc jamais simple. Il me semble que la division de cet ouvrage en quatre parties, plus la table, la préface, l'introduction et une annexe est raisonnable.

L'éditeur de site nous servira à dessiner l'apparence de ces pages et à en créer un modèle dans lequel nous importerons les chapitres.

### *L'exportation*

Lorsque nous exportons un texte pour l'éditer en plusieurs parties, plusieurs possibilités nous sont offertes. Nous pouvons l'exporter intégralement et le découper après. Nous pouvons le découper d'abord en plusieurs fichiers que nous exportons séparément, nous pouvons enfin exporter séparément des sélections du texte.

Il n'est pas de solution qui soit absolument meilleure qu'une autre. Tout dépend d'abord des performances de nos programmes et de notre système. Tous les traitements de texte ne permettent pas d'exporter seulement une sélection. Tous dépend aussi des commodités de navigation et de sélection dans un document.

Le choix d'exporter le document entier en un seul fichier a l'avantage de pouvoir faire des corrections systématiques sur sa totalité en une seule fois. Par exemple, Open Office a tendance à utiliser dans les balises d'image (<img>) l'identifiant « name » (facultatif) à la place de « alt » (obligatoire) ; il est possible de faire un remplacement systématique sur tout le document en une seule fois.

### *Le modèle de page*

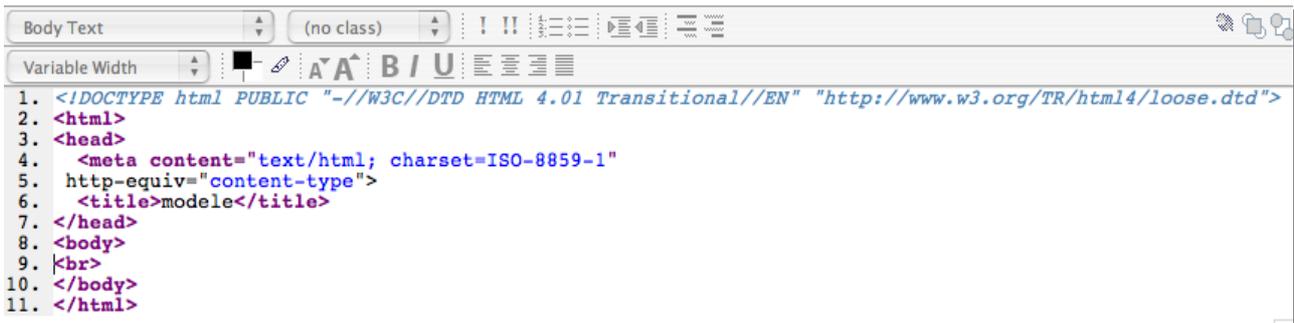
Il est logique que toutes les pages aient la même présentation et que nous n'ayons pas à recommencer le même travail pour chacune. Dans notre traitement de texte, l'ouvrage a déjà une présentation soignée, mais elle est adaptée au papier et pas à l'écran.

La mise en page pour l'écran pose des problèmes très particuliers. Certains s'acharnent à ouvrir la fenêtre de leur navigateur sur toute la surface de leur grand écran, d'autres cherchent à lire sur celui de leur téléphone portable. Même dans des cas plus raisonnables, la différence entre des

affichages de 800 ou de 1280 pixels ne sont pas identiques, surtout en tenant compte de la distance en centimètres, sur des écrans qui tendent à devenir à la fois plus grands pour les machines de bureau, et plus petits pour des ultra-portables.

### La création de la page

Que se passe-t-il dès qu'on crée une nouvelle page dans un éditeur HTML ? Dans le volet « normal », rien de visible ; dans le volet « balises HTML », rien non plus puisqu'on n'y a encore rien rentré. Il en va bien autrement dans l'onglet « code source ».



```

1. <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
2. <html>
3. <head>
4.   <meta content="text/html; charset=ISO-8859-1"
5.     http-equiv="content-type">
6.   <title>modele</title>
7. </head>
8. <body>
9. <br>
10. </body>
11. </html>

```

— Nous voyons d'abord le DDT et nous pouvons nous assurer qu'il est bien en HTML 4.01 *Transitionnal* :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

— Puis nous voyons en dessous des enclassements de balises. Tout y est enserré entre la balise ouvrante <html> et la balise fermante </html>. Nous y voyons comment fonctionnent toutes ces balises : elles s'appliquent à tout ce qu'elles enferment.

Notre page est en HTML mais nous aurions pu la faire en XHTML. La DDT aurait alors annoncé « XHTML 1 », et les balises auraient été <xhtml></xhtml>.

Entre cette paire de balises, on en trouve encore deux autres paires : <head></head> et <body></body>.

— Nous l'avons déjà vu, entre les deux dernières, <body></body>, nous avons tout ce qui est destiné à s'afficher dans la fenêtre du navigateur. Pour l'instant, il n'y a rien. (Sauf un saut de ligne, <br>.)

— Entre les balises <head></head>, se trouve le code à interpréter par des programmes. Nous y trouvons la paire <title></title> qui contient le titre de la page. Il est destiné à s'afficher dans la fenêtre du navigateur. Pour l'instant, dans la barre de la fenêtre de l'éditeur, c'est le nom du fichier qui est affiché : modele.html.

Nous préférons ne pas mettre d'accents dans les noms de fichiers, et n'utiliser que les caractères ASCII. Nous ferons bien aussi de proscrire les majuscules pour éviter toute erreur. En effet, les navigateurs ne sont pas sensibles à la casse en local, mais ils le deviennent en ligne, et cela génère souvent des erreurs.

— Nous avons aussi une balise META :

```
<meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
```

Il en existe bien d'autres plus ou moins indispensables, plus ou moins utiles. On observera que ces balises-là ne vont pas par paires, et il en existe d'autres dans ce cas qui se placent dans le HEAD ou dans le BODY (comme le saut de ligne <br>).

Cette balise « charset » est la plus indispensable, Elle indique au navigateur avec quel jeu de caractères il doit afficher la page. S'il n'y en avait pas, il l'afficherait avec le jeu de caractères défini

par défaut, et notre page ne pourrait être lue correctement que par ceux qui utilisent par défaut l'ISO-8859-1 sur leur système. Nous ignorerions donc une part importante des habitants de la planète, et notamment des francophones et gallicisants.

— Il en est qui peuvent toujours être utiles : `<META name="date" content="nnnn-n-nn">`, qui indique la date de création du fichier sous la forme « année-mois-jour ».

— D'autres le sont quelquefois : `<META name="robots" content="noindex">`. La balise « robot » indique aux moteurs de recherche ce qu'ils doivent faire de notre page. La variable « noindex » (*no index*) leur demande de ne pas l'indexer. C'est très pratique pour un travail qui, sans être secret, ne concerne que des gens précis qui en connaissent l'URL, ou qui est à un stade d'évaluation. L'autre valeur pourrait être : « index ».

Un éditeur HTML devrait nous permettre d'ajouter toutes les META qu'on souhaite, mais il ne faut quand même pas trop y compter.

Bien d'autres balises peuvent se mettre encore dans le HEAD.

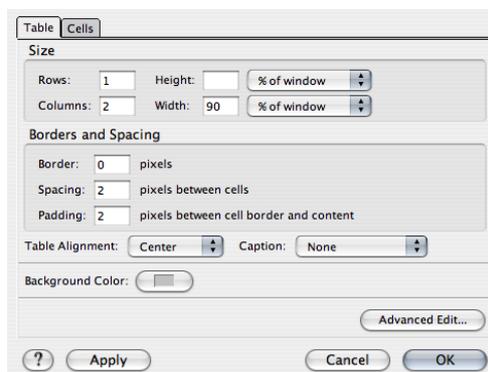
### Structure

Toutes nos pages devront donc avoir :

- Une même couleur de fond.
- Une même en-tête. Celle-ci devrait au moins contenir le nom de l'ouvrage, celui de l'auteur, un lien vers l'entrée du site qui l'héberge ; peut-être une adresse pour contacter l'auteur, une icône, un logo...
- Un pied de page, indiquant peut-être la date, le copyright et/ou la licence, etc.
- Des marges, car il est pénible de lire un texte qui va d'un bord à l'autre de la fenêtre.
- De quoi naviguer d'une page à l'autre.

Pour structurer cela, on peut utiliser des tableaux. Il vaudrait mieux cependant utiliser des blocs : ce serait plus simple si l'on voulait ensuite faire le mouvement inverse de la page web au traitement de texte. On utilisera pour cette fois des tableaux malgré tout, car s'il est plus facile de gérer des blocs directement avec du code, on ne le connaît pas encore. Il est au contraire beaucoup plus facile de créer les tableaux sans connaître de code, mais on devra savoir qu'on se compliquera la vie pour rééditer le HTML sur un traitement de texte.

On fera donc trois tableaux. Un pour l'en-tête, avec au moins deux colonnes si l'on veut y placer un logo.



On lui donnera pour valeur 90% de la taille de la fenêtre par exemple. Pour la bordure, on prendra 0 pixels, puisque le tableau sert seulement à structurer le contenu et ne doit pas être vu en tant que tel. Et on le centrera.

Pour les cellules, on donnera par exemple 120 pixels pour la première colonne qui devra contenir un logo.

The screenshot shows the 'Table Cells' dialog box. It has two tabs: 'Table' and 'Cells'. The 'Cells' tab is active. Under 'Selection', there is a dropdown menu set to 'Cell', and buttons for 'Previous' and 'Next'. A note states: 'Current changes will be applied before changing the selection.' The 'Size' section has checkboxes for 'Height' and 'Width'. 'Width' is checked and set to '120 pixels'. The 'Content Alignment' section has checkboxes for 'Vertical' and 'Horizontal'. 'Horizontal' is checked and set to 'Center'. There are also checkboxes for 'Cell Style' (set to 'Normal'), 'Text Wrap' (set to 'Wrap'), and 'Background Color' (with a note '(Let table color show through)'). A note at the bottom says: 'Use checkboxes to determine which properties are applied to all selected cells'. Buttons include '?', 'Apply', 'Cancel', 'OK', and 'Advanced Edit...'.

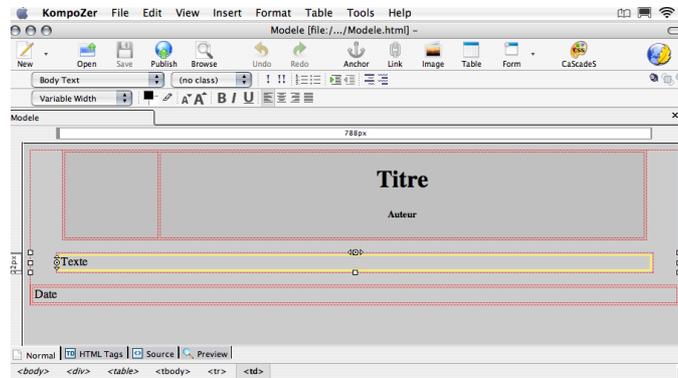
On en fera un autre pour le texte, avec une seule colonne, une seule rangée et une largeur définie, pour que les lignes ne s'étalent pas trop sur un grand écran. On lui donnera une largeur de 800 pixels, pour que le texte entre bien dans la fenêtre sur un petit écran. On le centrera aussi.

The screenshot shows the 'Table' dialog box. It has two tabs: 'Table' and 'Cells'. The 'Table' tab is active. Under 'Size', there are fields for 'Rows' (1), 'Columns' (1), 'Height' (empty), and 'Width' (800 pixels). The 'Borders and Spacing' section has 'Border' (0 pixels), 'Spacing' (2 pixels between cells), and 'Padding' (2 pixels between cell border and content). The 'Table Alignment' section has 'Table Alignment' (Center) and 'Caption' (None). There is a 'Background Color' section with a note '(Let page color show through)'. A note at the bottom says: 'Use checkboxes to determine which properties are applied to all selected cells'. Buttons include '?', 'Apply', 'Cancel', 'OK', and 'Advanced Edit...'.

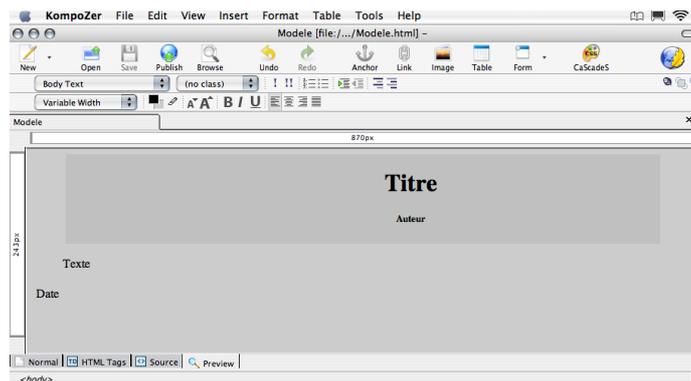
Puis on en fera un troisième de même pour le pied de page.

## CE QU'EST UN SITE

Notre modèle devrait donc avoir cette apparence dans la fenêtre de KompoZer, avec l'onglet « normal », après qu'on ait mis un gris léger en fond de page, et peut-être un plus soutenu comme couleur de fond du premier tableau.



L'intérêt pour un débutant de travailler avec un tel logiciel est qu'il lui permet de voir affichés les cadres invisibles de tous les objets qu'il rajoute. Dans la fenêtre d'un navigateur, il ne les verrait pas plus que dans l'onglet « Aperçu ».



Mais en code source, ce serait un peu inextricable :

```
8. <body
9. style="color: rgb(0, 0, 0); background-color: rgb(204, 204, 204);"
10. link="#666666" alink="#cc0000" vlink="#999999">
11. <div style="text-align: center;">
12. <table
13. style="background-color: silver; width: 908; text-align: left; margin-left: auto; margin-right: auto;"
14. border="0" cellpadding="2" cellspacing="2">
15. <tbody>
16. <tr>
17. <td style="width: 120px; text-align: center;"></td>
18. <td>
19. <h1 style="text-align: center;">Titre</h1>
20. <h5 style="text-align: center;">Auteur</h5>
21. </td>
22. </tr>
23. </tbody>
24. </table>
25. <br>
26. <table
27. style="width: 800px; text-align: left; margin-left: auto; margin-right: auto;"
28. border="0" cellpadding="2" cellspacing="2">
29. <tbody>
30. <tr>
31. <td>Texte</td>
32. </tr>
33. </tbody>
34. </table>
35. <br>
36. <table style="text-align: left; width: 100%;" border="0"
37. cellpadding="2" cellspacing="2">
38. <tbody>
39. <tr>
40. <td>Date</td>
41. </tr>
42. </tbody>
43. </table>
44. </div>
45. </br>
46. </body>
```

### Le fichier du texte

Maintenant que nous avons la matrice de nos pages, et que nous pouvons la parfaire davantage, nous devons nous occuper du fichier de leur contenu que nous avons exporté en HTML. (Nous aurions pu l'exporter aussi en autant de fichiers.)

Ouvrons maintenant ce fichier « txt.html » et voyons ce qu'il contient. Affichons-le dans le volet « Code HTML » de Kompozer. Nous voyons tout de suite que la partie HEAD est bien plus importante que sur notre modèle.

```

1. <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
2. <html>
3. <head>
4.   <meta http-equiv="CONTENT-TYPE"
5.   content="text/html; charset=iso-8859-1">
6.   <title></title>
7.   <meta name="GENERATOR" content="OpenOffice.org 3.1 (Unix)">
8.   <meta name="AUTHOR" content="Jean-Pierre Dep&eacute;eacute;tris">
9.   <meta name="CREATED" content="20090914;12115300">
10.  <meta name="CHANGEDBY" content="Jean-Pierre Dep&eacute;eacute;tris">
11.  <meta name="CHANGED" content="20090914;14222000">
12.  <style type="text/css">
13.    <!--
14.      @page:right { margin: 2cm }
15.      @page:left { margin: 2cm }
16.      P { text-indent: 0.6cm; margin-bottom: 0cm; text-align: justify; widows: 2; orphans: 2; pag
17.      H2 { margin-top: 1.02cm; margin-bottom: 0.21cm }
18.      H2.western { font-family: "Times New Roman"; font-size: 14pt; font-style: italic }
19.      H2.cjk { font-family: "Arial"; font-size: 14pt; font-style: italic }
20.      H2.cjl { font-size: 14pt; font-style: italic }
21.      H3 { margin-bottom: 0.21cm }
22.      H3.western { font-family: "Times New Roman"; font-size: 12pt }
23.      H3.cjk { font-family: "Arial" }
24.    -->
25.  </style>
26. </head>
27. <body dir="ltr" lang="fr-FR">

```

Nous allons y chercher ce qui nous intéresse pour l'exploiter dans notre fichier « modele ».

Le DDT est en principe le même ; la META « charset », aussi. Quelques autres META peuvent nous être utiles et nous les copierons dans le HEAD de notre modèle. (Nous remarquons cependant qu'Open Office exporte en HTML 4.0, et Kompozer en, 4.01.)

On voit une nouvelle balise qui nous intéresse davantage : <style></style>. Nous pouvons immédiatement en copier le contenu dans un éditeur de texte. Nous l'enregistrerons au premier niveau du site sous le nom, par exemple, de « methode.css » (sans majuscule toujours). Nous nous en occuperons plus tard.

```

@page:right { margin: 2cm }
@page:left { margin: 2cm }
P { text-indent: 0.6cm; margin-bottom: 0cm; text-align: justify; widows: 2; orphans: 2;
page-break-before: auto }
H2 { margin-top: 1.02cm; margin-bottom: 0.21cm }
H2.western { font-family: "Times New Roman"; font-size: 14pt; font-style: italic }
H2.cjk { font-family: "Arial"; font-size: 14pt; font-style: italic }
H2.cjl { font-size: 14pt; font-style: italic }
H3 { margin-bottom: 0.21cm }
H3.western { font-family: "Times New Roman"; font-size: 12pt }
H3.cjk { font-family: "Arial" }

```

### La balise BODY

Voyons maintenant comment est la balise <body>.

Quelles que soient les qualités d'un programme, il ne lit pas dans nos pensées. Soit nous pouvons lui spécifier exactement ce que nous attendons, et cela s'appelle programmer, soit il

accomplit bêtement les tâches pour lesquelles il est fait. Nous serons donc toujours partiellement déçu.

Nous savons déjà qu'il n'y a pas d'identifiant « alt » dans les balises d'image. Si l'on est scrupuleux, on les corrigera (fonction « Remplacer »). L'identifiant « alt » indique le texte qui devra être lu par un programme de reconnaissance vocale qu'utilisent les malvoyants. Pour l'occasion, elle ne leur apprendra guère plus qu'il se trouve là une image qu'ils ne peuvent pas voir ; ce qui est déjà quelque chose. Si encore une fois on est scrupuleux, on renseignera davantage les champs.

Nous trouvons aussi beaucoup de balises <font></font> qui ne devraient pas se trouver dans du HTML 4, et c'est pourquoi nous avons choisi « Transitionnal ». Si l'on est très scrupuleux, on les supprimera aussi ; ça ne devrait en principe rien changer, dans la mesure où tout le texte est composé dans la même police et que tous les autres enrichissements sont déjà indiqués autrement.

Nous trouverons encore, tout en bas, Une balise <div type="FOOTER"> :

```

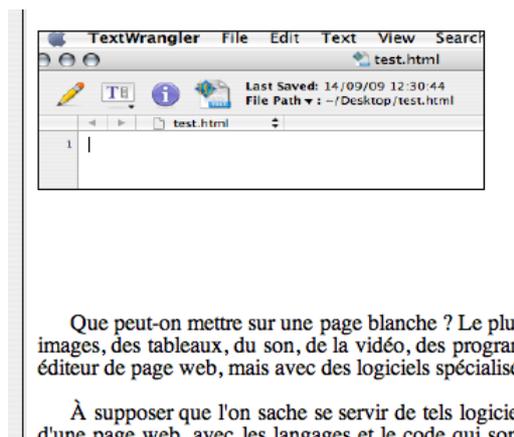
133. </p>
134. <div type="FOOTER">
135. <p
136. style="text-indent: 0cm; margin-top: 0.5cm; widows: ; orphans: ; page-break-before: auto;"
137. align="center"> <sdfield type="PAGE"
138. subtype="RANDOM" format="PAGE">2</sdfield></p>
139. </div>
140. </body>
141. </html>

```

C'est une indication pour imprimer le pied de page, qui n'a pas une réelle utilité dans une page web, et nous la supprimons aussi.

Ce devrait être à peu près tout, si le texte a été composé avec Open Office dans les règles de l'art. Voyons ce que donne le fichier dans le volet « Aperçu ».

Si la présentation ne correspond pas bien à ce que nous souhaitons, nous ne nous en soucierons pas trop. Nous pourrions la corriger plus tard autrement avec la feuille de style. Nous noterons simplement les principaux détails qui ne nous conviennent pas.



Les images sont alignées à gauche et non centrées, les paragraphes ont trop d'espace entre eux, etc.

## La CSS

Les CSS sont un langage de mise en forme des fichiers HTML. La CSS de notre page est déjà toute faite : nous l'avons déjà trouvée entre les balises `<style></style>`. Nous allons apprendre d'abord succinctement à la lire, et nous pourrons commencer à la corriger.

```
<style type="text/css">
<!--
@page:right { margin: 2cm }
@page:left { margin: 2cm }
P { text-indent: 0.6cm; margin-bottom: 0cm; text-align: justify; widows: 2; orphans: 2;
page-break-before: auto }
H2 { margin-top: 1.02cm; margin-bottom: 0.21cm }
H2.western { font-family: "Times New Roman"; font-size: 14pt; font-style: italic }
H2.cjk { font-family: "Arial"; font-size: 14pt; font-style: italic }
H2ctl { font-size: 14pt; font-style: italic }
H3 { margin-bottom: 0.21cm }
H3.western { font-family: "Times New Roman"; font-size: 12pt }
H3.cjk { font-family: "Arial" }
-->
</style>
```

Les deux premières lignes concernent les marges de la page si on l'imprime. Ces indications d'impression, comme les dernières balises de la partie BODY, ne sont pas très judicieux pour du HTML, et même déconseillées. Si l'on tient à la mise en page sur papier, il vaut mieux utiliser un autre format comme le PDF. Si l'on souhaite pouvoir modifier le document, on utilisera alors un format comme le XML, TeX, ou au pis aller du RTF. Le principal avantage du HTML est de permettre de modifier autant qu'on veut la mise en page sans que cela ne corrompe aucune propriété du texte et de sa structure. On pourra donc effacer ces deux premières lignes.

La troisième ligne concerne les paragraphes.

`P { text-indent: 0.6cm; margin-bottom: 0cm; text-align: justify; widows: 2; orphans: 2; page-break-before: auto }`

P désigne le *sélecteur*, ici les paragraphes. Entre les accolades sont des *déclarations* qui décrivent comment le style est appliqué. La déclaration a deux parties : la *propriété* (text-indent) et la *valeur* (0.6 cm), dans ce cas, elle indique un alinéa de 0,6 centimètres. La deuxième déclaration justifie le texte. Les deux suivantes interdisent les orphelines en fin ou en début de page. C'est tout.

Les quatre lignes suivantes concernent les titres de deuxième niveau. La première, H2, concerne tous ces titres. La deuxième, ne s'applique qu'aux titres suivis de `class="western"` au sein de la même balise (`<H2 class="western">`). Le principe est le même pour les deux suivantes.

`H2 { margin-top: 1.02cm; margin-bottom: 0.21cm }`

`H2.western { font-family: "Times New Roman"; font-size: 14pt; font-style: italic }`

`H2.cjk { font-family: "Arial"; font-size: 14pt; font-style: italic }`

`H2ctl { font-size: 14pt; font-style: italic }`

On comprend la suite.

On retiendra quelques points à ne jamais négliger : Dans un fichier HTML, la casse du code n'a aucune importance ; dans les CSS, non. **On doit être attentif aux majuscules et aux minuscules.** À l'inverse, les espaces autour des ponctuations (“{” “;” “:”) n'ont aucune importance. L'ordre des déclarations n'a pas non plus d'importance.

Le contenu de cette feuille de style est insuffisant, et il n'est pas étonnant que la présentation nous déçoive.

Tout d'abord, nous avons choisi une seule et même police pour tout le document et elle n'est indiquée que dans une seule déclaration : `font-family: "Times New Roman"`.

La taille de la police n'est pas toujours indiquée, et quand c'est le cas, elle l'est en points, c'est-à-dire en mesures absolues. Les valeurs absolues possibles dans une CSS sont les millimètres (mm), les centimètres (cm), les pouces (in), les points (pt) et les picas (pc). Il est préférables d'utiliser des valeurs relatives aux écrans : em (hauteur des caractères), ex (hauteur de la lettre x), px (pixel). Il vaut mieux encore utiliser des indications plus relatives encore : medium, small, large, et x-small, xx-small, x-large, xx-large ; ou encore smaller, larger.

Dans le sélecteur P nous lisons : *margin-bottom: 0cm*. Cela signifie qu'il n'y a pas d'espace après le paragraphe ; comme les paragraphes sont cependant trop espacés, c'est qu'il y en a devant. Nous allons donc copier et coller cette déclaration et changer *bottom* par *top* : *margin-top: 0cm*. Ces deux déclarations devraient faire qu'il n'y ait plus d'espace entre deux paragraphes. (Nous pourrions aussi changer partout *cm* par *px*, ce qui en l'occurrence n'a aucune importance.)

Voilà à quoi devrait ressembler finalement notre déclaration de paragraphe :

```
P {font-family: "Times New Roman": font-size: medium ; text-indent: 25px; margin-bottom: 0px; margin-top: 0px; text-align: justify; widows: 2; orphans: 2; page-break-before: auto }
```

Nous pouvons continuer pour tous les sélecteurs jusqu'à ce que la présentation nous satisfasse complètement, ou du moins suffisamment. Ce que nous sommes en train de faire n'est qu'un exercice pour comprendre le principe et savoir l'appliquer. Nous allons ensuite recommencer plus rigoureusement.

### La CSS externe

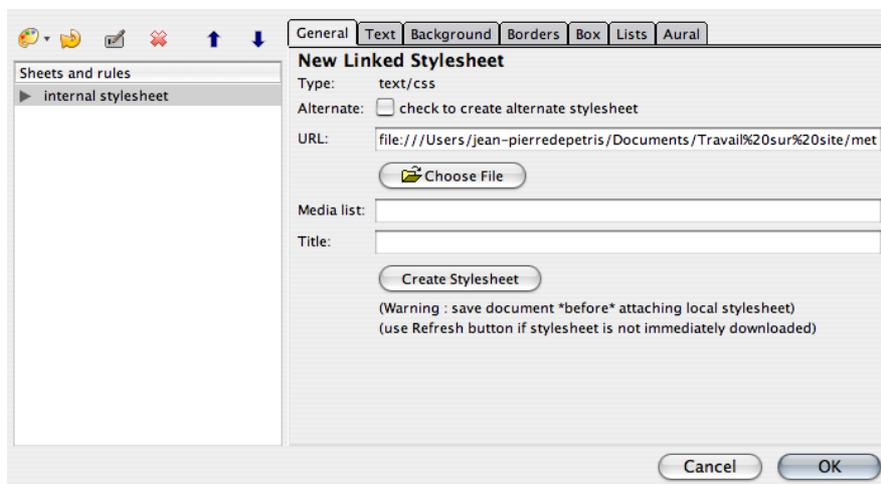
Quand nous sommes satisfaits, nous copions encore une fois le nouveau contenu de la balise STYLE, et nous allons le coller à la place de celui de notre fichier « *methode.css* ».

Nous n'y ajoutons rien d'autre. Il n'a besoin d'aucune sorte de déclaration.

Nous avons maintenant un fichier CSS externe qui est très pratique si nous voulons apporter des modifications en une seule fois sur toutes nos pages.

Avant cela, pour que ce fichier puisse nous servir, il doit pouvoir être lu dans la page HTML de notre modèle. Nous allons donc créer un lien dans celle-ci.

Nous commençons par ouvrir le fichier « *modele.html* ». Nous cherchons dans le menu outil l'éditeur de CSS, nous sélectionnons « *Linked Stylesheet* » dans le menu déroulant du bouton à l'extrême gauche, et nous sélectionnons notre fichier à l'aide du bouton « *Choose File* ».



Nous observons le changement dans le volet code HTML.

```
<title>Modele</title>
<link rel="stylesheet" type="text/css"
href="methode.css">
</head>
```

Maintenant, nous pouvons créer les diverses pages de notre site et y coller les différentes parties de notre fichier « txt.html ». Créons notre premier fichier par le menu « enregistrer sous » et appelons-le « methode\_1.html ».

Ouvrons aussi notre fichier « txt.html » et sélectionnons-y le passage que nous souhaitons coller. Copions-le et revenons à la page « methode\_1.html ». Nous plaçons notre curseur dans le tableau du milieu et nous y collerons le texte. Le résultat devrait correspondre à nos attentes.

## ***UNE APPROCHE PLUS RIGOREUSE DU MÊME CAS***

### ***Remarque sur le code source et les programmes***

Il n'y a aucune raison à priori de se préoccuper du code si des programmes sont là pour l'écrire à notre place. Cependant, à l'usage, il est souvent plus simple de l'entrer à la main que de se lancer dans un labyrinthe de boutons, de fenêtres et de menus déroulants, qui, la plupart du temps ne nous satisfont même pas pleinement. De plus, au fur et à mesure que nous utilisons du code, nous le manipulons plus facilement et plus rapidement. Nous disposons aussi d'éditeurs de texte qui peuvent faire pour nous la moitié du chemin.

Quel que soit le travail que nous effectuons avec un programme, il devra bien, à un moment ou à un autre, s'en émanciper. Il devra sortir du programme, de l'ordinateur, du système. Sinon, c'est comme si l'on n'avait rien fait. Le temps seul d'ailleurs nous y contraint, quand le programme devient obsolète, ce qui arrive assez vite.

Les programmes qui doivent nous simplifier la vie en écrivant le code à notre place, n'en demandent pas moins un certain apprentissage. Il est souvent nécessaire de le reprendre à zéro quand nous en changeons. Les langages, eux, évoluent de façon plus modulaire et ne demandent presque jamais de tout recommencer.

On l'a vu avec les jeux de caractères : l'ISO n'annule pas l'ASCII ; ni l'UFT, l'ISO. Le HTML 4 ne nous fait pas oublier le 3, et les CSS ne rendent pas obsolète le HTML. On n'acquiert rien en vain, et les nouvelles briques se placent sur les anciennes.

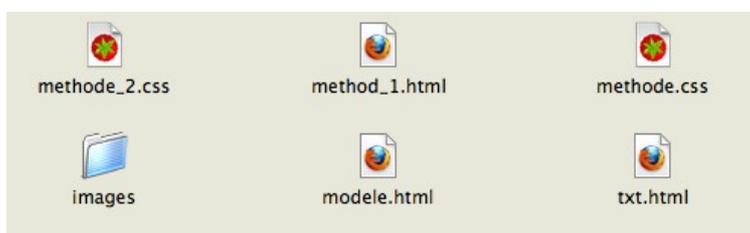
### ***Remarques sur l'organisation des répertoires***

Tous ceux qui ont une pratique soutenue de l'écrit savent, même s'ils ne songent pas à le dire, que le plus dur n'est pas d'écrire, mais de s'y retrouver dans un nombre toujours croissant de pages qui deviennent au fil des temps une jungle inextricable. En réalité, cette prolifération est cannibale ; elle revient à un effacement. Même ceux qui ne pratiquent pas si intensivement l'écriture, ont pu expérimenter que cette auto-destruction de l'écrit surgit plus rapidement qu'on s'y attend, et de telle sorte qu'elle nous surprend toujours. Dans l'action, nous surestimons toujours notre capacité à nous y retrouver dans des documents, et nous tendons à oublier l'exacte durée du procès d'écriture — généralement assez longue.

Bien souvent, un ouvrage échoue à voir le jour car il est devenu inexploitable bien avant d'être achevé. Que les documents soient physiques ou numériques n'y change pas grand chose. Créer et entretenir un site est une excellente discipline pour apprendre à s'y retrouver. À la fois, elle nous impose immédiatement un ordre rigoureux pour lier entre eux les éléments de notre travail, et maintient notre attention sur le procès à long terme par les liens externes. Il est donc important de bien organiser ses fichiers et ses dossiers qui vont très vite se multiplier.

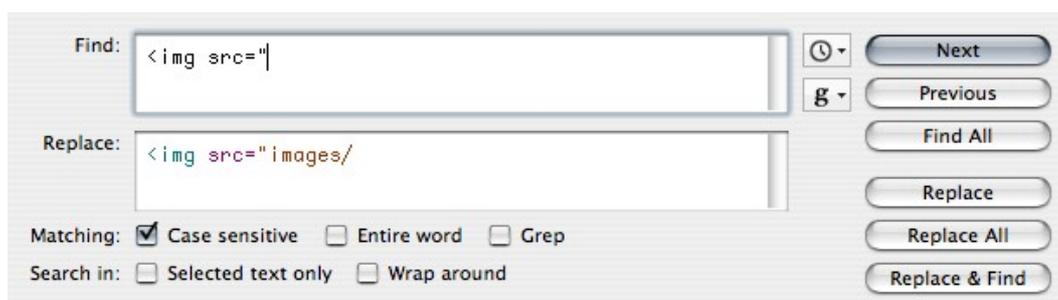
Nous avons créé un dossier pour notre ouvrage, qui contenait déjà notre modèle de page (*modele.html*). À l'aide de celui-ci, nous avons créé les autres pages, peut-être *index.html* ; *intro.html* ; *part1.html* ; *part2.html* ; *part3.html* ; *part4.html* ; *annexe.html*...

Nous avons exporté le texte entier en HTML, et nous l'avons enregistré sous le nom *txt.html*. Pendant cette opération un nombre considérable de fichiers d'images ont été générés. Tout laisser dans le même dossier est problématique ; on ne va plus s'y retrouver. Une bonne idée serait de placer tous nos fichiers d'images dans un nouveau dossier « images », à la racine du premier. Seulement dans ce cas, tous les liens seront perdus. À la place de nos images, ne s'afficheront plus que des cadres vides.



Pour ranger nos dossiers et nos fichiers qui se multiplient si vite, nous avons besoin d'un éditeur de texte qui sache renommer et déplacer des fichiers et des dossiers tout en corrigeant automatiquement les liens entre eux. Cela, Kompozer ne sait pas le faire. On devrait cependant pouvoir trouver ce genre d'outil pour n'importe quel système, gratuit ou payant.

À défaut d'un tel logiciel, on peut toujours faire un remplacement automatique.



Il suffira d'appuyer sur le bouton « Replace All » pour que tous les liens soient mis à jour. L'avantage est ici évident de réaliser cette opération avant, sur un seul gros fichier qui contient le texte intégral.

### Le formatage de la page avec des CSS

Nous avons formaté les pages à l'aide de tableaux. Le procédé est plus simple quand on utilise un éditeur en mode wysiwyg, mais il génère des complications dans le code source, et dans l'exploitation du fichier si l'on souhaite le convertir plus tard en d'autres formats. Il est plus judicieux d'utiliser la balise <div>.

Il nous suffit de trois ou quatre divisions : haut, corps, pied, et peut-être logo :

```
<div class="haut"></div>
<div class="corps"></div>
<div class="pied"></div>
<div class="logo"></div>
```

Voilà donc à quoi ressemblerait notre modèle :

```
<body>
  <div class="logo"></div>
  <div class="haut">
    <H1>Titre</H1>
    <H5>Auteur</H5>
  </div>
  <div class="corps">
  </div>
  <div class="pied">
    <P>Date</P>
  </div>
</body>
```

Le code est tellement limpide qu'un débutant saurait le lire même s'il n'est pas colorisé. On peut encore y ajouter des annotations personnelles pour s'y retrouver plus facilement. Pour les rendre invisibles, on les écrit entre « <!-- » et « --> ».

```
<body>
  <div class="logo"></div>
  <div class="haut">
    <H1>Titre</H1>
    <H5>Auteur</H5>
  </div>
  <!-- fin du haut de page -->
  <div class="corps">
  <!-- Coller ici le corps du texte -->
  </div>
  <!-- Pied de page-->
  <div class="pied">
    <P>Date</P>
  </div>
</body>
```

Voilà, tout est fait sur le modèle HTML. Il ne reste plus qu'à faire la présentation sur une feuille de style. Nous avons deux possibilités : soit nous complétons la feuille de style que nous avons déjà, soit nous en créons une seconde.

Il n'y a pas de limite au nombre feuilles de style qui peuvent être associées à une page HTML. Si celle que nous avons déjà faite nous convient pour le corps du texte, il est préférable d'en créer une autre pour la mise en forme. Il sera plus commode alors de tester les résultats, ou d'apporter à tout moment, même bien plus tard, des modifications sur l'ensemble des pages. On trouvera plus facilement les sélecteurs et leurs déclarations.

Cette feuille de style concernera donc les sélecteurs DIV. Nous pouvons ajouter aux quatre sélecteurs un autre générique pour toutes les balises DIV, qui déclarera, une fois pour toutes, les caractères communs à tous les autres, par exemple les marges de la page.

```
div. {
    }
div.logo {
    }
div.haut {
    }
div.corps {
    }
div.pied {
    }
```

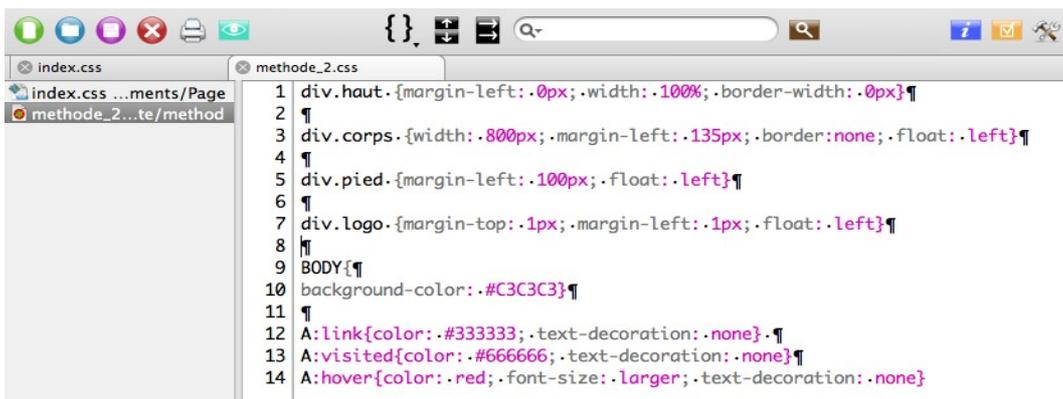
Si nous avons mémorisé la syntaxe, il n'est pas très difficile d'entrer à la main le contenu. Il est toutefois difficile de la mémoriser. En effet, le principal intérêt d'une CSS externe est de pouvoir être utilisée avec de nombreuses pages HTML. On a donc d'autant moins l'occasion de manipuler son code. Rien n'est plus simple non plus que de copier un fichier CSS d'un dossier à l'autre, dans lequel on n'aura la plupart du temps rien à changer.

C'est pourquoi, même quelqu'un qui édite très fréquemment du HTML trouve très peu l'occasion d'écrire du code CSS. Tout au plus, il modifiera quelques valeurs, mais ne retiendra pas le lexique et la syntaxe exacte, aussi simple qu'elle soit en réalité. Heureusement, beaucoup d'éditeurs de texte peuvent écrire le code à sa place. Soit qu'ils possèdent un outil pour éditer des feuilles de style, soit qu'ils sont entièrement dédiés à leur édition.

On va donc cette fois encore lier la seconde CSS dans le fichier « methode.html ». On commencera, dans la section HEAD, par copier et coller notre lien avec la première feuille de style, et on modifiera la seconde occurrence de « methode.css » en « methode2.css ».

```
6. <title>Modele</title>
7. <link rel="stylesheet" type="text/css"
8. href="methode.css">
9. <link rel="stylesheet" type="text/css"
0. href="methode2.css">
11.</head>
```

Pendant que nous y sommes, nous pouvons ajouter à notre seconde feuille de style des indications pour les couleurs de fond (*BODY { background-color: #C3C3C3}*) et pour les liens (*A:link* : liens ; *A:visited* ; liens visités ; *A:hover* ; liens survolés).



```
1 div.haut {margin-left: .0px; width: .100%; border-width: .0px;}
2
3 div.corps {width: .800px; margin-left: .135px; border: none; float: .left;}
4
5 div.pied {margin-left: .100px; float: .left;}
6
7 div.logo {margin-top: .1px; margin-left: .1px; float: .left;}
8
9 BODY {
10 background-color: .#C3C3C3;
11
12 A:link {color: .#333333; text-decoration: .none;}
13 A:visited {color: .#666666; text-decoration: .none;}
14 A:hover {color: .red; font-size: .larger; text-decoration: .none}
```

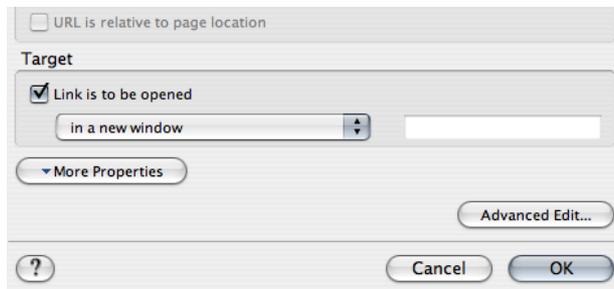
## La navigation

Avant d'en arriver là, nous aurions dû nous occuper des moyens de navigation. Pour passer d'une page à l'autre, voire à une division d'une même page, nous devons leur donner des liens.

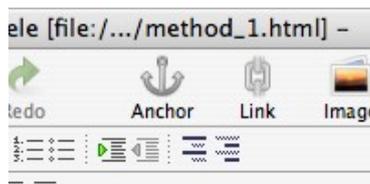
Le plus simple, serait de créer une table des matières, peut-être dans le fichier « index.html », peut-être sur une page spéciale, ou au moins chaque nom de chapitre ouvrirait la page. Il est facile d'éditer des liens à l'aide du bouton de la barre d'outils de Kompozer ou de tout autre éditeur.



Il est alors judicieux de faire s'afficher la page dans un nouvel onglet, grâce à la valeur « target="\_blank" ». On pourra ainsi revenir aisément au plan qui restera ouvert.



Il peut-être utile aussi de créer des cibles, ou ancres, dans le document, qui faciliteront la navigation en ouvrant la page à la hauteur du paragraphe recherché.



Il n'est pas très difficile d'entrer tous ces codes à la main. Toutes les balises de liens et d'ancres sont de structure `<a></a>`.

**Les balises de liens** s'écrivent selon la syntaxe `<a href="adresse"></a>`.

Quand le lien est dans le même dossier, elle s'écrit :

```
<a href="nom_du_fichier.html"></a>
```

Si le lien va dans un autre dossier :

```
<a href="nom_du_dossier/nom_du_fichier.html"></a>
```

S'il y a plusieurs sous-dossier :

```
<a href="dossier_1/dossier2/nom_du_fichier.html"></a>
```

Si le fichier est dans un autre dossier :

```
<a href="../nom_du_dossier/nom_du_fichier.html"></a>
```

```
<a href="../../dossier/nom_du_fichier.html"></a>
```

...

Si le fichier est distant, on écrit son URL :

```
<a href="http://Nom_du_site.com/nom_du_dossier/nom_du_fichier.html"></a>
```

Les balises d'ancres s'écrivent selon la syntaxe `<a name="nom_de_la_cible"></a>`.

La syntaxe du lien pointant sur la cible s'écrit en ajoutant au lien le nom de la cible précédé d'un dièse :

```
<a href="nom_du_fichier.html#nom_de_la_cible"></a>
```

Le lien avec le code pour ouvrir dans une nouvelle fenêtre s'écrit :

```
<a href="nom_du_fichier.html#nom_de_la_cible" target="_blank"></a>
```



On utilisera le même principe pour naviguer d'une page à l'autre. Dans la mesure où le texte peut toujours être imprimé ou importé, le plus discret et le plus simple sera toujours le mieux. La seule suite des caractères « -> » est très bien

On peut à la rigueur faire s'afficher dans une bulle les mots « Page suivante ». La syntaxe est : `<a href="adresse du fichier" title="Page suivante"></a>`

On peut créer les liens déjà dans le traitement de texte, à condition du moins que les pages (ou les ancres) existent déjà, même si elles sont encore vides, ou qu'on ait déjà un nom à leur donner. Ils peuvent alors être exportés en HTML et en PDF.

## CE QU'IL IMPORTE DE RETENIR

On n'écrit pas avec un éditeur de pages web ni l'on ne gère un site avec un traitement de texte. C'est là qu'est toute la difficulté de l'édition en ligne. On doit donc se donner les moyens de passer alternativement de l'un à l'autre sans que le texte se corrompe. On doit aussi pouvoir changer d'éditeur et de traitement de texte sans provoquer de catastrophes, et être capable au besoin de coopérer sur de mêmes documents en permettant à chacun de travailler selon ses habitudes et avec ses outils familiers. Pour parvenir à cela, on doit s'en tenir à quelques règles bien précises.

- **Veiller au code source** : Si l'on insère dans un modèle un texte exporté en HTML, les deux fichiers devront être encodés dans la même version de HTML, de préférence une déclinaison de la version 4 qui gère les CSS, en attendant la 5.

- Ils devront aussi avoir le même **encodage de caractères : ISO ou UFT**. Si le texte inséré est en ISO et le doctype en UFT, ce n'est pas grave, le navigateur interprétera très bien le texte. Si à l'inverse le texte inséré est en UFT et le doctype en ISO, l'affichage sera, selon les langues, défectueux ou illisible. (C'est pourquoi il est plus sûr d'exporter les textes en ISO. Le réglage se fait par les préférences du programme.)

- **Veiller à la préservation des styles**. Le texte exporté en HTML ne devrait comporter à peu près que des balises de titre (H1, H2, etc.) de paragraphe (P), ou de listes (LI) ; un minimum de classes (CLASS) ou de pseudo-classes (ID) ; un minimum de balises SPAN. Il est alors plus facile de moduler la présentation par des feuilles de style externes (CSS).

Le code exporté est d'autant plus propre que le texte a été saisi correctement : pas de sauts de ligne vides, pas de tabulations inutiles, etc. Exporter ses textes en HTML et en vérifier le code est une excellente façon d'apprendre à bien utiliser son traitement de texte.

On verra au chapitre suivant comment nettoyer le code source.

## La syntaxe des CSS

Il importe d'abord de comprendre la syntaxe des CSS. Elle se construit sur ce modèle :

*Sélecteur { propriété 1: valeur 1 ; propriété 2: valeur 2 }*

Exemple: *p { font-size: medium ; color: black }*

Les sélecteurs peuvent avoir des attributs :

*Sélecteur.attribut { propriété 1: valeur 1 ; propriété 2: valeur 2 }*

Exemple : *p.note { font-size: medium ; color: black }*

Les CSS peuvent être locales, globales ou externes :

**Locales**, elles sont insérées dans le code source. Par exemple :

`<p style="font-size: small; color: black"></p>`

Un éditeur de page web utilise par défaut des CSS locales.

**Globales**, les déclarations applicables au document entier sont définies dans la tête de la page entre les balises `<style>` et `</style>`, au milieu des balises `<HEAD></HEAD>`. C'est ainsi qu'elles sont exportées du traitement de texte comme on l'a vu plus haut.

**Externes**, elles sont sur des fichiers distincts, comme on l'a vu aussi. Ceux-ci sont liés au fichier HTML par les balises à l'intérieur du HEAD : `<link rel="stylesheet" href="nom_du_fichier.css" type="text/css">`. Il existe des éditeurs de feuilles de style externes, et les éditeurs de page web ont souvent un module interne pour cela.

Un fichier HTML peut avoir tout à la fois des CSS locales, globales et externes. Dans le cas de conflits, les CSS locales sont prééminentes sur les globales, et les globales sur les externes.

### Les sélecteurs et leurs possibles attributs

Les sélecteurs peuvent être la plupart des éléments du code HTML : p (paragraphe), h1, h2... (les divers niveaux de titre), li (listes), et aussi les barres horizontales (hr), les tableaux, leurs colonnes, leurs rangées et leurs cellules, les liens, actifs ou visités (a-link, a-hover, a-visited), les divisions (div)... et même la page entière (body).

On peut donner des attributs différents à un même sélecteur, par exemple un paragraphe de notes où l'on souhaitera que les caractères soient plus petits. On l'écrira ainsi dans la page HTML :

`<p class="notes"></p>`

et dans la CSS externe :

*p.notes { font-size: smaller }*

On peut souhaiter avoir des titres et des sous-titres dans des notes qui soient tous plus petits.

Dans ce cas, on écrira ainsi dans la page HTML :

`<p class="notes"></p>`

`<h1 class="notes"></p>`

`<h2 class="notes"></p>`

Dans la CSS on pourra utiliser une notation générique d'attribut qui évitera d'inscrire autant de fois le sélecteur :

*.notes { font-size: smaller }*

L'attribut ID s'écrit de la même façon que CLASS dans le HTML : `<p id="cit"></p>`. Dans les CSS, il est indiqué par un dièse :

```
p#cit { font-style: italic }
#cit { font-style: italic }
```

S'il y a conflit entre les attributs et les sélecteurs, ID a la prédominance sur CLASS qui a la prédominance sur le sélecteur.

La balise SPAN a la prédominance sur tous les autres sélecteurs et attributs. Elle s'utilise ainsi et seulement sous forme locale :

```
<p>Corps de texte <span style="font-style: italic">passage en italique</span> suite du
texte.</p>
```

On trouvera en ligne toutes les ressources pour aller plus loin.

### Les unités de mesure

Certaines unités correspondent à l'affichage fluide de l'écran, les autres à l'impression fixe sur papier, nous les disons relatives ou absolues.

nom de l'unité	abréviation	définition	Valeur
empâtement	em	L'épaisseur de trait d'un caractère	relative
ex	ex	La largeur de la lettre x dans une police	relative
pica	pc	1 pica fait 12 points	absolue
point	pt	1/72 de pouce	absolue
pixel	px	un point sur un écran	relative
millimètre	mm	unité d'impression	absolue
centimètre	cm	unité d'impression	absolue
pouce	in	unité d'impression	absolue

Les feuilles de style permettent d'utiliser toutes ces valeurs pour la taille des caractères (font-size) ou pour définir la taille d'éléments comme les marges, les bordures, etc.

Pour les caractères, il est préférable d'utiliser des valeurs plus relatives encore :

- On peut donner à la propriété "font-size" : *xx-small*, *x-small*, *small*, *medium*, *large*, *x-large*, *xx-large*.
- On peut encore utiliser les valeurs *smaller* et *larger*. Elles réduisent ou grossissent respectivement la taille des caractères d'un degré par rapport à ceux qui les entourent.
- Pour élargir ce choix, on peut combiner la taille avec la graisse (font-weight). Quatre valeurs sont utilisables: *normal*, *bold*, *bolder*, *lighter*.
- La graisse des caractères peut être encore définie plus finement par des nombres : 100, 200, 300, 400, 500, 600, 700, 800, 900. 400 est équivalent à *normal*, 700 à *bold*.

Il est judicieux d'utiliser les premières valeurs du tableau, « em » (épaisseur de trait d'un caractère) ou « ex » (largeur de la lettre x dans une police), pour définir les tailles d'à-peu-près tous les sélecteurs. Si le visiteur grossit ou réduit les caractères de la page web, la taille de tous les autres éléments seront alors changés en proportion.

La construction qui suit permettra d'obtenir des paragraphes dont les lignes contiendront à-peu-près le même nombre de caractères quel que soit l'affichage :

Fichier HTML : `<div class="corps"><p>...</p></div>`

Fichier CSS : `div.corps { width: 40em }`

### Les couleurs

Les couleurs peuvent être notées sur les feuilles de style par leur code hexadécimal, par leur nom (en anglais) ou par une notation en RGB (trichromie : rouge, jaune, bleu). Avec un peu d'habitude, cette dernière est plus intuitive pour trouver une couleur en entrant sa valeur.

Par exemple, la couleur "whitesmoke" (#f5f5f5) s'écrit ainsi : `rgb(245, 245, 245)`. Les mêmes valeurs pourraient être données en pourcentage : `rgb(96%, 96%,96%)`.

Noms des 16 couleurs de base avec leur code			
black #000000		green #008000	
silver #C0C0C0		lime #00FF00	
gray #808080		olive #808000	
white #FFFFFF		yellow #FFFF00	
maroon #800000		navy #000080	
red #FF0000		blue #0000FF	
purple #800080		teal #008080	
fuchsia #FF00FF		aqua #00FFFF	

### La compacité du code

Si l'on utilise pleinement les ressources des CSS externes, on préférera avoir des blocs très simples avec des sélecteurs sans attributs (`<p>` et non `<p class="xxx">`). Si l'on veut diviser sa page, on utilisera plutôt des balises `<div>` (`<div class="header">` ; `<div class="corps">` ; `<div class="note">`...) auxquelles on attribuera, peut-être sur une feuille de style distincte, des propriétés et des valeurs (`width` ; `font-size: smaller` ; etc.)

La plupart des traitements de texte ne permettent pas d'exporter un tel code. Il en existe cependant qui offrent la possibilité d'exporter en HTML sans mise en forme, comme Bean pour Mac OS.



TextEdit et Itext sur Mac OS offrent dans les préférences d'opter pour l'exportation sans feuille de style, et l'on devrait pouvoir en trouver de semblables pour les autres systèmes, mais on risque de voir seulement remplacées les balises de CSS par celles d'un HTML obsolète.

Dans le meilleur des cas, on obtient un code qui ressemble à ceci :

```
27 <p>La plupart des traitements de texte ne perm  
28 Il en existe cependant qui offrent la possibil  
29 d'exportation en HTML sans mise en forme.</p>-  
30 <p><br></p>-  
31 <p>Bean, TextEdit et Itext sur Mac OS donnent
```

Dans les faits, il est rare d'avoir la bonne surprise d'obtenir un code qui n'ait pas à être nettoyé.

# Écrire et éditer

## *APERÇU*

### *Essai de synthèse*

Tout ceci, on doit bien le reconnaître, est un peu compliqué. C'est sans doute plus difficile que de passer son permis de conduire ; mais les erreurs ont des conséquences moins désastreuses et toujours rattrapables. (Combien de temps, à propos, est-il nécessaire en moyenne pour apprendre à conduire ?) C'est beaucoup moins difficile en tout cas que d'apprendre une langue naturelle.

Est-il toutefois nécessaire de se donner cette peine, quand traînent tant de moyen en ligne de faire un blog ou d'utiliser des programmes dits « sociaux », qui se prennent en main en quelques heures, si ce n'est quelques minutes ? Cette question se prête à deux sortes de réponses. La première consiste à montrer qu'il y aurait malgré tout sur la longue durée un gain de temps et d'efforts — on l'a déjà un peu vu précédemment — ; la seconde à montrer comment ces méthodes accroissent notre puissance de travail, et la maîtrise sur la qualité et la pérennité de nos documents.

### *La communication facile*

Les deux questions se tiennent. Si l'on est convaincu de n'avoir jamais rien d'autre à faire que diffuser des messages de quelques paragraphes tout au plus, et qui perdront tout intérêt très vite, de telles connaissances sont bien inutiles.

On aurait cependant intérêt à se méfier des commodités qu'offre l'internet, d'ajouter des commentaires à des commentaires, voire des collaborations à des collaborations. Ceci ne met pas en cause l'utilité de ces moyens. On peut organiser très vite des groupes de travail efficaces sur la plus grande échelle, mais l'auto-cannibalisme de l'écrit évoqué plus haut se manifeste alors avec une voracité hallucinante. Plus l'outil est efficace, plus il peut nous conduire vite à un abîme de néant.

Les outils ne sont pas en cause. Il n'est qu'à voir les différents Wikis de développement de logiciels, ou l'encyclopédie en ligne Wikipedia. Tous ces sites ne se réduisent justement pas à des conglomerats d'interventions. Ils sont structurés autour du programme qu'ils développent, de son manuel, de ses tutoriels. (Ceux-ci sont d'ailleurs le plus souvent proposés sur des sites semblable à celui que nous avons appris à réaliser.) Wikipedia ne repose pas non plus sur des interventions sans suite, mais sur le travail de spécialistes qui coopèrent, et qui doivent bien synthétiser leur savoir, qu'ils puisent et produisent à la fois, sur d'autres sites. Les articles renvoient d'ailleurs à des sites aboutis.

Il serait fallacieux de croire que si chacun apporte sa pierre, ça construira une cathédrale. Ça fera seulement un tas de pierres où les dernières enseveliront les premières. Il est cependant tout à fait exact sur le web que la pierre d'un édifice peut aussi être celle d'un autre. Mais ça ne dispense évidemment pas de construire ces édifices.

Parce que toute pierre peut servir à divers édifices, parce que la pierre prise à un édifice ne l'en prive pas, le net n'a pas besoin des structures hiérarchiques prénumériques. Il annule en réalité

moins la hiérarchie qu'il ne les démultiplie. Pour cela, pour s'émanciper de la hiérarchisation prénumérique des tâches, les utilisateurs-producteurs doivent être le plus autonomes possible.

### *L'évolution du traitement numérique du texte*

Observons comment depuis trente ans a évolué la façon de traiter numériquement le texte. Elle s'est d'abord scindée en deux.

— D'un côté, des chercheurs ont commencé à s'échanger de gros documents, soit avec le tout nouveau HTML, dans un style aussi spartiate que celui de la machine à écrire et du stencil ; soit sous forme de fichiers EPS prêts à l'impression, édités en TeX ou en LaTeX.

— De l'autre, elle a épousé la division des tâches de l'imprimerie et de l'édition : saisie du texte déjà écrit au kilomètre ; mise en forme avec du style (généralement avec Word, mais aussi CorelWord, StarOffice, ClarisWorks, avec leurs formats propriétaires) ; mise en page avec un logiciel de PAO (le plus souvent XPress, mais aussi RagTime, avec un nombre sans cesse croissant de filtres d'importation) ; relecture de l'auteur pour approuver son ouvrage ; flashage des pages pour en faire des plaques offset ; etc. On pourrait encore faire passer cela pour de la « rationalisation des tâches » à propos d'un gros tirage, mais c'était quand même une multiplication de travaux inutiles.

— L'évolution de l'édition en ligne rend maintenant l'affichage à l'écran du HTML aussi soigné que peut l'être l'imprimé : choix des polices, style, mise en page... Le même format PDF est devenu un format standard pour l'imprimerie comme pour l'édition en ligne de documents non modifiables. On dispose d'outils pour le marquer et l'annoter à l'écran, et même le rediffuser avec ou sans ses notes.

— L'édition sur papier tend à abandonner aux auteurs tout le travail de composition du texte. Les auteurs écrivent d'ailleurs tous maintenant avec des traitements de texte sophistiqués, qu'ils sachent s'en servir ou pas. De plus en plus quelques formats dominent. Le fichier (généralement envoyés en « .doc ») est mis en page dans un logiciel de PAO de manière si standard que si l'auteur ne sait pas se servir de son traitement de texte, ses erreurs de formatage demeurent (ignorance des espaces insécables par exemple), et s'il sait, la mise en page finale est inférieure à la sienne. L'ouvrage est finalement exporté au format PDF pour être imprimé.

Il découle de cela une double évidence :

**Le fichier numérique est proprement et immédiatement *ce qu'on écrit*, non pas ce qu'on saisit après l'avoir écrit.**

Bien sûr, on peut toujours utiliser la plume ou le crayon ; on l'utilise quoi qu'on fasse, mais on produit alors un brouillon, un premier jet. On écrit vraiment, on corrige, on travaille son style avec un traitement de texte. C'est avec lui encore qu'on structure son ouvrage, qu'on le réorganise, qu'on circule dans ses pages. Bien sûr, on les imprime et on les corrige aussi au crayon, mais c'est le fichier numérique qui sert de référence, où l'on y entre des notes et des marque-pages datés automatiquement.

<p>Jean-Pierre Depétris, 28/09/09 02:08</p> <p>Et où l'on rentre des notes et des marque-page.</p>	<p>son style avec un traitement de texte. C'est avec lui encore qu'on structure son o le réorganise, qu'on circule dans ses pages. Bien sûr, on les imprime et on les co crayon, mais c'est le fichier numérique qui nous sert de référence.</p> <p>C'est notamment à partir de ce fichier numérique qu'on corrigera d'imprimerie, et qu'on aimerai avoir numérisée aussi.</p>
--	--

C'est notamment à partir de ce fichier numérique qu'on corrigera une épreuve d'imprimerie, qu'on préférera avoir numérisée aussi.

Le fichier numérique est aussi ce qui est imprimé. Il n'est pas un bouquin en librairie, ou une revue, un journal, qui ne soit (que) l'impression d'un fichier numérique.

## QUELQUES AUTRES CONCLUSIONS

De cela, on déduit :

Que ce fichier est le **véritable texte**, dont toute impression, privée ou commerciale, est une copie.

Qu'il peut prendre tour à tour ou simultanément trois formes, **trois états** : celle d'un format exploitable dans un traitement de texte (XML, RTF ou assimilé), un autre où le texte devient aussi fluide qu'une eau qui épouse la forme du récipient (HTML et annexes), un troisième où le moindre détail ne doit plus être modifié (PDF).

Que le texte est immédiatement **édité en même temps qu'écrit**, et qu'on ne saurait pas ce que signifierait faire l'un sans l'autre.

Sur un traitement de texte, écrire et éditer sont bien évidemment une même chose. Produire un fichier PDF à partir de là peut se résumer à appuyer sur un bouton. Mais éditer en HTML n'est apparemment pas la même chose qu'écrire. C'est une suite d'opérations assez complexes. On peut s'interroger sur son utilité si notre texte est déjà dans un format portable comme l'ODT, ou exporté en PDF.

### Trois états

Il n'est en réalité pas exclus que ces trois états n'en viennent pas à être réduits à deux seulement : un fichier éditable en XML et un fichier non-modifiable — un état daté — en PDF. C'est ainsi déjà que le site d'Open Office propose ses documentations en ligne.

How-Tos Writer [traitement de textes]			
Ces 'Comment faire' sont des explications pas à pas et illustrées d'utilisation des fonctions d'OpenOffice.org. Les formats dont le lien est activé (format en bleu) sont ceux disponibles. La colonne 'V2' indique que le document a été mis à jour vers la version 2.			
Styles et modèles de documents			
Titre	Description	Formats	V2
<b>Numérotation des chapitres</b>	Comment numérotter les divers chapitres qui constituent un texte. Comment exploiter le plan du document pour générer son sommaire. Comment l'actualiser voire le supprimer. Comment créer des sommaires de chapitres et des sommaires pour des sous-parties des documents.	<a href="#">PDF</a> <a href="#">ODT</a>	<a href="#">V2</a>
<b>Le navigateur</b>	Déplacements rapides dans les différents éléments d'un document, gestion du document et recopie d'objets.	<a href="#">PDF</a> <a href="#">ODT</a>	<a href="#">V2</a>
<b>Concevoir une feuille de styles</b>	Ce document complète Principes du traitement de texte et Styles et modèles. <b>Mise à jour du 6 septembre 2008.</b>	<a href="#">PDF</a> <a href="#">ODT</a>	<a href="#">V2</a>
<b>Mise en page paysage</b>	Avoir des pages en paysage et les en-têtes/pied de page en portrait. <b>Mise en ligne le 14 octobre 2007</b>	<a href="#">PDF</a> <a href="#">ODT</a>	<a href="#">V2</a>

Le HTML, le langage du web, ne servirait plus alors qu'à gérer une vitrine, des rayonnages, à présenter les contenus. Déjà, d'ailleurs, les navigateurs affichent très bien le PDF. Ce procédé a encore cependant de grandes lacunes.

Pour le PDF, le fichier peut être assez lent à ouvrir en ligne. On ne le sent pas avec une connexion rapide pour soi tout seul, sinon le PDF devient vite mal-commode.

Quant au fichier XML, on doit le télécharger sur un disque dur pour l'ouvrir dans un traitement de texte, ce qui n'est pas réellement pratique pour le consulter. On préférerait qu'il soit possible de coller directement le XML dans une page HTML entre une paire de balises, comme on a fait avec notre contenu en HTML, et de le gérer comme n'importe quel objet, image, vidéo, script... Le HTML 5 en réserve peut-être la possibilité.

Pour autant, ces trois états du texte sont moins dépendants des trois sortes de formats que l'inverse. Ils sont inhérents à la nature de l'écrit.

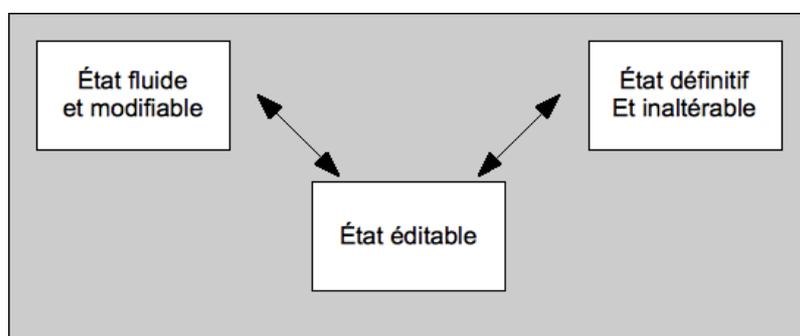
Tout écrit doit aboutir à un « état » fini, un jour, voire à une heure précises, pour lequel on s'assure qu'il ne sera plus modifié. On peut évidemment toujours passer à des états successifs, en autant de versions, de rééditions. C'est à cela que répond le format PDF.

Il doit en même temps rester fluide et malléable sans qu'il se corrompe, et se couler dans n'importe quel contenant (navigateur, traitement de texte, PAO, écran, papier...). Le format HTML correspond bien à cet usage, il est relativement simple à manipuler et laxiste sur les fautes de code.

Il doit encore rester aisément modifiable et offrir toutes les possibilités de le retravailler avant de l'exporter dans les deux précédents formats. Pour cela, le format du traitement de texte demeure irremplaçable. Au mieux, le XML peut devenir une sorte d'espéranto entre les différents outils d'édition, mais c'est un langage complexe et plus rigoureux que le HTML, dont le code est plus dur à manipuler.

Souhaiter réduire le choix entre ces trois formats pour n'en garder que deux, sinon un, revient à en attendre des qualités totalement contradictoires.

Tant que nous demandons à un texte d'être à la fois infiniment modifiable et définitivement achevé, nous aurons besoin d'un état intermédiaire, d'un état où il demeure virtuellement les deux ; où l'on contrôle ses mutations ; un état transitoire en quelque sorte.



### *Du point de vue de l'écriture*

Cette partition qui prend la forme d'outils et de langages numériques distincts, aide en réalité à écrire, en nous permettant de penser ce qui autrement serait (était) impensable : la nature contradictoire du signe écrit.

Un texte n'est pas une abstraction. Dans son état présent, son actualisation, tout peut y avoir une importance significative : la taille de l'écran, de la fenêtre ou du papier, l'aisance à le manier et le naviguer, la taille des caractères, la police, le nombre de signe par ligne et de lignes par page, etc. Tout cela a une incidence sur la lecture, et plus encore sur l'écriture.

Certains textes deviennent illisibles s'ils sont trop découpés en de petites pages. Ils se lisent mieux en lignes serrées sur de grandes. La page, celle du livre ou d'un navigateur, a une incidence directe sur l'écrit. Elle est toutefois relative : quelques caractères de plus ou de moins dans une ligne, une ligne de plus ou de moins dans un page, ont une importance négligeable.

Cette importance dépend seulement du texte. Il est préférable que des vers ne soient pas coupés par des sauts-de-ligne, ni qu'ils paraissent collés à la marge gauche. Hors ce cas extrême, il peut être bon que la longueur des lignes épouse une respiration des phrases ; mais cela peut n'avoir aussi qu'une importance négligeable.

Il se trouve que de telles décisions sont maintenant entièrement entre les mains de l'auteur, du moins au bout de ses doigts, quand il écrit sur son traitement de texte ou quand il publie en ligne, tout autant qu'elles le sont avec la plume et du papier. Elles le sont devenues plus encore, puisque l'auteur peut à chaque instant tout changer. Il est le seul maître de ce à quoi il donne de l'importance ou non.

Lorsque nous avons importé le texte dans la page HTML, nous avons opté pour une largeur de ligne de 800 pixels, et pour une taille de caractères *medium*. Nous aurions pu donner des valeurs moins relatives : 10 centimètres de largeur, par exemple, pour un corps de 12 points.

Il appartient à l'auteur seul de décider des ratios de fluidité ou de l'immuabilité de ce qu'il édite, il lui appartient de **les écrire aussi**, comme il écrit les mots.

On doit bien alors comprendre qu'il s'agit d'abord d'exigences d'écriture, et très secondairement des qualités de lecture d'un texte publié. L'enjeu est celui de la plus grande commodité, et de l'efficacité de tout le procès d'écriture.

### *Écrire et publier*

Le numérique, l'ordinateur personnel et l'internet ne réduisent pas ce procès d'écriture à une étape antérieure à la publication en ligne et qui la prépare — même en associant écriture et édition. Cette étape de la publication est au contraire intégrée dans le procès lui-même, puisqu'elle n'a aucune raison de l'interrompre. À tout moment, l'auteur peut partager son travail avec qui il le souhaite, de proches collaborateurs référencés, un cercle plus ou moins ouvert, quiconque. Il peut obtenir des avis, des corrections, des critiques, des données complémentaires. Même s'il repousse aussi tard qu'il le souhaite ce moment, il pourra toujours et à tout instant tenir compte des retours que suscite son travail, pour le modifier.

On dit que le web contient beaucoup de travaux en cours, en réalité, tous les travaux qu'on y trouve le sont virtuellement.

### *Pourquoi un site*

Nous avons d'abord éludé la question du « pourquoi » pour lui préférer celle du « comment » éditer un site. Il n'y a bien sûr pas de réponse unique, à moins qu'elle ne soit la plus simple : « Pour partager des documents. » Autant s'en tenir à cette définition générique : « Partager des documents », qui est à la fois exacte, précise et ouverte.

On ne dirait certainement pas la même chose pour l'édition imprimée. Il s'agit pourtant bien souvent aussi de partage, mais il ne se joue pas aussi immédiatement (sans médiation), dans la relation de celui ou de ceux qui produisent un document imprimé, et ceux qui l'acquièrent et le lisent. Quand bien même s'agirait-il d'un bulletin interne, d'une publication de chercheurs destinée à leurs confrères, la séparation entre le moment de la production et celui de la réception, renvoie l'idée de partage à une figure de style.

Quand j'édite en ligne cette méthode que je suis encore en train d'écrire, quelle est mon intention ? Rendre service à qui peut en avoir besoin ? Dans ce cas, j'aurais repoussé ce moment le tard possible, quand j'aurai été sûr de n'avoir laissé aucune erreur, aucun oubli, quand j'aurai jugé que tout était parfait. Or ce n'est pas précisément mon intention. J'attends au contraire de la publication qu'elle m'aide à atteindre ce point.

Je peux commencer par interroger des correspondants mieux renseignés que moi sur certains points, et je leur enverrai des extraits par courriel. Rapidement, il sera préférable que l'ouvrage soit en ligne tout entier, en l'état, pour que j'envoie plutôt des liens dans mes courriels. Dans ce cas je peux toujours protéger mon édition par un mot-de-passe réservé à mes correspondants, ou seulement en interdisant la recherche par les moteurs.

Me voilà donc déjà avec un site pour un texte qui se trouve dans un état bien inférieur à celui que j'aurais accepté pour une impression publique. Je pourrais pourtant le laisser aussi en libre accès, car quelqu'un que je ne connais pas peut me donner des retours utiles. Il suffira que je prévienne que ce travail est en cours.

L'indication « Version 0.n » y suffira. Le nombre « 0 » signifiera que cette version n'est pas finalisée (antérieurs à la version 1), et le nombre « n » sera celui de cette version d'évaluation.

### Les outils collaboratifs

Pourquoi ne pas utiliser des outils collaboratifs ? Ces outils ne sont pas à négliger, ils savent se rendre utiles, mais ils ne le sont que pour faire ce pour quoi ils ont été programmés — et ce n'est certainement pas à écrire. Même sur un site collaboratif, et peut-être plus là qu'ailleurs, on a besoin de sites sur lesquels publier des documents et vers lesquels pointer des URLs, et plus encore d'adresses de courriel.

Dans la pratique, de tels outils servent d'abord des objectifs précis et circonscrits dans le temps. Quand leur vie se prolonge, ils tendent à devenir des capharnaüms, sauf à poursuivre une démarche plus ou moins encyclopédique comme Wikipedia. Plus ils durent, et plus le temps nécessaire à les administrer s'accroît.

On ne doit pas entretenir trop d'illusions sur leur caractère « collaboratif » qui laisserait entendre que tout le monde y tient une place égale. L'internet abolit moins les hiérarchies qu'il ne les multiplie. Il multiplie les chefs, au point que chacun devient le sien, le centre des réseaux qui convergent et partent de lui. Un wiki, comme n'importe quel site, n'échappe pas à cette règle. L'administrateur ne peut pas empêcher que les autres soient chefs ailleurs, mais sur son wiki, le chef, c'est lui.

Il n'est pas sérieusement question ici d'autorité et de censure, mais de la nécessité d'administrer. Plus le temps passe et plus le wiki croît, plus le rôle de *l'admin* devient important, et difficilement partageable, mais sans-doute les tâches peuvent être déléguées. D'ailleurs, les collaborateurs n'ont ni l'envie ni de bonnes raisons d'avoir tous les droits d'administration.



Les outils collaboratifs et « sociaux », blog, wiki, forum, liste de diffusion... offrent des services dans la mesure où ils se situent comme modulations des deux pôles de l'internet : le site public et le courriel privé. Leur usage peut devenir problématique si l'on se prive de ces deux termes qui les encadrent.

Ces deux pôles sont d'ailleurs eux-mêmes modulables : en limitant plus ou moins l'accès au site (non-indexation, mot-de-passe...); en envoyant des courriels groupés (il n'est pas plus difficile d'envoyer en même temps des milliers de courriels qu'un seul, si l'on a préparé un dossier d'adresses). Nous avons déjà dans ces deux moyens de quoi permettre toute sorte de collaborations, et qui peuvent largement y suffire.

À quoi servent donc tous ces outils ? Ils peuvent permettre une adaptation plus fine à des objectifs poursuivis ; donner des moyens supplémentaires et plus commodes.

Ils peuvent aussi offrir un accès plus facile à ceux qui ne savent pas utiliser les outils de base : ceux qui ne savent pas faire un site, ou qui ne savent pas gérer sur le moyen et le long-terme une grande quantité de courriels. On devrait se garder alors de faire à partir de la seconde réponse un remède pire que le mal. S'il est justifié de ne pas faire de la technique un moyen de sélection, voire de censure, permettre de l'ignorer n'est certainement pas identique à en donner les clés.

## ***L'USAGE***

### ***Le plus simple est le mieux***

Il n'y a pas si longtemps, on trouvait sur n'importe quel CD des logiciels simples d'emploi pour créer un site, Netscape, alors le navigateur le plus utilisé, possédait un module à cet effet, Composer, et n'importe quelle revue expliquait en quelques pages illustrées comment s'en servir.

La principale difficulté était déjà la même qu'aujourd'hui : comment, pour éditer davantage qu'un petit contenu de base, passer d'un logiciel à l'autre, d'un format à l'autre ? Cette difficulté aurait plutôt diminué depuis, et le reste n'est pas devenu plus compliqué. Le code du HTML 4 n'est pas plus compliqué que celui du 3. Il serait même, lui aussi, devenu plus simple en s'étant déchargé de nombreuses fonctions sur les CSS.

Le HTML 4 et les CSS offrent plus de possibilités, mais ils n'en imposent aucune. Si l'on y tient, on peut continuer à éditer avec du HTML 4, et même bientôt du 5, d'une façon aussi spartiate qu'avec du HTML 2, et même 1. On a le droit d'ignorer les CSS, et de présenter un texte brut, comme dactylographié sur une vieille machine à écrire. Ce peut être un choix, et des plus justifiables.

D'ailleurs, plus une édition HTML est propre et minimale, plus il est facile, si en vient l'envie, de l'enrichir avec des feuilles de style ; si l'on veut, quand on veut. Rien ne contraint plus qu'hier de faire un site parfait, beau, richement paginé, si tant est que ces critères aillent ensemble. Le HTML est justement un format très peu exigeant sur la perfection.

### ***Le travail en cours***

Comme on l'a vu, l'internet favorise la publication de travaux en cours. Mieux, sur le net, tout travail est virtuellement en cours. Même l'édition d'un auteur depuis longtemps disparu peut être enrichie de notes, de références nouvelles.

Le travail en cours doit être perpétuellement modifiable, de la simple correction de coquilles ou d'inexactitudes, jusqu'à des recompositions profondes. Ceci est donc très différent de l'exemple étudié plus haut d'un texte en principe complet, édité en une seule fois, et qui se prêterait facilement à une division des tâches. La même personne, l'auteur, doit pouvoir intervenir à tout moment et à tous les niveaux : sur son fichier de travail — dans un traitement de texte donc ; sur le fichier PDF — c'est simple, il suffit de l'exporter ; sur l'édition en ligne — qui est à l'évidence plus difficile à synchroniser.

### *L'idée fallacieuse de perfection*

Inutile de s'illusionner, il n'y a pas de bonne solution universelle. En trouverait-on une que tous ceux qui veulent bien coopérer devraient l'adopter pour qu'elle le devienne réellement. Il est préférable d'abandonner définitivement l'idée de perfection, avec celle qu'il n'y ait qu'une seule méthode et une seule voie pour atteindre un but.

Contre la première impression qu'elles pourraient donner, les techniques numériques en découragent et en dispensent. Elles paraissent au départ exiger une rigueur méthodologique contraire à tout amateurisme, mais elles nous incitent très vite à nous arranger et à tricher. Pour cela cependant, pour que ça marche malgré tout, on doit quand même avoir quelques lumières sur ce qu'on fait.

Pourquoi l'informatique est-elle ainsi ? Pour une raison qui est au moins évidente si elle n'est pas la seule. Utiliser un programme est déjà programmer, comme on l'a vu, même si c'est sans le savoir, et plus encore régler des préférences et enregistrer des scripts. À partir de là, il y a peu pour corriger directement du code, au moins celui de scripts. Bref, il n'y a aucune limite bien étanche entre le producteur et l'utilisateur de programme.

On modifie les programmes selon la façon dont on travaille. On suggère des modifications aux développeurs, voire on en propose le code. Beaucoup de logiciels sont en source libre et peuvent être modifiés collégialement. Si jamais des options divergent, deux groupes peuvent partir chacun de leur côté.

Il est souvent préférable d'utiliser plusieurs petits programmes qui ne font qu'une chose, mais qui sont très paramétrables pour la faire exactement comme nous voulons, que de plus gros qui imposent leur méthodologie, et ne font rien à notre goût. Les meilleurs programmes ont souvent été écrits pas une seule personne pour répondre à ses propres exigences. Les exemples n'en manquent pas. Ils ne sont pas nécessairement écrits par les meilleurs programmeurs, mais par des gens qui ont une conception très ingénieuse des opérations qu'ils veulent exécuter.

### *L'édition en temps-réel*

Il n'est donc pas de méthode bien définie pour publier en ligne un travail en cours en temps-réel. On devra dans tous les cas exporter passage après passage, puis réexporter le passage corrigé, et recommencer. Il est donc important de trouver des programmes qui font cela le mieux possible et qui épargnent de fastidieuses retouches.

L'entraînement qui ne manque pas d'en résulter permet à lui seul de procéder assez vite, mais pour cela il est au moins nécessaire de compartimenter au mieux : son modèle de page ; ses feuilles de style ; ses fichiers exportés. Il est plus simple alors de retoucher chacun séparément. Dans la pratique, ce travail continu est moins fastidieux que la publication finale d'un ouvrage complet.

### Le nettoyage du code

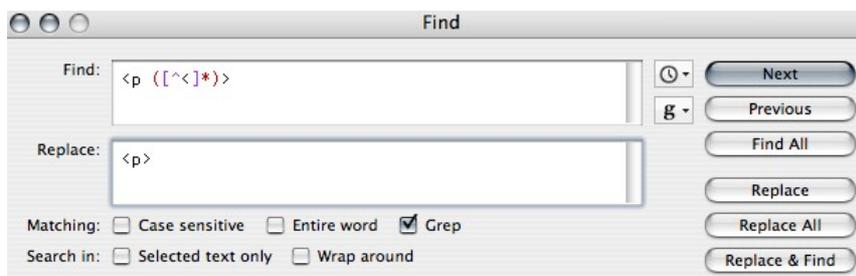
L'idéal serait de pouvoir exporter un texte dans un HTML qui n'ait d'autres identifiants que des balises simples (H1, H2,... P, etc.) sans CLASS ni SPAN, et moins encore FONT, seulement quelques balises de formatages comme <i> (italique) ; <b> (gras) ; <small> (plus petit) ; <sup> (exposant)...

Pour cela, on ne devrait utiliser dans le traitement de texte que des styles biens définis : *Corps de texte*, *Titre 1*, *Titre 2*, etc. Malheureusement, les traitements de texte cherchent toujours plus à trop bien faire. Ils multiplient pour un rien les identifiants et les variations dans les feuilles de style internes. À l'évidence, les programmeurs répondent à des demandes : celles que la page en ligne ressemble le plus exactement possible à celle du traitement de texte, ce qui n'est pas futé.

Filterer son texte d'un excédent de balises de style peut être une épreuve. Pour l'accomplir, les bons éditeurs de texte disposent d'outils puissants. Il est possible de rechercher et de remplacer des formules variables.

Un texte bien saisi ne devrait avoir besoin de rien d'autre qu'une balise <p> en début de chaque paragraphe. Pour effacer tous les autres contenus éventuels d'une telle balise, on utilise la formule « ([^<]\*) ».

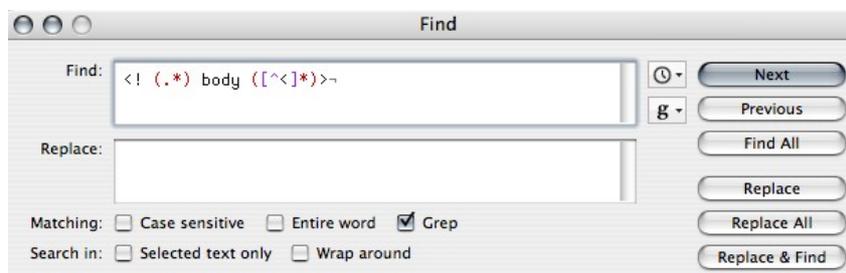
On recherche donc : « <p ([^<]\*)> » et on le remplace par : « <p> ».



On peut recommencer l'opération pour toutes les balises où elle est nécessaire. Si l'on doit les répéter à chaque exportation, il sera préférable d'enregistrer un script. Il suffira alors de le lancer à l'ouverture du fichier exporté en HTML, pour que toutes ces opérations soient effectuées en un seul clic.

Pendant qu'on y est, on peut même supprimer tout le haut et tout le bas de la page, du DDT, à la balise <BODY>, et de la balise </BODY> à </HTML>.

Dans ce cas, on recherche d'abord : « <! (.\*) body ([^<]\*)>- » et on ne le remplace par rien ; puis : « </body> (.\*) </html> ».



On enregistre ces opérations dans le même script, et l'on n'a plus qu'à enregistrer ce qui reste de la page pour le coller tel quel dans la partie de celle où il doit prendre place. (Il est à noter que des éditeurs de texte permettent d'effectuer cette opération sur le contenu de dossiers entiers.)

La totalité de l'opération ne devrait alors pas prendre plus d'une minute lorsqu'on l'aura déjà exécutée deux ou trois fois.

### *Évaluer le plus simple*

Comme on le voit, le numérique demande souvent des opérations mentales assez considérables, mais pour ne plus avoir ensuite à s'en soucier. Il appartient toujours à chacun d'évaluer si de tels efforts valent bien ceux qu'ils économisent.

Comme le précédent procédé évite justement d'y revenir par la suite, on va rapidement oublier comment s'y prendre. Qu'on soit débutant ou confirmé, cet effort restera toujours à peu près égal si l'on doit le refaire. On cherchera dans les aides et les manuels, et l'on s'estimera heureux si l'on parvient à préparer ses scripts en seulement une heure ou deux. Ils nous feront peut-être gagner dix minutes pour chaque utilisation. Cela n'en vaut pas la peine si l'on ne s'en sert que trois ou quatre fois. Peut-être vaut-il mieux abandonner l'idée de trop bien contrôler son code, et utiliser ses pages comme elles sont exportées. Il suffira d'exécuter l'opération contraire : coller le haut et le bas de la page modèle dans celle qui est exportée, et ne plus se soucier de rien.

Publier en temps-réel suppose au contraire la répétition fréquente de ces opérations. Dans ce cas, il devient avantageux de faire l'effort d'automatiser une procédure qui se traduira dans la durée par un moindre effort.

## **LA COLLABORATION**

La qualité de nos écrits dépend pour l'essentiel de la collaboration. Les autres nous apportent des compléments, voient des faiblesses qui nous échappent, forcent à mieux expliquer, tirent des conclusions inattendues, débusquent des coquilles, des contre-vérités, des approximations, des déductions viciées, etc. Ces collaborations peuvent utiliser des canaux variés. Le plus simple est le courriel privé, bien suffisant la plupart du temps.

### ***Le courriel***

Le courrier électronique n'a rien à envier à celui sous enveloppe. Il est plus rapide, mais la poste n'est pas très lente non plus, et le temps mis à lire et à répondre est, lui, difficilement compressible. Il est surtout plus rapide pour composer, adresser, envoyer, répondre, trier, ranger, conserver, garder sous la main, rechercher, faire des envois multiples et quasiment gratuit ; aussi a-t-on tendance à en faire un usage intensif.

Il semble devenir inutile de faire en sorte que le courrier occupe le moins d'espace mémoire possible, avec l'accroissement des supports de stockage, mais rien n'est moins sûr. Comment réduire la taille de ses messages ? Les logiciels de courrier sont généralement réglés par défaut sur le HTML. Il suffit de changer cela dans les préférences. En texte brut, même une très longue lettre occupera rarement plus de 10 ko. Il n'en va pas seulement de l'espace pris sur le disque, du temps passé à relever son courrier ou de la possibilité de l'archiver sur des supports amovibles, mais surtout de sa lisibilité. Ce fut dès le début une forme de politesse, et elle demeure valable.

### ***Pièces-jointes et URL***

Le HTML permet de conserver des styles, ou des différences de couleur. Il est utile, par exemple, pour indiquer par courriel des corrections apportées à un texte. Dans ce cas, il est

préférable de l'envoyer en pièce jointe, ou, mieux, de le laisser sur son propre site et de n'envoyer que l'URL. Il est dans tous les cas mal venu d'envoyer un courrier en HTML à quelqu'un qu'on ne connaît pas ou sans qu'il l'ait sollicité, à plus forte raison s'il contient des images et des fonds de page.

Si on laisse sur son serveur plusieurs documents volumineux à télécharger, il est bon de les compresser. Le clic sur le courriel lancera le téléchargement qui, une fois fini, déclenchera automatiquement le logiciel de décompression, et le correspondant trouvera sur son bureau un fichier compressé « machin.zip », et un dossier décompressé : « dossier machin ». Il utilisera ce dossier quand et comme bon lui semble, et il pourra l'archiver indépendamment de son courrier.

Il est à éviter d'envoyer en pièces jointes des fichiers dans un format propriétaire, sauf, éventuellement, si l'on est sûr que son correspondant utilise exactement les mêmes outils.

### *La correspondance électronique reste de la correspondance*

Le courrier informatique et sur papier n'ont rien à s'envier. L'usage aisé et intensif du premier, surtout pour les échanges en temps-réel, tend parfois à provoquer un relâchement de la langue. Parfois aussi, des problèmes de reconnaissance de caractères spéciaux donnent l'apparence d'un tel relâchement. Qu'on ne s'y laisse pourtant pas tromper. La langue relâchée et les tournures fautives ont le même effet sur un courriel que sous un pli cacheté, si ce n'est pire.

La grande quantité de courrier électronique demande surtout de la concision et la clarté, que facilitent d'ailleurs les moyens de correction à l'écran et les outils linguistiques. Il est vrai que dans une liste de discussion, je ne reprocherai pas à mon correspondant de ne pas avoir fait usage d'un correcteur grammatical s'il résout mon problème en trois minutes, mais, dans l'ensemble, la langue que je lis dans mes courriels n'a pas de raison d'être moins soutenue que celle de mon courrier sur papier.

Le courrier n'est pas une forme mineure et négligeable d'écriture, voire de littérature. Si une très grande part du courrier électronique doit finir à la poubelle, mais pas proportionnellement supérieure à celle du courrier sur papier, une part qualitativement plus importante est la matière première d'écritures ultérieures. Le courrier électronique trouve là sa réelle supériorité sur le papier : il prend moins d'espace en pouvant être stocké sur des supports amovibles, il est plus facile de le classer et beaucoup plus facile d'y retrouver ce que l'on cherche, et plus encore d'en reprendre des passages par simple copier-coller.

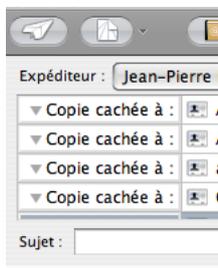
Comme nous recevons nos courriels en texte brut ou en HTML, il est indépendant de toute application, et dans une moindre mesure du système. Il est donc très facile à éditer. Pour cela, on doit au moins savoir où on le classe, notamment pour le sauvegarder. Il n'est pas toujours évident de trouver où sont exactement placés les fichiers qui s'ouvrent dans notre logiciel de courrier. On peut les trouver dans le dossier de l'application elle-même, dans les préférences du dossier système, ou encore à la racine de son disque dur. Il est utile de savoir qu'on peut remplacer ces dossiers par des alias si l'on souhaite les ranger dans des endroits plus accessibles, pour en faire des sauvegardes régulières, dans son dossier « courrier » par exemple.

La plupart des logiciels de courrier permettent de faire des filtres pour trier automatiquement sa correspondance dans des dossiers distincts, selon leur objet ou le nom de l'expéditeur.

### *Les envois multiples*

Il est possible d'envoyer le même courriel à plusieurs personnes en même temps. C'est très commode pour travailler ensemble. Pour un groupe structuré qui collabore de façon continue, il est utile de créer un dossier dans son carnet d'adresse. On pourra envoyer un courrier au groupe entier en sélectionnant ce dossier ou en saisissant son nom.

Quand on fait des envois multiples, il est important de ne pas envoyer trainer n'importe où les adresses de ses correspondants. Même si nous sommes certains qu'aucun d'eux ne négociera ces adresses ou n'enverra du *spam* ; nous ne pouvons jamais savoir où ces adresses pourraient finir par se retrouver. Il est préférable de cacher les noms des destinataires, à plus forte raison s'ils n'ont pas de raison de se connaître.



Souvent, « copie cachée » n'est pas écrit en toutes lettres, mais « CCI » (copie carbone invisible), à ne pas confondre avec « CC ».

## Les outils de coopération

### *Liste de diffusion et forum*

Le site demeure le meilleur moyen d'écrire et d'échanger en ligne, et le courriel, le meilleur moyen pour communiquer. Pour un travail collaboratif continu, il peut être avantageux de créer une liste de diffusion : chacun s'y inscrit indépendamment, et reçoit son courrier sans connaître, sauf le modérateur, les adresses de tous les abonnés.

Très proche de la liste de diffusion est le forum. Plutôt que de recevoir, lire et écrire du courrier, on lit et écrit directement en ligne sur les pages du forum. Comme une liste de diffusion, la lecture d'un forum peut être ouverte au public ou privée. Pour écrire dans une liste ou sur un forum, il est toujours nécessaire de s'inscrire.

Le principal avantage du forum est de permettre plusieurs rubriques correspondant à différents sujets de discussion.

### *Le blog*

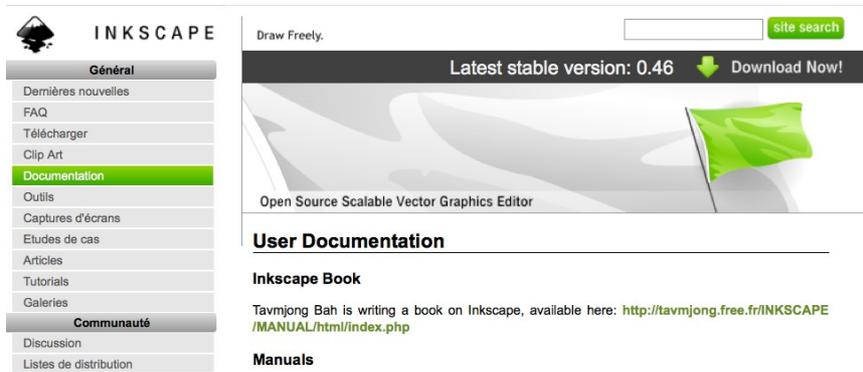
Le blog est un peu le site du pauvre. Il peut suffire de s'inscrire chez des fournisseurs qui offrent des blogs clé en main, tout prêts, dont on peut modifier plus ou moins légèrement l'apparence. Sinon on peut installer son propre blog n'importe où.

Un blog ne peut contenir que très peu de rubriques, ce qui le limite pour classer beaucoup de contenu. Il gère principalement l'ordre chronologique, ce qui en fait un bon outil d'information. Les articles se placent à la suite des autres, le dernier en haut de la page. C'est pratique pour diffuser les dernières nouvelles, mais pas pour conserver des documents sur la durée.

Il offre aussi la possibilité d'ajouter aisément des commentaires aux articles. Tout cela peut en faire un bon outil de collaboration, avec des droits d'administration partageables ou non. Chaque article peut être rendu public ou réservé à ceux qui sont référencés, ainsi que la possibilité d'y laisser des commentaires.

### Le wiki

Le Wiki est une sorte d'intermédiaire entre le site et le blog. Il peut contenir un nombre illimité de pages hiérarchisées avec des liens internes. Il permet à chacun, s'il en a l'autorisation, d'éditer librement les pages existantes ou d'en créer de nouvelles. Le modèle du wiki est Wikipedia, qui donne la mesure de sa puissance.

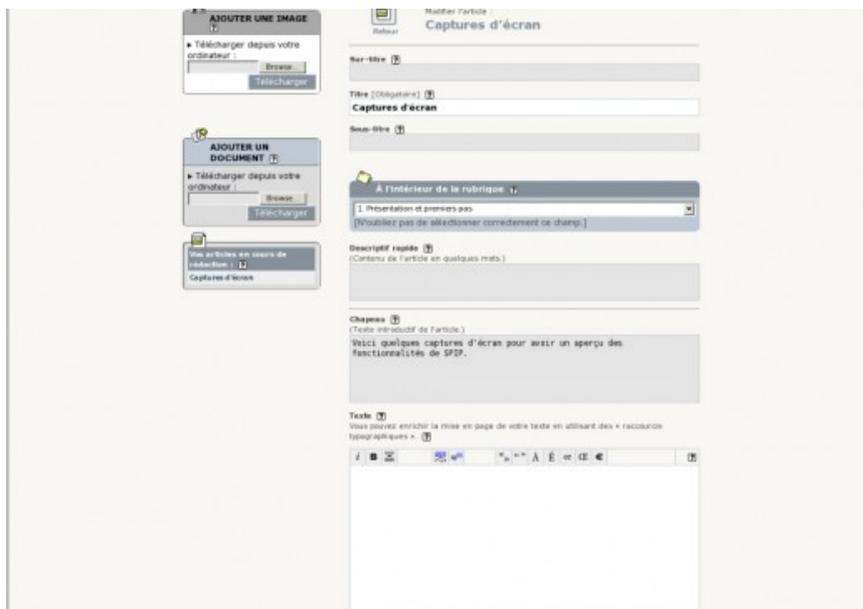


Le wiki est le moyen le plus simple de faire un site collaboratif. Sa présentation reste cependant très spartiate quoi qu'on y fasse.

### Le système de gestion de contenu ou CMS

Il est très malcommode de gérer un site à plusieurs. Qu'il offre une image personnelle, anonyme ou institutionnelle, il est toujours entre les mains d'un webmestre. On peut avoir de bonnes raisons de ne pas s'en contenter.

Des CMS (*Content Management System*) peuvent alors être très efficaces et permettre à tous les membres référencés d'éditer eux-mêmes leurs articles comme s'ils étaient sur leur propre site.



On voit ci-dessus l'interface Spip d'édition d'un article.

Il existe des quantités de solutions pour créer un site collaboratif, parmi lesquelles se brouillent les distinctions bien nettes entre *blog* (textes à la suite dans un ordre inversement chronologique et possibilité de commentaires), *wiki* (où tout visiteur est virtuellement collaborateur) ou *CMS* (site collaboratif). Le code source lisible et les licences libres permettent une floraison de programmes et d'extensions tous plus ingénieux les uns que les autres. Le plus difficile est de savoir ce que l'on veut et d'y trouver les outils les mieux adaptés.

Il est cependant des difficultés plus subtiles. Il n'est d'abord pas évident qu'un texte consciencieusement écrit et publié suggère des réactions bien intéressantes à chaud. La facilité avec laquelle nous pouvons écrire un commentaire public nous le fait souvent juger consternant si nous le relisons plus tard. À l'inverse, une critique ou une réflexion bien pensées et bien écrites peuvent sans peine être publiées en ligne avec un lien sur le texte cible, ou transmises par un courriel qui est aussi bien publiable sur le site de celui qui le reçoit que de celui qui l'envoie.

Nous ne sommes pas toujours les meilleurs juges de ce que nous disons. Parfois nous n'en percevons ni la portée, ni la pertinence, ou encore ni les erreurs ni les limites. Elles nous sont le plus souvent révélées par des publications critiques ou des courriers privés, alors que des commentaires rapides les masquent au contraire.

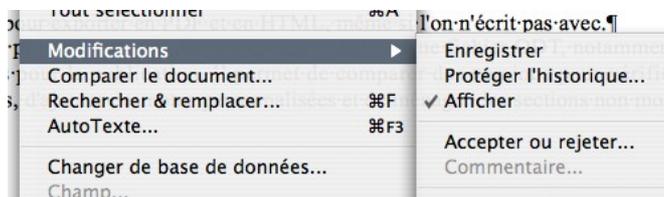
Travailler à plusieurs sur un même site est utile, mais à la condition que chacun sache déjà tirer tout le parti de ses propres outils. Dans le cas contraire, on ne doit pas entretenir trop d'illusion pour faire de ces moyens des palliatifs aux insuffisances d'un collaborateur. La difficulté est alors que ces nouveaux moyens ne l'empêchent pas d'utiliser et de cultiver les siens propres.

### *Le dossier de travail*

La meilleure façon de travailler ensemble peut être le partage du même fichier de travail, c'est-à-dire celui sur lequel on écrit avec son traitement de texte. Pour cela, on doit utiliser le même traitement de texte, ou bien en trouver qui soient très compatibles, et bien en gérer les formats.

Les fichiers Microsoft Office sont souvent utilisés ainsi, dans la mesure où, en dominant le marché, leurs formats peuvent faire figure de standard. Toutefois la suite Microsoft Office est chère, elle ne fonctionne pas sur tous les systèmes et son code n'est pas très limpide. Open Office est donc un meilleur candidat. Il est gratuit et s'installe partout. Personne n'a donc de raisons irréfutables de refuser de l'installer, sauf peut-être un espace disque un peu étroit. Open Office peut, de plus, rendre des services pour exporter en PDF et en HTML, même si l'on n'écrit pas avec.

Il est très pratique de travailler à plusieurs sur le même fichier ODT, notamment à préparer des épreuves pour la publication. Il permet de comparer des versions pour vérifier les dernières modifications, d'ajouter des notes personnalisées et de ménager des sections non modifiables.



On peut légitimement avoir des réticences à trop généraliser un tel usage. Même si le programme est libre, ce n'est jamais une trop bonne idée de s'en rendre excessivement dépendant, ni d'en imposer l'utilisation à un collaborateur ; de lui interdire d'employer ses propres outils comme il l'entend, même pour un travail en commun. Il est toujours préférable, même pour une collaboration, de laisser chacun libre de suivre sa propre voie.

## *Le site*

### *Comment ne pas rendre aisément accessible tout ce qu'on publie en ligne*

Il est difficile d'utiliser pleinement les ressources de l'internet sans avoir son propre site, au moins sous la forme d'un espace de sauvegarde sans accès public. On trouve cependant peu d'hébergeurs qui les offrent gratuitement, contrairement aux sites publics. La différence entre la publication et la sauvegarde en ligne est de toute façon relative. La publication est dans tous les cas une forme de sauvegarde, et il est toujours possible de protéger certains dossiers. Si des fichiers ne sont pas indexés, et si aucun lien ne pointe sur eux, il y a bien peu de chances que jamais quelqu'un ne les ouvre.

La sauvegarde en ligne de données confidentielles est de toute façon peu recommandée, et ce qui n'est pas public à proprement parler, peut toujours être partagé.

### *Pourquoi ne pas rendre aisément accessible tout ce qu'on publie en ligne*

L'intérêt de ne pas rendre trop accessible tout ce qu'on publie en ligne n'est presque jamais personnel, car, si l'on a des secrets, il vaut mieux les ranger ailleurs. L'intérêt est d'abord celui des autres internautes. Il est inutile de les attirer sur ses pages si l'on pense qu'elles ne sont pas assez abouties, si l'on n'est pas sûr de ce qu'on y dit, ou, plus généralement, si elles n'apportent rien au commun des mortels.

L'intérêt est aussi pour ses collaborateurs. Il est bien rare que des gens s'associent pour un projet sans avoir le souci qu'il ne soit pas éventé prématurément, que ce soit pour les pires ou les meilleures raisons. Même quand ce souci n'a aucune raison d'être, une certaine confidentialité favorise l'intimité.

Toutes ces raisons exceptées, il n'y a pas de raison de craindre les visites. Les retours sont généralement plus intéressants qu'on aurait pu l'espérer, et des rencontres peuvent avoir lieu qui auraient été impossibles autrement.

### *Comment rendre intelligemment son site visible*

Il n'y a aucune raison de forcer la visibilité de son site à l'aide de tous les petits secrets techniques que l'on trouve partout. La plupart des moteurs de recherche, dont Google, font assez intelligemment leur travail. Il importe seulement d'optimiser les moyens de trouver chez soi ce qu'on peut y chercher — et certainement pas de faire perdre son temps au plus grand nombre.

Lorsqu'on fait une recherche en ligne, le premier critère de pertinence d'une page est la quantité de liens entrants et sortants sur le même sujet. Si nous cherchons, par exemple, des informations sur la boule provençale, le site le plus pertinent sera sans-doute celui qui aura le plus de liens entrants et sortants sur la boule provençale. Si les liens concernent aussi la cuisine provençale ou la boule lyonnaise, il le sera certainement moins, sauf peut-être si l'on s'intéresse à la Provence ou aux boules en général, mais dans ce cas, on n'aura pas fait une recherche sur la boule provençale. En somme, pour être bien référencé, il suffit de donner de bonnes références.

Sauf à louer des espaces publicitaires, on n'a pas de raison d'attirer sur ses pages plus de monde que nécessaire. Il y en a beaucoup plus à attirer spécifiquement ceux qui y trouveront satisfaction et pourront nous renvoyer des retours intéressants. C'est d'ailleurs vrai même dans le cas de location d'espaces publicitaires, qui seront toujours plus intéressants — pour l'offre comme pour la demande — s'ils sont très bien ciblés.



texte s'affichera, de préférence dans une autre fenêtre. Si je renvoie au site de l'auteur, il n'y aura pas d'ambiguïté, et moins encore si je l'écris en toutes lettres : <http://jdepetrus.free.fr/>, ce qui sera tout particulièrement utile si le texte doit être imprimé.

Les liens ancrés avec ambiguïté font perdre du temps en surfant en vain. Voir à ce propos La synecdoque et le lien hypertexte de Bernard Lombard : <http://www.synec-doc.be/doc/tropes.html>.

### *Les liens et l'écriture*

Les liens ne concernent pas seulement la lecture, mais surtout l'écriture, dans la mesure où un net découpage entre les deux serait pertinent. Surévalués pour le langage HTML, ils sont souvent négligés pour les autres états du texte, traitements de texte ou fichiers PDF, qui permettent toute sorte de liens internes ou externes. Les alias de fichiers ou de dossiers sont eux-mêmes des liens qui permettent d'ouvrir un même document sans qu'il soit nécessaire d'en multiplier les copies. *L'abonnement* est une technique comparable, elle aussi, au lien. Il permet que les corrections apportées à un document soient automatiquement reportées sur un autre. Une correction faite sur un programme de PAO sera ainsi reportée en temps-réel sur le traitement de texte, ou inversement.

### *Les liens et la sauvegarde*

Il est essentiel de faire des sauvegardes, mais il est déconseillé de multiplier les copies d'un même document. Les sauvegardes doivent être faites sur un autre support que son disque dur pour qu'elles ne soient pas détruites en cas d'effacement de celui-ci. Il est préférable que celles-ci se trouvent même dans un autre lieu. Un site sur un serveur distant peut être une bonne solution pour la sauvegarde. Elle permet notamment d'avoir accès à son travail où que l'on se trouve à partir de n'importe quel ordinateur.

Mettre un document en ligne n'impose en rien de le rendre public. Si aucun lien n'y renvoie, personne n'y aura accès. On aura de plus cette possibilité d'en proposer la lecture à un correspondant en joignant son URL à un courrier.

Pour garantir qu'un document ne soit pas indexé par un moteur de recherche si l'on ne le souhaite pas, on peut, comme on l'a déjà vu, utiliser des méta tags.

`<META name="robots" content="noindex">` (Interdit l'indexation)

`<META name="robots" content="nofollow">` (Interdit de poursuivre les liens)

Ceci peut s'automatiser, en utilisant un modèle, par exemple. Compresser un fichier garantit aussi qu'il ne sera pas indexé. On peut encore protéger un dossier par un mot de passe. Si l'on n'a pas de grands secrets à cacher, ces procédés sont bien suffisants.

Il peut également être justifié de créer des sites miroirs. On emploiera alors des outils de synchronisation automatique.

Il est un peu moins avisé de dupliquer ou de rééditer sur plusieurs sites les mêmes documents, les siens ou ceux d'un tiers. Il est inutile d'éditer sur son site un document en libre accès sur un autre, ou seulement de le copier, quand un lien suffirait à l'ouvrir. Il ne pourrait plus alors être mis à jour. Il est impératif alors d'indiquer au moins l'adresse du document original.

### *Sur la propriété des auteurs*

Les textes sont librement lisibles et peuvent être rapatriés sur le web ; ils n'en demeurent pas moins propriété des auteurs. Quand bien même les payerait-on, ou les achèterait-on sur papier qu'ils n'en deviendraient pas les nôtres. Pas plus qu'ailleurs, on ne doit donc omettre de citer les auteurs et les références de ses emprunts.

Beaucoup d'œuvres sont en « copyleft », c'est-à-dire qu'on peut les copier et les modifier librement, mais l'obligation demeure d'en citer les auteurs et de permettre l'accès direct à l'original.

La facilité avec laquelle on peut rechercher, importer ou copier sur le net ne menace pas réellement nos droits d'auteurs, si nous ne les mettons pas nous-mêmes en danger. Il suffit de ne pas laisser inconsidérément traîner des documents incomplets : des documents qui ne seraient reliés à rien, dépourvus de date ou de nom d'auteur. Tout document devrait contenir le nom de son auteur (ou ce qui en tient lieu, pseudonyme, login, logo) et au moins un lien sur une page d'accueil. Il devrait être daté (si ce n'est contenir un numéro de version et un copyright). Tout ceci peut, là encore être automatisé.

Rien n'est plus agaçant, quand on a importé un texte intéressant, ou qu'on l'a imprimé, de ne plus retrouver son origine, de ne pas pouvoir en exploiter une citation et la référencer, ou contacter son auteur. Ceci est vrai pour un document rendu public, mais aussi pour tout document qui pourrait le devenir accidentellement, ou même pour son usage privé.

## ***CONCLUSIONS PROVISOIRES***

L'internet n'a pas vocation à prendre la suite des places de marché et des centres commerciaux. S'il offre toutes les facilités de commandes en ligne et de paiement sécurisé, il ne propose là rien de fondamentalement plus nouveau que l'achat par catalogue. Les marchandises devront bien finalement être livrées physiquement. Quant à la possibilité de vendre des produits spécifiquement numériques et indépendants de tout support, ce n'est pas dans son essence ni dans sa nature profonde, et son développement met plutôt les prémices d'un tel commerce en danger.



Pour des quantités de raisons, il est impossible de donner un prix à un produit numérique. Et quand on veut en justifier un, on est obligé d'en chercher des raisons qui n'avaient jamais été évoquées avant. Quoiqu'on fasse, l'objet numérique ne s'intègre pas dans les coutumes marchandes qui remontent au néolithique.

Il apparaît tout à fait normal et conforme à d'antiques mœurs, de payer un CD, un livre, une calculatrice ou une règle à calcul, une machine à écrire, une consultation de voyage, un acte notarié, une place de cinéma, etc. Chacun perçoit au moins confusément, mais avec certitude, le coût de ce qu'il achète, et la part relative, souvent négligeable, qui revient à celui qui vend et aux différents acteurs. Rien de tel avec un document ou un programme numérique.

Chacun comprend, par exemple, que plus un livre est vendu, plus il rapporte à son éditeur et à son auteur ; mais chacun comprend aussi que les coûts, en l'occurrence d'impression et de distribution, grimpent aussi en proportion. La vente de documents ou de programmes numériques

ne fait varier aucun coût. Bien des logiciels, d'ailleurs, quand ils ont atteint un seuil d'amortissement, sont offerts, et parfois même leur code, pour être librement modifié.

On va invoquer pour des produits numériques le prix légitime d'un travail, mais on ne l'avait jamais fait avant. C'est le prix de la richesse produite qui a, au contraire, toujours justifié celui du travail, et qui justifiait même qu'il s'agissait bien de travail, et non de passe-temps, de loisir. À partir de quoi justifierait-on et mesurerait-on le prix d'un travail ?

Il est à peu près impossible d'empêcher d'utiliser un produit numérique sans le payer. Même sans en craquer le code, il est toujours possible de faire circuler celui qui a été acheté. La seule chose qui soit vendable, serait l'assistance et la fourniture de mises-à-jour, si ce n'est qu'elles peuvent être assurées bien souvent par les utilisateurs eux-mêmes. Dans le numérique, en somme, comme dans les aspects les plus authentiques de la vie, il n'est rien de plus indécidable que qui donne et qui reçoit.

Prenons pour exemple le texte que je suis en ce moment-même en train d'écrire, et supposons-lui un lecteur qui n'ait rien d'autre à en faire que le lire. Autant dire qu'il n'a rien à en faire du tout, qu'il n'obtient en réalité rien de moi, et moi-même, pas davantage de sa lecture. Il n'y a tout simplement pas d'échange. S'il parvient au contraire à faire un usage intelligent de mon travail, sans qu'il n'ait rien à me donner, je finirai bien par en retirer quelque chose, au moins indirectement.

Il se peut aussi qu'il se donne la peine de me corriger, me compléter, m'apprendre des choses que j'ignore, ou encore, qu'il trouve chez moi les éléments d'un travail plus important et plus intéressant. Nous aurons de toute façon bien du mal lui et moi à faire la part de ce que chacun reçoit et donne. Même si nous correspondons, nous ne saurons jamais quand nous faisons naître chez l'autre une idée fertile et décisive, ou un haussement agacé d'épaule.

On néglige certainement trop que le numérique et l'internet sont avant tout des outils de travail, bien avant d'être des moyens de commerce, ou de loisir, et même de communication. On reconnaît au mieux un minimum d'importance à la production et à l'échange de documents, mais on néglige totalement la capacité de produire des énoncés subtils et développés ; l'importance de la réflexion et de la recherche plume en main — ou clavier sous les doigts — à l'aide d'une langue soutenue à la grammaire complexe. On déblaie souvent plus de terrain à l'aide de courriels et de pages web bien écrits et bien pensés, qu'à travers des rencontres de visu.

La personne ou le groupe de travail optent souvent d'abord pour laisser l'outil numérique, c'est-à-dire le plus puissant outil pour penser, à la charge de spécialistes. Cette attitude, naturelle au début, peut devenir assez vite stérile. Elle a d'ailleurs été rapidement dépassée par des groupes de travail efficaces. Les termes « intelligence artificielle » n'ont de sens que si on les comprend comme des prothèses destinées à accroître les capacités de ceux qui les utilisent. Le meilleur spécialiste ne peut savoir bien répondre aux questions qui ne sont pas les siennes, et attendre après lui revient à perdre les principaux avantages des outils numériques. Ces attitudes nourrissent et se nourrissent à leur tour d'une conception erronée du travail. Il appartient à l'homme de concevoir et de décider de ses procédures, et non de leur obéir, comme il appartient au berger de commander ses chiens. Le seul fait de ranger les outils pour l'écriture sous la rubrique « bureautique » dit tout de leur mécompréhension.

L'informatique implique certains efforts, un effort d'apprentissage toujours renouvelé d'abord, et aussi un effort d'entretien, de petite programmation, de sauvegarde, de mise-à-jour des programmes et des données. Ces opérations sont si variées, et parfois si rares, qu'on ne peut espérer se souvenir de tout sans être contraint perpétuellement de rechercher des informations. Ces efforts permanents sont assez lourds, et ils ne se justifient qu'à faire finalement gagner du temps et de l'attention en démultipliant la puissance de travail.

Tout cela ne peut fonctionner qu'avec une réelle autonomie personnelle. On peut toujours abandonner ces efforts et cette autonomie à de nombreuses offres de services en ligne ou non. Il se peut bien qu'on doive renoncer alors à la puissance de travail. À vrai dire, la question reste ouverte, mais comme questions techniques.

L'autonomie personnelle et le développement des techniques numériques ont parties-liées et se font réciproquement fins et moyens. Tout encourage et pousse chacun à constituer sa propre boîte à outils, à l'entretenir et la parfaire au fil des ans, à la personnaliser et l'adapter à sa façon de travailler et même de penser. Il n'est profitable à personne de lui imposer le contraire ; s'adapter à des outils et des méthodes.

Cette personnalisation ne s'oppose évidemment pas à la coopération et au partage. Bien au contraire, c'est comme si l'on disait que cultiver son style s'oppose à communiquer. Bien sûr, cette personnalisation englobe les techniques de portabilité et d'ouverture.

Cela suppose une entreprise à longue-portée, peut-être moins un gros travail qu'une posture sur le long-terme. Or tout le monde ne peut épouser une telle attitude, soit parce qu'il faut bien commencer un jour, soit parce que tout le monde n'a pas un usage du numérique qui la rende nécessaire.

C'est un double problème inhérent aux techniques du numérique, comme à tout usage de signes d'ailleurs, et donc à toute activité humaine, que de permettre son acquisition sans que ce soit au prix de limiter la personnalisation et le perfectionnement des autres. À l'évidence, cette double contrainte ne peut être pensée en termes d'opposition et de recherche de compromis, mais au contraire de perspectives convergentes.

# Bibliographie

- [http://jdepetris.free.fr/load/lettres\\_numeriques.html](http://jdepetris.free.fr/load/lettres_numeriques.html) *Des lettres numériques*, Jean-Pierre Depetris
- [http://jdepetris.free.fr/load/pc\\_fe/numerique/numeric.html](http://jdepetris.free.fr/load/pc_fe/numerique/numeric.html) Remarques provisoires sur le numériques, Jean-Pierre Depetris
- <http://jdepetris.free.fr/load/rapport.html> *Rapport sur l'édition en ligne*, Jean-Pierre Depetris
- [http://jdepetris.free.fr/pages/sections\\_ooo/sections\\_ooo.html](http://jdepetris.free.fr/pages/sections_ooo/sections_ooo.html) *Comment on fait des sections (chapitres) avec Open Office ?* Jean-Pierre Depetris
- <http://www.sri.ucl.ac.be/SRI/rfc1855.fr.html> *Les règles de la Netiquette*. Traduction française du Network Working Group <http://tools.ietf.org/html/rfc1855> Netiquette Guidelines d'octobre 1995. Cette traduction a été réalisée par [Jean-Pierre Kuypers](#).
- <http://www.faqs.org/docs/artu/> *The Art of Unix Programming*, Eric Steven Raymond
- [http://www.linux-france.org/article/these/cathedrale-bazar/cathedrale-bazar\\_monoblock.html](http://www.linux-france.org/article/these/cathedrale-bazar/cathedrale-bazar_monoblock.html) *La cathédrale et le bazar*, Eric Steven Raymond
- Typographie française en ligne* [http://www.synapse-fr.com/typographie/TTM\\_0.htm](http://www.synapse-fr.com/typographie/TTM_0.htm)
- <http://www.la-ponctuation.com/index.html> *la-ponctuation.com*
- [http://www.granddictionnaire.com/btml/fra/r\\_motclef/index800\\_1.asp](http://www.granddictionnaire.com/btml/fra/r_motclef/index800_1.asp) *Grand dictionnaire terminologique*, Office québécois de la langue française
- Tutoriel CSS* <http://fr.html.net/tutorials/css/default.asp> HTML.net