

UNIVERSITÉ

de Picardie

Jules Verne

**LA GAZETTE  
DU  
DEUST T.S.I.C.**



EDITORIAL:

Cela fait un moment que nous ne nous sommes pas vu. Et oui, presque une année scolaire s'est écoulée depuis le dernier numéro. Au moment où j'écris ces lignes, nous sommes au mois d'avril 2006 et je travaille presque 24 heures sur 24 sur mon projet professionnel. Et ce projet n'est qu'une partie de mes activités de programmeur web. En effet, il m'arrive de réfléchir un peu et au bout de ma réflexion, je me dis qu'il est parfois temps d'essayer de nouvelles choses. Après avoir longtemps travaillé en HTML, je me suis mis à l'XHTML. J'ai également décidé d'apprendre un tout petit peu le langage PHP que je déteste. Je lui préfère de loin Java qui bien de plus difficile à la base, je le trouve plus simple, plus compréhensible. Au terme de ce numéro, vous pourrez me dire ce que vous en pensez. Car dans ce troisième opus de la Gazette du DEUST, nous allons voir l'ensemble des langages qui existent et qui nous sont fort utiles pour la programmation web. Certains font partis de l'enseignement du DEUST, d'autres non. Mais qu'importe, le but de la gazette est de vous apporter des compléments de cours, comme des petits plus qui vous seront utiles. Pas toujours, mais vous les connaîtrez tout de même.

NUMERO 3 - MAI 2006

SOMMAIRE :

- page 2 : JAVA
- page 4 : VOTRE PREMIER PROGRAMME JAVA
- page 8 : HTML
- page 13 : DHTML/XHTML
- page 16 : JAVASCRIPT
- page 21 : Les Feuilles de Style CSS
- page 26 : JAVA / C comparaison
- page 28 : PHP
- page 30 : JSP Java Server Pages
- page 32 : JSTL
- page 33 : HISTORIQUE DE L'INFORMATIQUE
- page 35 : LA VALIDATION HTML
- page 36 : TRUCS & ASTUCES

LA GAZETTE DU DEUST TSIC est une publication indépendante éditée par EdoWorldConcept, Inc.

Rédacteur en chef et directeur de la publication : DAVID NOGENT.

Direction Artistique ODAE EDOTM.

Remerciements : Emmanuel MARTY, J.L. DELEAGE, Sophie Cazier, Yann DUVAUCHELLE. L'Université de Picardie Jules Verne n'est pas responsable du contenu de cette revue qui n'engage que son auteur.

[http://david.nogent.free.fr/la\\_gazette.html](http://david.nogent.free.fr/la_gazette.html)

[david.nogent@etud.u-picardie.fr](mailto:david.nogent@etud.u-picardie.fr)

(c) 2006 EdoWorldConcept, Inc. Droits Réservés.

# JAVA™

C'est un défi considérable dans lequel je me lance. Quel défi? Et bien, celui de vous présenter ce formidable langage de programmation qu'est Java mais aussi faire de vous des adeptes de ce langage que souvent vous considérez comme trop difficile à appréhender.

JAVA est un langage de programmation développé par Sun Microsystems depuis 1991. Il est à la fois un langage de programmation classique et un langage destiné à un emploi à travers le Web. Il est le langage le plus puissant qu'il puisse exister aujourd'hui. En effet, comme un langage classique (comme Delphi, Visual Basic, C/C++ ou Pascal), Java permet le développement de programmes utilisables sur une machine isolée. Cependant, à la différence des autres langages classiques, Java est indépendant du matériel et du système d'exploitation. Ceci le prédispose donc à une intégration dans un site web. Vous pouvez donc programmer en Java sur un PC Windows, votre programme sera également exécutable sur une machine Unix. Comment cela ? Et bien c'est très simple...

1/ Tout d'abord, vous programmez en Java. Vous obtenez ce que l'on appelle un «Code Source» de forme '\*.java'.

2/ En passant par la commande «javac» (*cf cours JSP de Monsieur J.L. DELEAGE, Module 2 pour ceux qui ont un trou de mémoire*) vous obtenez un fichier 'pré-compilé' que vous allez pouvoir utiliser sur votre plateforme Windows ou sur n'importe qu'elle autre plateforme.

3/ Ce fichier 'pré-compilé' est de la forme '\*.class'. Il va être récupéré par la Machine Virtuelle Java qui est installée sur la machine qui va devoir exécuter votre programme. Cette machine virtuelle va compiler votre programme en le compilant dans un langage compatible avec le matériel et le système d'exploitation. Car la machine virtuelle Java est propre à la machine de l'utilisateur final. Ainsi le développeur n'a pas à se préoccuper de savoir quel système est utilisé par l'utilisateur final.

Ce qui n'est pas le cas de Visual Basic. Si vous programmez en V.B. sous Windows, vous ne pouvez développer que des applications Windows... et jamais vous ne pourrez développer des applications pour Mac ou Linux. Vous saisissez la différence?

Sun Microsystems vise la stratégie suivante et destine Java à travailler avec vous sur votre bureau où il trouvera sa place comme une tasse de bon café.

Voici donc ce qui fait le succès de Java: sa portabilité. Le slogan de Sun est clair: *Write once, run anywhere* (écrivez une fois, exécutez n'importe où).

Mais il y a un autre aspect qui fait que Java rencontre, depuis sa création, un énorme succès. *Si vous ne me croyez pas, regardez les jeux de votre téléphone portable, ils sont écrits en Java.* ;) Cet autre aspect, c'est la sécurité !

Contrairement à certains langages permissifs (dont le langage C est un exemple), Java offre un certain niveau de sécurité qui contribue à sa fiabilité. Cette 'sécurité' consiste en un contrôle des accès aux fichiers et à la mémoire. Ainsi, il est impossible de voir le programme déborder des zones de mémoire et des fichiers qu'il doit normalement traiter.

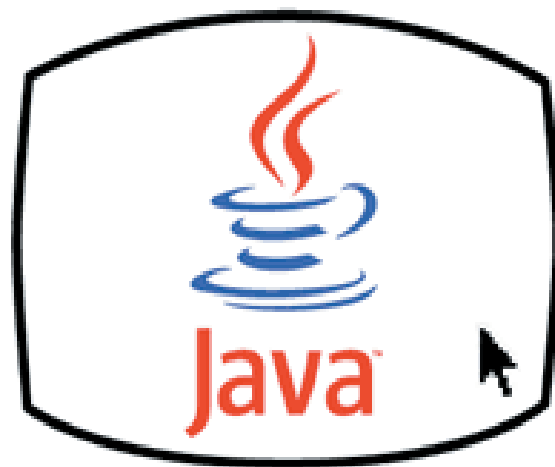
Je dois aussi vous parler d'un autre aspect de Java que beaucoup de personnes oublient: sa simplicité. Contrairement aux 'on dit', Java n'est pas compliqué et est, tout à fait, abordable... Il faudra certes travailler pour le maîtriser mais cela ne relève pas de l'impossible.

Il y a trois manières différentes d'utiliser un programme Java:

1/ une application: c'est un programme autonome comme permettent d'en créer tous les autres langages de programmation. Mais à la différence, que cette application est indépendante de la plate-forme et s'exécute à travers une machine virtuelle Java.

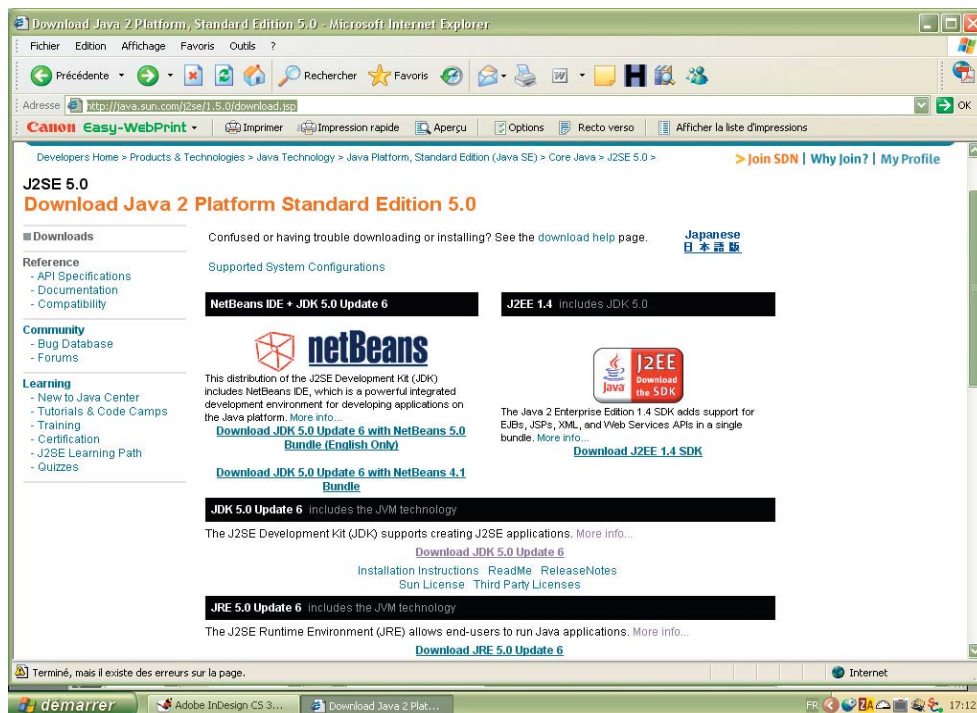
2/ une applet (=application internet): c'est un programme Java destiné à être intégré dans une page web. Lors de la consultation d'une page web contenant une applet, celle-ci est transférée sur l'ordinateur client, puis elle s'exécute à travers la machine virtuelle Java intégrée au navigateur.

3/ une servlet (=server application internet): Comme l'applet, c'est un programme Java intégré à une page web. Cependant, le code java n'est pas envoyé par le serveur vers le poste client. Ce code est exécuté au niveau du serveur, l'ordinateur client n'étant chargé que de l'affichage et de la saisie des informations. C'est le cas de vos pages J.S.P. !



# VOTRE PREMIER PROGRAMME JAVA

On vous a dit que Java était un langage trop difficile à étudier et qu'il valait mieux pour vous y renoncer. Et, je vais vous montrer le contraire. Mais tout d'abord, il nous faut installer le Java Development Kit de Sun Microsystems que vous trouverez à cette adresse: <http://java.sun.com/j2se/1.5.0/download.jsp> .



Une fois le programme téléchargé, installez-le! Veillez à bien noter l'emplacement où vous l'installez car vous en aurez besoin après. Par défaut, l'emplacement est `C:\Program Files\Java\jdk1.5.0_06` .

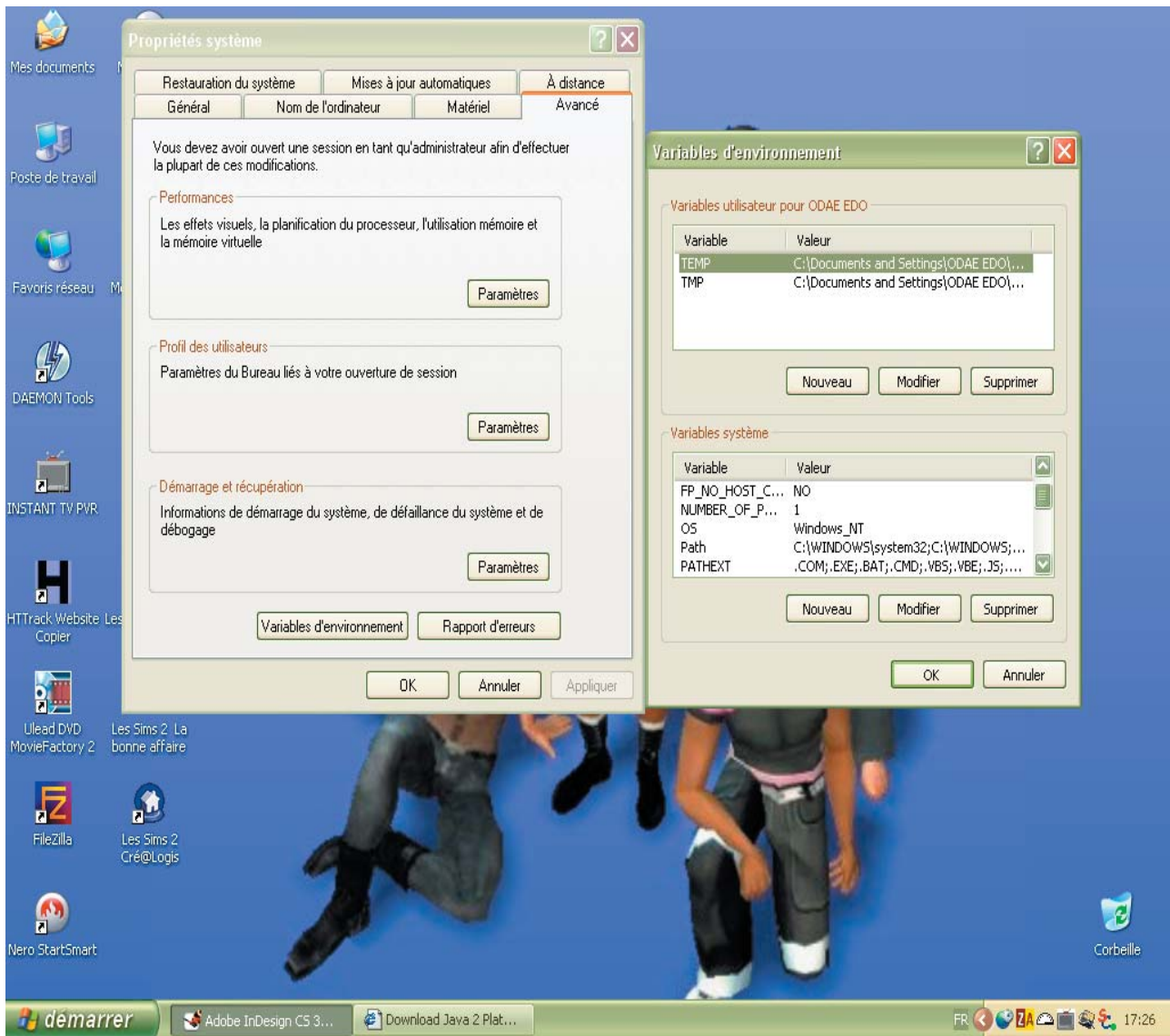
Maintenant que le JDK est installé, il faut réaliser la manipulation suivante (c'est la même manipulation que celle vue dans le cours Java de M. DELEAGE - Module 2) :

DEMARRER > Panneau de configuration > Systèmes

Une fenêtre appelée «Propriétés Système» s'affiche. Cliquez sur l'onglet «Avancé», puis sur «Variables d'environnement».

Vous devez obtenir les mêmes fenêtres que la capture d'écran se trouvant à la page suivante....





La nouvelle fenêtre «Variables d'environnement» s'affiche. Rendez-vous dans la zone «Variables Système», sélectionnez la ligne «Path» et cliquez sur «Modifier». Une nouvelle fenêtre s'ouvre... Dans la zone «Valeur de la variable», vous pouvez apercevoir une ligne... N'y touchez pas! Ne la supprimez pas sous aucun prétexte!

Voici par exemple la valeur de ma variable «Path»:  
`%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem`

Allez à la fin de la ligne (valeur de votre Path), ajoutez un point-virgule et le chemin de la commande «javac»:

valeur de votre path...;C:\Program Files\Java\jdk1.5.0\_06\bin;

Maintenant appuyer sur OK et le tour est joué.

Ouvrez l'Invite de commandes et tapez «javac»

Le message suivant doit apparaître:

Microsoft Windows XP [version 5.1.2600]  
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\ODAE EDO>javac

Usage: javac <options> <source files>

where possible options include:

-g	Generate all debugging info
-g:none	Generate no debugging info
-g: {lines,vars,source}	Generate only some debugging info
-nowarn	Generate no warnings
-verbose	Output messages about what the compiler is doing
-deprecation	Output source locations where deprecated APIs are used
-classpath <path>	Specify where to find user class files
-cp <path>	Specify where to find user class files
-sourcepath <path>	Specify where to find input source files
-bootclasspath <path>	Override location of bootstrap class files
-extdirs <dirs>	Override location of installed extensions
-endorseddirs <dirs>	Override location of endorsed standards path
-d <directory>	Specify where to place generated class files
-encoding <encoding>	Specify character encoding used by source files
-source <release>	Provide source compatibility with specified release
-target <release>	Generate class files for specific VM version
-version	Version information
-help	Print a synopsis of standard options
-X	Print a synopsis of nonstandard options
-J<flag>	Pass <flag> directly to the runtime system

C:\Documents and Settings\ODAE EDO>

---

Si vous obtenez un autre message, c'est qu'il y a une erreur...

Maintenant, on peut travailler notre JAVA !!!!

Pour programmer notre petit programme Java, nous allons prendre le bloc-note. C'est simple et efficace...

Tapez les lignes suivante:

```
public class Javabien {
public static void main(String[] arguments) {
String line1 = "bonjour, ";
String line2 = "on s'amuse comme on peut";
String quote = line1 + line2;
System.out.println(quote);
}
}
```

Sauvegardez votre document sous le nom Javabien.java

Ouvrez l'invite de commandes windows... et tapez:

```
javac Javabien.java
```

```
javac compile notre document *.java en code executable *.class
```

Maintenant, pour exécuter notre programme qui vient d'être compilé sous la forme Javabien.class, on tape :  
java Javabien

le programme s'exécute et affiche sur votre écran la phrase suivante:

«bonjour, on s'amuse comme on peut.»

Vous voyez, c'était pas si compliqué!  
Quoi, vous avez pas compris ???  
Bon, bon, je reviens sur le code...

```
LINE 1    public class Javabien {  
LINE 2    public static void main(String[] arguments) {  
LINE 3    String line1 = "bonjour, "  
LINE 4    String line2 = "on s\'amuse comme on peut";  
LINE 5    String quote = line1 + line2;  
LINE 6    System.out.println(quote);  
LINE 7    }  
LINE 8    }
```

La ligne 1 indique qu'il faut créer un fichier de classe Java appelé Javabien

La ligne 2 indique le commencement du programme principal

La ligne 3 indique la création d'une chaîne (String) line1 qui est égale à «bonjour, »

La ligne 4 indique une autre chaîne line2 qui est égale à «on s'amuse comme on peut»

Note: l'apostrophe est noté \' afin qu'il n'y ait pas de confusion avec le double apostrophe " " qui contient la valeur de la chaîne.  
Si, j'avais tapé «"on s'amuse" au lieu de «"on s\'amuse", à l'affichage, j'aurais obtenu «on s».

La ligne 5 indique la création d'une chaîne «quote» qui est égale à la concaténation de la chaîne «line1» et de la chaîne «line2»

La ligne 6 indique au système qu'il doit afficher à l'écran la variable «quote» (c'est à dire le contenu de la chaîne quote)

La ligne 7 indique la fin du programme principale

La ligne 8 indique la fin du programme de la classe

# HTML

Je sais ce que vous allez me dire... Je me répète là! Et bien, pas vraiment! En effet, dans le premier numéro de la Gazette du DEUST TSIC, je vous ai parlé du langage HTML de manière très rapide. Je reviens donc sur ce langage, aujourd'hui, en essayant d'être le plus complet possible...

Pour ceux et celles qui l'ont déjà oublié, HTML signifie HyperText Markup Language. Il s'agit du premier langage utilisé sur le World Wide Web. HTML a évolué jusqu'à la version 4.01. Progressivement, il est remplacé par le DHTML et le XHTML. Le HTML a été normalisé par le World Wide Web Consortium, un groupe chargé de normaliser les standards du web. Pour de plus amples informations, consulter le site <http://www.w3.org/>.

Un code HTML typique est de la forme:

```
<html>
<head></head>
<body></body>
</html>
```

Pour ceux qui ne l'ont pas remarqué, il s'agit des balises de base d'une page vierge. Entre les balises «<head></head>», on retrouvera l'en-tête de la page et entre les balises «<body></body>», on retrouvera le corps du document, c'est à dire ce qui va s'afficher à l'écran.

## 1°) Déclaration du document HTML

Comme je l'ai dit précédemment, il existe différentes versions du langage HTML. Ainsi, si l'on souhaite mettre sa page en conformité avec les normes du W3C, il faut préalablement déclarer la version HTML utilisée, avant même la première balise «<html>». Pour cela, on écrit les tags suivants :

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML Level1//EN">
pour la version 1 de HTML.
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML //EN">
pour la version 2 de HTML.
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 3.0 //EN">
pour la version 3.0 de HTML.
```



`<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">`  
pour la version 3.2 de HTML.

`<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4//EN">`  
pour la version 4 de HTML.

`<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">` pour la version 4.01 de HTML.

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

Si l'on ne maîtrise pas complètement les normes du HTML, il est préférable d'utiliser le tag «Transitional» qui tolère quelques écarts par rapport au standard.

Autre petit détail, HTML ne distingue pas les majuscules des minuscules, par conséquent, on peut écrire les balises en majuscules comme en minuscules.

exemple: `<HTML>` est identique à `<html>`

Toutefois, avec l'évolution du langage, il est préférable d'écrire les balises en minuscule. En effet, XHTML, la dernière évolution du HTML, n'apprécie que les minuscules. Donc, il est bon de prendre de bonnes habitudes dès le début. Cela vous facilitera le passage à XHTML.

## 2) Structure de l'en-tête

Voici une liste non exhaustive des balises que l'on peut retrouver dans l'en-tête d'une page HTML, placées entre les balises "`<head>`" et "`</head>`".

`<title></title>` permet de définir le titre de la page HTML. Le titre défini entre ces balises s'affichera dans la barre de navigation de votre navigateur web. Si les balises "title" ne sont pas définies, ce qui s'affichera dépendra alors de votre navigateur web.

`<base>` définit l'adresse de base du document.

La balise "base" peut contenir deux propriétés "href" et "target".

"href" définit l'emplacement du document et "target" permet de définir le nom de la fenêtre ou du cadre à utiliser pour afficher les documents ouverts par les liens de la page.

La syntaxe de BASE est :

`<base href = "http://david.nogent.free.fr/la_gazette.html" target = "nomDeLaFenetre">`

LA PROPRIÉTÉ "href" DOIT CONTENIR L'ADRESSE COMPLÈTE DE LA PAGE  
"`[PROCOLE]://[SERVEUR]/[DOSSIER]/[FICHER]`".

“<meta>” est une balise fort utile. Elle déclare des paramètres décrivant certains éléments du document ou utilisables par des programmes externes (par exemple, les moteurs de recherche).

“meta” peut comporter les propriétés “name” et “http-equiv”. “name” définit le nom d’un paramètre décrivant une propriété du document. “http-equiv” définit le nom d’un champ http reconnu par le navigateur web. On utilisera “content” pour préciser la valeur donnée aux propriétés “name” et “http-equiv”.

Par exemple, cela donnera :

```
<meta name = “Author” content = “DAVID NOGENT”>
```

```
<meta name = “Description” content = “page d’accueil de la gazette du deust tsic”>
```

```
<meta name = “Keywords” content = “deust, informatique, association étudiant”>
```

```
<meta name = “Date” content = “10/04/2006”>
```

```
<meta name = “Generator” content = “macromedia dreamweaver 8”>
```

LA PROPRIÉTÉ “NAME = “GENERATOR” N’EST PAS SOUVENT UTILISÉE. TOUTEFOIS, ELLE EST GÉNÉRÉE AUTOMATIQUÉMENT PAR CERTAINS LOGICIELS WYSIWYG (*“WHAT YOU SEE, WHAT YOU GET”*). AINSI, UN WEBMASTER PROFESSIONNEL SAURA QUEL TYPE DE DÉVELOPPEUR VOUS ÊTES. S’IL VOIT CETTE PROPRIÉTÉ AVEC, COMME VALEUR, LE NOM D’UN DE CES LOGICIELS, IL SAURA QUE VOUS N’ÊTES PAS UN VRAI PRO DU DÉVELOPPEMENT WEB. ;+)

```
<meta http-equiv = “expires” content = “Sun, 15 dec 2006 11:18:45 GMT”>
```

INDIQUE LA DATE À PARTIR DE LAQUELLE LA PAGE EST CONSIDÉRÉE COMME EXPIRÉE.

```
<meta http-equiv = “refresh” content = “15; url=la_gazette.html”>
```

INDIQUE QU’IL FAUT RAFRAICHIR LA PAGE AU BOUT DE 15 SECONDES.

```
<meta http-equiv = “content-type” content = “text/html; charset=iso-8859-1”>
```

SPÉCIFIE LE FORMAT (MIME TYPE) DU DOCUMENT ENVOYÉ AU NAVIGATEUR.

On peut aussi retrouver la balise “<style></style>” dans l’en-tête. Toutefois, nous sommes ici en HTML. Par conséquent, oublions la! Nous la retrouverons plus tard.

### 3) Introduction de commentaires dans le code

En programmation, il est parfois très utile de placer des commentaires dans le code source du programme. De cette manière, il sera facile à un programmeur de reprendre votre travail. Cela est valable pour tous les langages de programmation et le HTML n’y déroge pas!

En HTML, il existe deux manières pour écrire des commentaires.

La première façon est la version officielle reconnue par le W3C:

```
<!-- ceci est un commentaire html conforme au langage -->
```

La seconde est une stupidité inventée par Microsoft Corporation:

```
<comment>ceci est un commentaire made by microsoft</comment>
```

“<COMMENT>” N’EST RECONNU QUE PAR MICROSOFT INTERNET EXPLORER ! OUBLIEZ DONC CETTE BALISE COMPLÈTEMENT INUTILE. CAR LE COMMENTAIRE S’AFFICHENT DANS LES PAGES AVEC LES AUTRES NAVIGATEURS.

#### 4) Structure du corps “BODY”

Avant de poursuivre avec la structure du corps d’une page HTML, je vais vous présenter les différentes propriétés de la balise “<body>”.

En pure HTML, il est possible d’ajouter plusieurs propriétés au corps du document. On peut, par exemple, donner une couleur au fond de la page.

```
<body bgcolor = “#00FFFF” text = “#8B0000” link = “#000000” alink = “#FF00FF”  
vlink = “#808080”>
```

“bgcolor” définit une couleur d’arrière plan de couleur bleu cyan

IL EST POSSIBLE D’UTILISER UNE IMAGE EN ARRIÈRE PLAN EN UTILISANT LA PROPRIÉTÉ  
«background = “url de l’image”»

“text” définit la couleur par défaut du texte, il est de couleur rouge foncé

“link” définit la couleur par défaut des liens hypertextes présents sur la page

“alink” définit la couleur des liens hypertextes cliqués

“vlink” définit la couleur des liens hypertextes visités

#### *CODES HEXADÉCIMAUX\* RELATIFS AUX COULEURS:*

NOIR / BLACK = #000000  
BLANC / WHITE = #FFFFFF  
BLEU / BLUE = #00FFFF  
ROUGE / RED = #FF0000  
VERT / GREEN = #008000  
JAUNE / YELLOW = #FFFF00  
ORANGE / ORANGE = #FFA500  
GRIS / GRAY = #808080  
MARON / MAROON = #800000  
ROSE / PINK = #FFC0CB  
ARGENT / SILVER = #C0C0C0

\* Un nombre hexadécimal s’écrit avec les chiffres de 0 à 9 et les lettres de A à F. En HTML, on le fait précéder par le symbole “#”

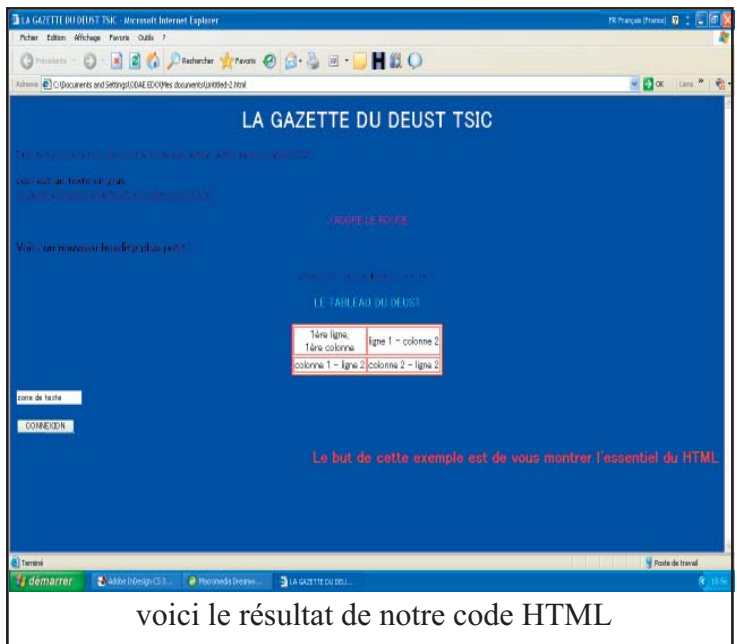
Afin de vous présenter, maintenant, la structure du corps d’un document HTML, je vais utiliser un code HTML. Cela vous permettra d’apprendre à lire un code source HTML et de le comprendre grâce aux commentaires.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<!-- déclaration de la version HTML utilisée -->
<html>
<head>
<meta name="author" content="DAVID NOGENT"><!-- définition du nom de l'auteur du document -->
<meta name="description" content="page exemple de la gazette du deust"><!-- description du document -->
<meta name="keywords" content="informatique, html"><!-- keywords pour les moteurs de recherche -->
<meta http-equiv="Content-Type" content="text/html; charset=shift_jis"><!-- format du document -->
<title>LA GAZETTE DU DEUST TSIC</title><!-- titre du document -->
</head>

<body bgcolor="#0000FF" text="#000000" link="#FF0000" alink="#990033" vlink="#333333">
<!-- définition de la balise body
    la couleur du fond est bleu
    la couleur par défaut du text est noir
    la couleur des liens hypertextes est rouge
    la couleur des liens cliqués est "franboise"
    la couleur des liens visités est gris foncé
-->
<h1 align="center"><font color="#FFFFFF">LA GAZETTE DU DEUST TSIC</font></h1><!-- fermeture des balises-->
<!-- définition d'un titre (heading) avec la balise h1 dont l'alignement est centré
    on précise avec la balise font color la couleur que l'on souhaite donner au texte
    ici, on choisit la couleur blanche -->
Mon texte ici sera noir car c'est le texte par défaut défini dans la balise BODY
<br><br><!-- aller retour chariot // saut de ligne -->
<b>ceci est un texte en gras</b><!-- texte gras // b comme bold -->
<br><!-- passage à la ligne suivante -->
<u>je passe &agrave; la ligne suivante et je souligne mon texte</u><!-- texte souligné // u comme underline -->
<!-- le caractère accentué 'à' s'écrit &agrave; en HTML -->
<p align="center"><!-- introduction d'un paragraphe que j'aligne au centre -->
<font color="#FF0000"><!-- Je définis une nouvelle couleur pour mon texte-->
J'ADORE LE ROUGE</font><!-- fermeture de la balise-->
</p><!-- fermeture de la balise p -->
<h3>Voici un nouveau heading plus petit !</h3><!-- les headings vont de h1 à h7 -->
<!-- la propriété align n'est pas défini par conséquent l'alignement est celui par défaut -->
<center><!-- je vais centrer un texte -->
<tt>Aimez-vous les machines &agrave; &ecirc;rire ?</tt>
<!-- avec cette balise mon texte semble avoir été tapé sur une marchine à écrire -->
</center><!-- fermeture de la balise center // retour à l'alignement par défaut -->
<h4 align="center"><font color="#00FFFF">LE TABLEAU DU DEUST</font></h4>
<!-- voici un autre heading avec un autre couleur de texte et l'alignement est centré -->
<table align="center" bgcolor="#FFFFFF" border="2" bordercolor="#FF0000">
<!-- j'introduis un tableau a l'aide de la balise TABLE dans laquelle
    je définis un alignement centré
    la couleur du font de mon tableau sera blanc
    j'y mets une bordure d'épaisseur 2 (la valeur 0 signifit PAS DE BORDURE)
    cette bordure est de couleur rouge -->
<tr><!-- introduction de la première ligne du tableau -->
    <td align="center"><!-- la première colonne de mon tableau avec alignement centré-->
    l&egrave;re ligne,<br><!-- passage à la ligne suivante -->
    l&egrave;re colonne
    <!-- le caractère 'è' s'écrit &egrave; en HTML -->
    </td>
    <td><!-- deuxième colonne -->
    ligne 1 - colonne 2
    </td>
</tr><!-- fin de la première ligne -->
<tr><!-- ligne numéro 2 -->
    <td><!-- colonne 1 -->
    colonne 1 - ligne 2
    </td>
    <td><!-- colonne 2 -->
    colonne 2 - ligne 2
    </td>
</tr><!-- fin de la deuxième ligne -->
</table><!-- fin du tableau -->
<form method="post" action="http://www.dep.u-picardie.fr/">
<!-- déclaration d'un formulaire
    on utilise la methode POST, mais on peut aussi utilisé la methode GET
    l'action demandé sera de se rendre sur la plateforme de la D.E.P.
-->
<input type="text" value="zone de texte">
<!-- déclaration d'une zone de texte -->
<br><br><!-- saut de ligne -->
<input type="submit" value="CONNEXION">
<!-- déclaration d'un bouton de type submit
    en cliquant sur ce bouton, la propriété ACTION s'exécutera -->
</form><!-- fin du formulaire -->
<h2 align="right"><!-- un nouvel heading avec alignement à droite -->
<font color="#FF0000"><!-- définition d'un couleur rouge pour le texte -->
Le but de cette exemple est de vous montrer l'essentiel du HTML
</font>
</h2>
</body>
</html>

```



# DHTML / XHTML

DHTML est une amélioration du code HTML auquel on associe des codes CSS (Cascading Style Sheets) et/ou Javascript . DHTML signifie Dynamic HTML.

Pour pouvoir faire du DHTML, on utilise la balise "<style>" dans l'en-tête du document. Je vous avez dit qu'on allait la retrouver et bien la voici!

Voici la syntaxe de la balise :

```
<style type="text/css">
<!-- entre les deux balises,
      on insere du code CSS
      que nous verrons ultérieurement -->
</style>

<style type="text/javascript">
<!-- entre ces deux là, on insere
      un code Javascript -->
</style>
```

Il existe un autre moyen. En effet, on peut disposer de fichiers \*.css et \*.js (javascript) à l'extérieur d'une page HTML. Dans ce cas là, la syntaxe sera :

```
<link rel="stylesheet" type="text/css" href="styles.css" media="screen">
<!-- utilisation de la balise link pour utiliser un fichier *.css -->

<script language="javascript" src="http://david.nogent.free.fr/javascript.js">
</script>
<!-- utilisation de la balise <script> pour utiliser un fichier *.js -->
```

Dans les pages DHTML, dans le cadre de l'utilisation de javascript, on peut utiliser cette version de la balise "<script>":

```
<script language="javascript">
<!-- on insere ici du code javascript -->
</script>
```

En dehors de ces balises, DHTML utilise les mêmes balises que le simple HTML.

Le XHTML est un peu différent. La syntaxe du XHTML qui signifie Extended HTML (HTML Etendu) est plus rigoureuse que celle du HTML. En effet, il est obligatoire d'écrire toutes les balises correctement en minuscules et elles doivent toutes être fermées. XHTML repose sur le XML qui n'était pas compatible avec le HTML.



La syntaxe du XHTML donnera :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta name="author" content="DAVID NOGENT" />
<title>DEUST TSIC</title>
</head>
<body>
<h1>Ceci est mon titre</h1>
Il faut bien que jeunesse se fasse<br />
n'écirc;tes vous donc pas d'accord ?
</body>
</html>
```

Comme vous pouvez vous en rendre compte, les balises `<meta>` et `<br>` ont changé. Comme il n'existe pas de balises fermantes pour ces balises, leurs syntaxes a changé en XHTML pour devenir `<meta/>` et `<br/>`.

Maintenant, voyons d'autres balises HTML qu'il est utile de connaître.

"`<i></i>`" permet de mettre le texte situé entre ces balises en italique.

Il en va de même avec les balises "`<em></em>`".

"`<strong></strong>`" permet d'obtenir des caractères en gras comme "`<b></b>`".

Les listes à puces :

```
<ul type = "circle">
<li>HJHX</li>
<li>DEFF</li>
</ul>
```

UL signifie Unordored List (liste non ordonnée)

elle peut être de type "circle" (cercle), "disc" (puce noire), "square" (carré)

LI contient les éléments de la liste.

```
<ol type="1" start="4">
<li>DEUST TSIC</li>
<li>LICENCE MOCI</li>
</ol>
```

A l'affichage cela donne:

4 DEUST TSIC

5 LICENCE MOCI

OL signifie Ordored List (liste ordonnée)

elle peut être de type "A", "a", "1", "1"

## Le TYPE

"A" déclare une séquence en lettres majuscules

"a" déclare une séquence en lettres minuscules

"l" déclare une séquence en chiffres romains

"l" déclare une séquence numérique

La propriété START indique la valeur de départ de la liste numérotée.

<hr> ou <hr /> permet de créer une ligne horizontale sur toute la longueur de la fenêtre du navigateur afin de délimiter une portion de document.

<strike></strike> permet de barrer du texte

<sub></sub> définit du texte en indice

<sup></sup> définit du texte en exposant

<img src = "upjv.png" height = "50" width = "200" alt = "logo de mon université;">

<img src = "upjv.png" height = "50" width = "200" alt = "logo de mon université;" />

La balise IMG permet d'afficher une image sur une page HTML/XHTML.

La propriété SRC définit l'adresse relative<sup>0</sup> ou absolue\* de l'image.

La propriété HEIGHT définit la hauteur de l'image. Ici 50 pixels.

WIDTH définit la largeur de l'image. Ici, elle est de 200 pixels. On aurait aussi pu redéfinir la hauteur et la largeur de l'image en pourcentage.

ALT décrit le document image. De plus, si l'image refuse de s'afficher, le contenu de ALT s'affiche à la place de l'image.

<sup>0</sup> une adresse relative est de la forme src = "upjv.png", cela signifie que l'image se trouve dans le même dossier que ma page HTML .

\* une adresse absolue est de la forme src = "http://david.nogent.free.fr/deust\_tsic/upjv.png" .

## Listes de définitions :

<dl> <!-- crée une liste de définition -->

<dt> UPJV <!-- terme à définir -->

<dd> Université de Picardie Jules Vernes <!-- définition du terme -->

</dl>

## Quelques caractères spéciaux :

&acute; pour " é ";

&grave; pour " è ";

&agrave; pour " à ";

&copy; pour " © ";

&reg; pour " ® ";

&nbsp; pour " " (espace vide);

&ecirc; pour " ê ";

&acirc; pour " â ";

&ccedil; pour " ç ";

&iexcl; pour le point d'exclamation inversé;

&iquest; pour le point d'interrogation inversé;

&icirc; pour " î ";

&iuml; pour " ï ";

&igrave; pour " ù ";

&amp; pour " & ";

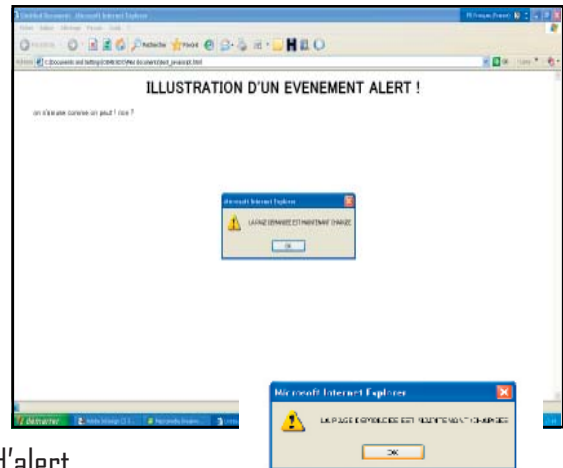
&euml; pour " è ";

# JAVASCRIPT

Loin de moi l'idée de vous réécrire un cours sur Javascript. Premièrement, je n'en suis pas capable. Deuxièmement, Javascript, qui bien que reposant un peu sur le langage Java, nécessite un esprit logique. J'en suis, hélas, dépourvu. ;+) Le mieux est donc d'étudier ensemble des exemples de code.

Générer un évènement:

```
<html>
<head>
<title> Generer un evenement avec Javascript</title>
<script language = "javascript"> <!-- script javascript -->
function PageChargee() // ceci est une fonction javascript
// en javascript, les commentaires débutent par "//"
{
window.alert ('LA PAGE DEMANDEE EST MAINTENANT CHARGEE');
/* on peut aussi écrire en java un commentaire complet en
un bloc de cette maniere !
La fonction PageChargee demande l'execution d'un message d'alert
qui affichera le message 'LA PAGE DEMANDEE EST MAINTENANT CHARGEE'
lorsque la page sera affichee dans le navigateur */
}
</script>
</head>
```



```
<body onload="PageChargee()"><!-- on demande l'execution de la fonction lors du chargement de la page -->
<h1 align="center">ILLUSTRATION D'UN EVENEMENT ALERT !</h1>
<blockquote><!-- balise de tabulation -->
on s'amuse comme on peut ! non ?
</blockquote>
</body>
</html>
```

Le message qui s'affiche dans mon exemple n'est certe pas très original, mais je pense que vous imaginez déjà les possibilités d'un tel script...

Certains navigateurs un peu anciens ne reconnaissent pas le langage Javascript. Du coup cela provoque un affichage incohérent! Voici, une astuce pour ne pas avoir de mauvaise surprise:

```
<script language = "javascript"><!--
function PageChargee()
{
window.alert ('LA PAGE DEMANDEE EST MAINTENANT CHARGEE');
} //-->
</script> Ainsi, votre script passera pour un commentaire HTML sur les anciens navigateurs!
```



```

if (document.myQCM.rep20.checked)
{document.myQCM.rep20.value=c;}
else {document.myQCM.rep20.value=b;}
if (document.myQCM.rep22.checked)
{document.myQCM.rep22.value=a;}
else {document.myQCM.rep22.value=b;}
if (document.myQCM.rep25.checked)
{document.myQCM.rep25.value=c;}
else {document.myQCM.rep25.value=b;}
if (document.myQCM.rep26.checked)
{document.myQCM.rep26.value=c;}
else {document.myQCM.rep26.value=b;}
if (document.myQCM.rep28.checked)
{document.myQCM.rep28.value=c;}
else {document.myQCM.rep28.value=b;}
if (document.myQCM.rep30.checked)
{document.myQCM.rep30.value=c;}
else {document.myQCM.rep30.value=b;}
{
A=document.myQCM.rep03.value;
B=document.myQCM.rep04.value;
C=document.myQCM.rep05.value;
D=document.myQCM.rep07.value;
E=document.myQCM.rep12.value;
F=document.myQCM.rep14.value;
G=document.myQCM.rep16.value;
H=document.myQCM.rep19.value;
I=document.myQCM.rep20.value;
J=document.myQCM.rep22.value;
K=document.myQCM.rep25.value;
L=document.myQCM.rep26.value;
M=document.myQCM.rep28.value;
N=document.myQCM.rep30.value; // utilisation de nouvelles variables pour simplifier les calculs...
}
{
alert("YOU HAVE : " + (A-(-B)-(-C)-(-D)-(-E)-(-F)-(-G)-(-H)-(-I)-(-J)-(-K)-(-L)-(-M)-(-N)) + "/20")
}
// affichage du calcul des points dans une fenetre
}
// fin du script que l'on sauvegarde sous le nom "qcm_note.js"

```

2/ maintenant voici le code HTML qui va utiliser ce script

```

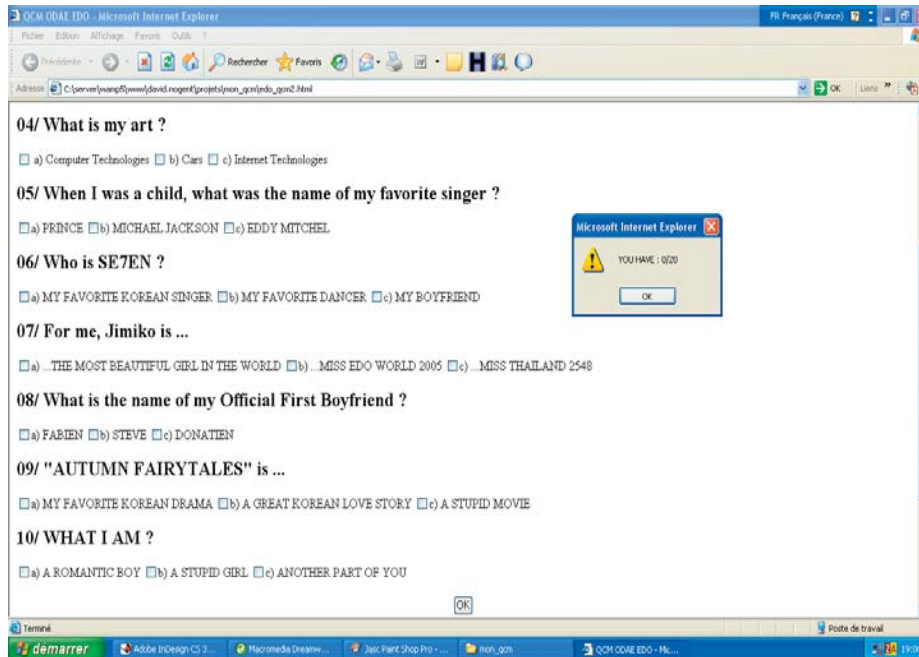
<html>
<head>
<title>QCM ODAE EDO</title>
<script language="javascript" src="qcm_note.js"></script> <!-- lien vers le fichier javascript -->
</head>
<body>
<center>
<p><h1><b><u> DO YOU REALLY KNOW ME ? </u></b></h1></p>
</center>
<form name="myQCM"><!-- ceci est un formulaire comprenant des cases à cocher -->
<h2>01/ What is my name ?</h2>
<input type="checkbox" name="rep01">a) DEBIDO NOGENTSU
<input type="checkbox" name="rep02">b) DAVID ALAIN NOGENT
<input type="checkbox" name="rep03">c) ODAE EDO
<br><br>
<h2>02/ What is my favorite color ?</h2>
<input type="checkbox" name="rep04">a) PINK
<input type="checkbox" name="rep05">b) BLACK
<input type="checkbox" name="rep06">c) BLUE

```



```
<br><br>
<h2>03/ What is the name of my secret love ?</h2>
<input type="checkbox" name="rep07">a) JIMIKO
<input type="checkbox" name="rep08">b) DAYSIREE
<input type="checkbox" name="rep09">c) HARISU
<br><br>
<h2>04/ What is my art ?</h2>
<input type="checkbox" name="rep10">
a) Computer Technologies
<input type="checkbox" name="rep11">
b) Cars
<input type="checkbox" name="rep12">
c) Internet Technologies
<br>
<br>
<h2>05/ When I was a child, what was the name of my favorite singer ?</h2>
<input type="checkbox" name="rep13">a) PRINCE
<input type="checkbox" name="rep14">b) MICHAEL JACKSON
<input type="checkbox" name="rep15">c) EDDY MITCHEL
<br><br>
<h2>06/ Who is SE7EN ?</h2>
<input type="checkbox" name="rep16">a) MY FAVORITE KOREAN SINGER
<input type="checkbox" name="rep17">b) MY FAVORITE DANCER
<input type="checkbox" name="rep18">c) MY BOYFRIEND
<br><br>
<h2>07/ For me, Jimiko is ...</h2>
<input type="checkbox" name="rep19">a) ...THE MOST BEAUTIFUL GIRL IN THE WORLD
<input type="checkbox" name="rep20">b) ...MISS EDO WORLD 2005
<input type="checkbox" name="rep21">c) ...MISS THAILAND 2548
<br><br>
<h2>08/ What is the name of my Official First Boyfriend ?</h2>
<input type="checkbox" name="rep22">a) FABIEN
<input type="checkbox" name="rep23">b) STEVE
<input type="checkbox" name="rep24">c) DONATIEN
<br><br>
<h2>09/ "AUTUMN FAIRYTALES" is ...</h2>
<input type="checkbox" name="rep25">a) MY FAVORITE KOREAN DRAMA
<input type="checkbox" name="rep26">b) A GREAT KOREAN LOVE STORY
<input type="checkbox" name="rep27">c) A STUPID MOVIE
<br><br>
<h2>10/ WHAT I AM ?</h2>
<input type="checkbox" name="rep28">a) A ROMANTIC BOY
<input type="checkbox" name="rep29">b) A STUPID GIRL
<input type="checkbox" name="rep30">c) ANOTHER PART OF YOU
<br><br>
<center><input type="button" name="OK" value="OK" onClick="Note()">
<!-- en cliquant sur le bouton OK, exécution de la fonction Note() du fichier "qcm_note.js" -->
</center></form><!-- fin du formulaire principal -->
<br><br>
<h5><i>If you have 20/20, you are really my best friends<br>
If you have between 15/20 and 20/20, you are a good friend<br>
If you have between 10/20 and 15/20, you know me<br>
If you have between 0/20 and 10/20, <b>ARE YOU REALLY MY FRIENDS ?</b><br>
PLEASE, SEND ME YOUR SCORE AT <a href="mailto:edo80800@msn.com"><font color="#FF0000">edo80800@msn.
com</font></a><br>
THANK YOU SO MUCH TO PLAY WITH ME...</b></i></h5>
<center>
<form method="get" action=" ../principle_frame.html"> <!-- un mini formulaire -->
<input type="submit" value="RETURN"> <!-- ce bouton de type submit charge une page html -->
</form>
</center>
<center>Copyright &copy; 2005 EdoWorldConcept. All Rights Reserved.</center></body>
</html>
```

A l'affichage, cela donnera :



*si le résultat de cette affichage, ne vous convient pas,  
revoyez mon code ligne par ligne et essayer de l'exécuter...*

Et maintenant, amusons-nous avec des images... pour faire un diaporama avec javascript... ;+)

1b/ voici mon script "diaporama.js" :

```
// diaporama en javascript
function permuter(nouvelleImage)
{
var nomDuFichier = nouvelleImage.toString() + ".gif"
document.all.diaporama.src = nomDuFichier
}
/* la fonction permuter() construit un nom de fichier
d'après l'argument qui lui est passé puis utilise
ce nom pour afficher le fichier correspondant

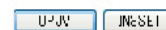
Note: c'est l'image initiale "upjv1.gif" qui détermine
la taille du cadre du diaporama
*/
```

2b/ mon code HTML "UPJV.html" :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=shift_jis" />
<title>DIAPORAMA EN JAVASCRIPT</title>
<script language="javascript" src="diaporama.js"></script>
</head>
<body>
<h1 align="center">DIAPORAMA EN JAVASCRIPT</h1>
<center>

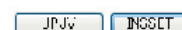
<br />
<input type="button" value=" UPJV " onclick="permuter('upjv1')"/>
<input type="button" value=" INSET " onclick="permuter('upjv2')"/>
</center>
</body>
</html>
```

## DIAPORAMA EN JAVASCRIPT



↑  
image 1

## DIAPORAMA EN JAVASCRIPT



↑  
image 2

# CSS - Cascading Style Sheets - Les Feuilles de Style

Les feuilles de style CSS sont un format standard du W3C permettant d'enrichir et de contrôler finement la présentation des pages web. Malgré les recommandations du W3C, elles sont peu utilisées. Ce qui est d'autant plus regrettable, dans la mesure où elles permettent un allègement du code HTML, PHP ou JSP.

Une feuille de style est un document "\*.css" qui prend la forme suivante :

```
/* Feuille de style CSS */
body { /* definition des proprietes du corps du document */
background-color:#FFCC00; /* fond de couleur Or */
border:#000000; /* definition d'une bordure de couleur noire */
border-top:double; /* definition de la bordure du haut de la page de type "double" */
border-top-width:20px; /* definition de l'epaisseur de la bordure du haut 20 pixels */
border-left:solid; /* definition de la bordure gauche de la page de type "solide" */
border-left-width:20px; /* definition de l'epaisseur de la bordure gauche 20 pixels */
border-right:solid; /* definition de la bordure droite de la page de type "solide" */
border-right-width:20px; /* definition de l'epaisseur de la bordure droite 20 pixels */
border-bottom:double; /* definition de la bordure du bas de la page de type "double" */
border-bottom-width:20px; /* definition de l'epaisseur de la bordure gauche 20 pixels */
font:"Courier New", Courier, monospace; /* definition d'une police par default */
font-family:"Courier New", Courier, monospace; /* famille de la police par default */
font-size:16px; /* taille de la police par default */
font-style:normal; /* style de police par default */
margin-left:10px; /* definition d'une marge gauche de 10 pixels */
margin-right:10px; /* definition d'une marge droite de 10 pixels */
height:540px; /* hauteur de la page web */
width:1200px; /* largeur de la page web */
}

h1 { /* definition des proprietes du TITRE H1 */
background-color:#999999; /* definition d'un arriere plan nuance de gris au TITRE H1 */
font:Arial, Helvetica, sans-serif; /* definition d'une police */
font-family:Arial, Helvetica, sans-serif;
font-size:36px; /* taille de la police H1 */
color:#FFFFFF; /* couleur du texte H1 */
text-align:center; /* alignement du TITRE H1 */
}

h3 { /* definition des proprietes du TITRE H1 */
text-align:center; /* alignement au centre */
}

table { /* definition des proprietes du tableau */
background-color:#6600CC; /* arriere plan de couleur violet */
border:#000000; /* bordure noire */
border:dotted; /* bordure en forme de pointilles */
text-align:center; /* alignement du texte au centre */
color:#FFFFFF; /* definition des proprietes du TITRE H1 */
font-size:14px; /* taille de la police */
}

td { /* definition des proprietes de la cellule */
border:#000000; /* bordure noire */
border:dotted; /* bordure en forme de pointilles */
}
/* Il existe de nombreuses autres possibilites... La seule limite reste notre imagination... */
```

Pour s'y retrouver, on va sauvegarder notre feuille de style sous le nom : "gold\_style.css".



Avant d'approfondir sur les feuilles de style CSS, je vais faire quelques commentaires sur le code de notre page HTML...

Je ne reviens pas sur la balise "<link>" qui a été vue dans la partie "DHTML/XHTML".

"<img>" permet, comme nous l'avons déjà vu, l'insertion d'une image dans un document HTML. Dans notre code HTML, nous avons ajouté quelques propriétés à la balise.

"align" permet de définir un alignement.

"vspace" définit un espace vertical de 20 pixels autour de l'image.

"hspace" définit un espace horizontal de 20 pixels autour de l'image

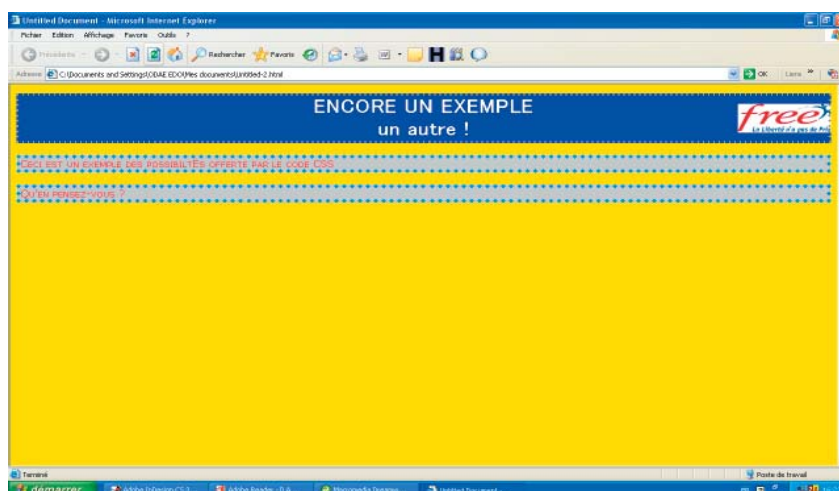
Grâce à l'utilisation de "hspace" et "vspace", on a la possibilité d'insérer notre image dans notre texte sans avoir à recourir à la création d'un tableau. Ceci dit, certains professionnels utilisent encore les tableaux pour effectuer leur mise en page. Mais, chacun programme comme il l'entend.

La balise "<p>" définit un paragraphe. Dans notre feuille CSS, nous ne lui avons donné aucune propriété. Cela a pour conséquence de lui attribuer les valeurs par défaut définies dans "body".

Voici d'autres possibilités:

```
body{
background-color:#FFFF00; <!-- arrière plan de la page -->
}
h1 {
border:dotted #999999; <!-- définit une propriété générale pour les bordures -->
text-align:center; <!-- alignement du texte "center", "right", "left" ou "justify" -->
color:#FFFFFF; <!-- couleur de la police -->
background-color:#0000FF; <!-- arrière plan pour l'élément H1, on aurait pu mettre une image-->
background-image:url(Mes images/logo_free.gif); <!-- voici donc l'image -->
background-position:right; <!-- elle est placée à droite -->
background-repeat:no-repeat <!-- mais elle ne se répètera pas -->
}
p{
border: thick dotted #0099FF;
background-color:#CCCCCC;
font:Verdana, Arial, Helvetica, sans-serif;
font-family:Verdana, Arial, Helvetica, sans-serif;
font-variant:small-caps; <!-- les minuscules apparaissent en majuscules ;+) -->
color:#FF0000;
text-align:left;
}
```

Si vous appliquez ces propriétés à une page HTML, vous pourrez obtenir un résultat comme celui-ci! Bien entendu, il y a encore plein de possibilité... comme je vous l'ai déjà dit, la seule limite c'est notre imagination... Pour en savoir plus sur les feuilles de style, consultez le site européen "<http://openweb.eu.org/css/>" en français...





Voici un autre exemple...

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=shift_jis" />
<title>Ma Page Thai !!!</title>
<style type="text/css">
body {
background-image:url(Mes images/dsd-thai-logo2.jpg); /* image en arrière-plan */
background-repeat:no-repeat; /* elle n'apparaît qu'une seule fois ! */
background-position:center; /* elle est positionné au centre de la page */
margin:30px; /* définition d'une marge de 30 px pour chaque côté de la page */
}

h1 {
font:Tahoma; /* définition d'une police de texte */
font-family:Tahoma; /* définition d'une police de texte */
font-size:72px; /* taille du texte */
text-align:center; /* alignement */
color:#3399FF; /* couleur du texte */
background-color:#000000; /* couleur de fond */
margin:20px; /* définition d'une marge de 20 px pour chaque côté de l'élément H1 */
border:dashed #FF0000; /* bordure de type DASHED rouge */
}

p {
border:double #000000; /* bordure de type DOUBLE noire */
font:Verdana, Arial, Helvetica, sans-serif; /* définition d'une police de texte */
font-family:Verdana, Arial, Helvetica, sans-serif; /* définition d'une police de texte */
font-size:18px; /* taille du texte */
text-align:justify; /* alignement du texte */
margin:10px; /* définition d'une marge de 20 px pour chaque côté de l'élément P */
height:200px; /* hauteur de P */
width:600px; /* largeur de P */
}

h2 {
font:Geneva, Arial, Helvetica, sans-serif; /* définition d'une police de texte */
font-family:Geneva, Arial, Helvetica, sans-serif; /* définition d'une police de texte */
font-size:18px; /* taille du texte */
color:#000000; /* couleur du texte */
}

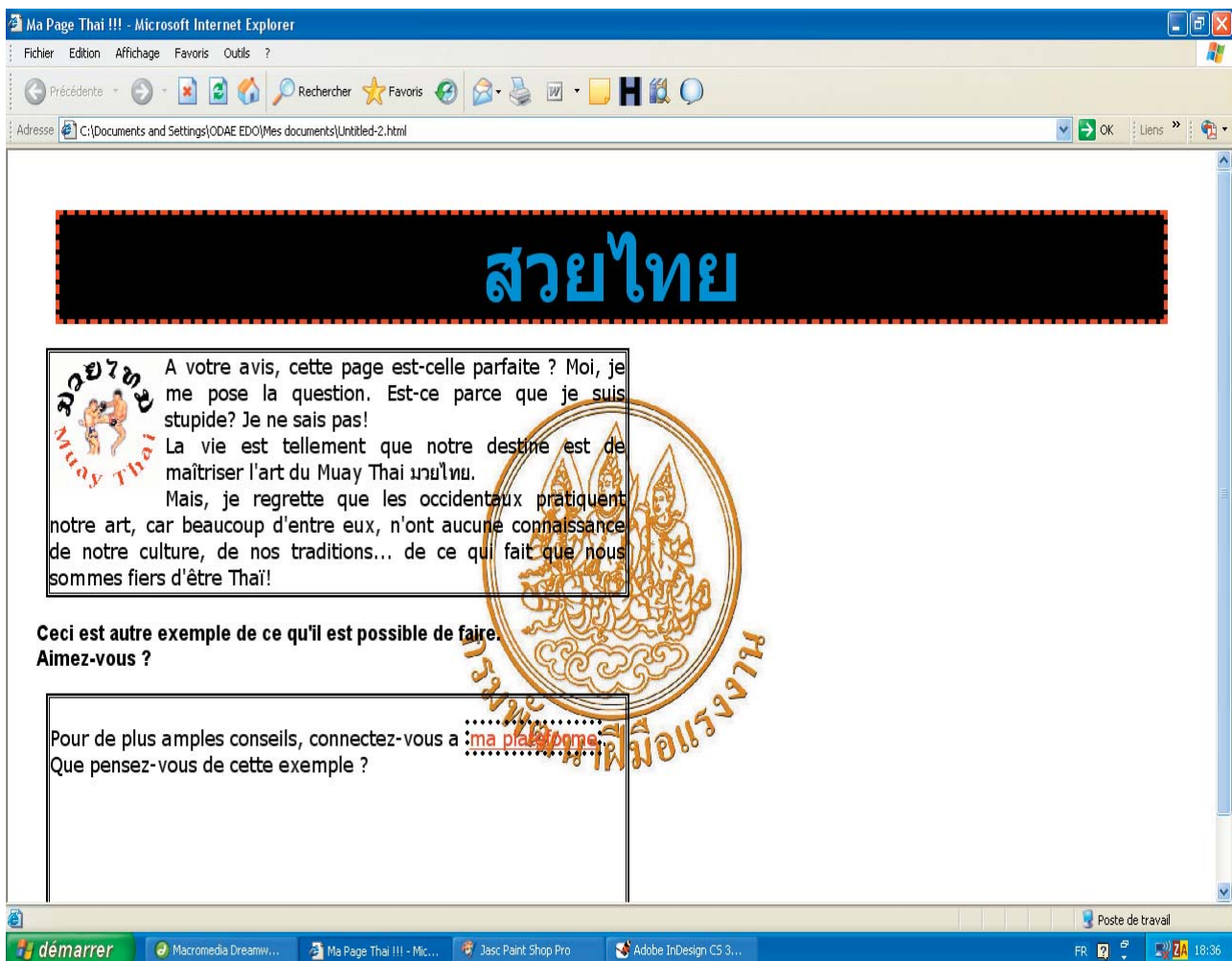
a {
border:dotted #000000; /* bordure de type DOTTED noire */
color:#FF0000; /* couleur du texte */
}
</style>
</head>
```

```

<body>
<h1>&#3626;&#3623;&#3618;&#3652;&#3607;&#3618;</h1>
<!-- BEAUTIFUL THAILAND in Thai -->
<p>A votre avis, cette page est-elle parfaite ? Moi, je me pose la question. Est-ce parce que je suis stupide? Je ne sais pas!<br/>
La vie est telle que notre destin  est de ma triser l'art du Muay Thai &#3617;&#3623;&#3618;&#3652;&#3607;&#3618;<br/>
Mais, je regrette que les occidentaux pratiquent notre art, car beaucoup d'entre eux, n'ont aucune connaissance de notre culture, de nos traditions... de ce qui fait que nous sommes fiers d' tre Tha !</p>
<h2>Ceci est autre exemple de ce qu'il est possible de faire.<br/>
Aimez-vous ?</h2>
<p><br/>Pour de plus amples conseils, connectez-vous   <a href="http://david.nogent.free.fr/">ma plateforme</a>.<br/>
Que pensez-vous de cette exemple ?</p>
</body>
</html>

```

Dans ce dernier exemple, notre code CSS est int gr    la page HTML... Le r sultat est le suivant:



Bien entendu, ce dernier exemple peut  tre am lior . On pourrait, dans notre code HTML, ajouter les balises “<center>” aux balises “<p>” et “<h2>”, voir en d but de code...

# COMPARAISON ENTRE JAVA ET SON ANCÊTRE C

Les étudiants du DEUST TSIC de l'INSSET de Saint-Quentin doivent étudier le langage C dans leur béaba de la programmation. Malgré que nous appartenons à la même section, l'étude du langage C ne fait pas partie de notre cursus. Ainsi, je pense qu'une comparaison entre le langage Java et le langage C vous permettra de connaître un tout petit peu C.

Pour cette étude, nous allons prendre un programme écrit en C puis en Java...

Voici notre programme "PERSONNE" en langage C :

```
LINE 1    #include <stdio.h>
LINE 2    #include <string.h>
LINE 3    struct Personne {
LINE 4    char Nom [25];
LINE 5    char Societe [32];
LINE 6    };
LINE 7    void Presente(struct Personne P)
LINE 8    {
LINE 9    printf ("Je m'appelle %s\n", P.Nom);
LINE 10   printf ("Je travaille à %\n", P.Societe);
LINE 11   }
LINE 12   void main()
LINE 13   {
LINE 14   struct Personne Individu;
LINE 15   strcpy (Individu.Nom, "Durand");
LINE 16   strcpy (Individu.Societe, "Outils JAVA S.A.");
LINE 17   Presente (Individu);
LINE 18   }
```

et maintenant le même programme en JAVA :

```
LINE 1    class Personne {
LINE 2        public String nom;
LINE 3        public String societe;
LINE 4        public void presenteToi(){
LINE 5            System.out.println("Je m'appelle " + nom);
LINE 6            System.out.println("Je travaille à " + societe);
LINE 7        }
LINE 8    }
```

*Pour l'exécuter, enregistrez-le en "Personne.java".*

## COMMENTAIRE SUR LE PROGRAMME EN C:

“LINE 1” à “LINE 6”, définition des paramètres.

“LINE 7”, il s’agit de la fonction Presente.

Le but de ce programme est défini dans “LINE 9” et “LINE 10”.

L’instruction “PRINTF” demande l’affichage de chaînes de caractères.

“LINE 12”, il s’agit de la fonction principale.

Elle définit le contenu des variables “Nom” et “Societe”.

## COMMENTAIRE SUR LE PROGRAMME EN JAVA :

“LINE 1”, définition de la classe Personne.

“LINE 2” et “LINE 3”, initialisation des variables “nom” et “societe”.

“LINE 4”, fonction principale “presenteToi”.

La fonction principale demande l’affichage de chaînes de caractères définie à “LINE 5” et “LINE 6”.

Cependant, il y a une petite erreur dans le programme en JAVA, en effet, contrairement à son homologue en C, les variables “nom” et “societe” ne sont pas définies.

Par conséquent, si l’on essaye d’exécuter le programme en JAVA.

Il va y avoir une erreur durant l’exécution...

En effet, comment afficher deux inconnues.

Le programme ne sait pas le faire de manière autonome.

A notre niveau, il faut donc définir les variables “nom” et “societe”.

Un programmeur plus avancé que nous créerait un autre programme qui demanderait la saisie d’un nom et d’une société et ferait ensuite appelle à ce programme pour afficher le résultat.

Pour avoir un programme parfaitement exécutable, il faut réécrire le code comme ceci (enfin, ce n’est qu’un exemple) :

```
public class Personne {
public static void main(String[] arguments) {
String nom = “Durand”;
String societe = “Outils JAVA S.A.”;
String line1 = “Je m’appelle “;
String line2 = “Je travaille a “;
String lineA = line1 + nom;
String lineB = line2 + societe;
System.out.println(lineA);
System.out.println(lineB);
}
}
```

### Information de dernière minute:

L’installation de Visual Studio 2005 remplace le jdk de Sun par le J# de Microsoft. Par conséquent, lors de l’utilisation de la commande “java”, le message suivant apparaît :

*Exception in thread “main” java.lang.NoClassDefFoundError: Javabien*

# PHP™

Ca y est le moment fatidique arrive. Il me faut vous parler de PHP. Pas facile, je n'aime pas PHP. J'y suis même allergique. Et ne me demandez pas le pourquoi du comment, je n'en sais absolument rien... Est-ce parce que mon mentor est un spécialiste Java ou est-ce parce que je veux tout simplement faire de la résistance face à l'engouement que provoque PHP ? Quoiqu'il en soit, il faut que l'on s'y mette...

PHP signifie Personal Home Page. Bien que dans certains livre sur PHP, on trouve la signification Hypertext PreProcessor.

Un code PHP de base ressemble à ceci:

```
<html>
<head>
<title>Mon Premier Script PHP</title>
</head>
<body>
<?php
$phrase = "PHP, c'est mon dada..."; // ceci est une variable PHP
$date = date("Y/m/d"); // ceci est une variable définissant la date du jour
echo "<h1>Mon Premier Script en PHP</h1>"; // echo = affichage à l'écran
echo "<br /><br />";
echo "Salut tout le monde !<br />";
echo $phrase, "\n"; // affichage de la variable $phrase. \n = retour à la ligne
echo "<h3>C'est encore un test &agrave; la dada</h3>";
echo "Aujourd'hui, nous sommes : $date.<br />";
echo "PHP c'est ainsi !<br />";
$compteur=0; // initialisation du compteur à 0
$compteur++; // incrémentation du compteur
echo "<h1>$compteur</h1>";
if ($compteur=1) // si ... faire
{
$compteur--; // décrémentation du compteur
echo $compteur, "<br /><br />";
}
for($i=0;$i<=3;$i++)
// une boucle for (instruction initiale, instruction conditionnelle, instruction finale)
{
echo "$i. PHP5 !<br />";
}
?>
</body>
</html>
```

En PHP, les commentaires sont créés de la même manière que Javascript et Java (héritage du langage C)

En PHP, on peut faire plein de choses. Toutefois, PHP sans formulaire et sans base de données, c'est comme conduire une Ferrari avec un moteur de RS. Il y a quelque chose qui ne va pas, on est limité. Il va donc nous falloir créer un formulaire simple afin de vous démontrer l'intérêt de PHP par rapport à HTML...

Mais avant cela, quelques précisions :

Contrairement à HTML, PHP ne s'exécute pas côté client mais côté serveur. Il vous faudra donc installer et configurer un serveur compatible PHP sur votre machine si vous souhaitez programmer en PHP. De ce fait, le code source est caché de l'utilisateur final. Soit dit en passant, c'est la même chose pour Java. Pour programmer en PHP, vous pouvez utiliser EasyPHP ou WAMP5 qui installeront à la fois Apache 2, PHP5 et MySQL5. Actuellement, la version courante d'EasyPHP est la version 1.8 et ne comprend que PHP4 et non PHP5.



WAMP 5 semble donc être une meilleure solution pour ceux qui veulent rester à jour. Vous trouverez dans le cours de Monsieur Mohamed SIDIR tout ce dont vous avez besoin de savoir pour installer EasyPHP et/ou configurer un serveur Windows IIS (partie 2 «La Programmation Web avec PHP» de l'Option 1 du DEUST TSIC - <http://www.dep.u-picardie.fr/>). Vous trouverez également en complément sur la plateforme de la Gazette une documentation pas à pas sur l'installation de WAMP5 ([http://david.nogent.free.fr/la\\_gazette.html](http://david.nogent.free.fr/la_gazette.html)). Autre petit détail, sous les ordinateurs japonais comme les Sony VAIO, «\n» est remplacé par «¥n».

Maintenant, passons à la conception de notre petit formulaire... ;+)

```
<html>
<head>
<title>Mon Formulaire</title>
</head>
<body>
<h1>Mon Formulaire</h1> <!-- mon titre -->
<form action="mon_php.php" method="post"> <!-- mon formulaire -->
<fieldset> <!-- encadrement de mon formulaire -->
<legend>Formulaire de connexion</legend><br/> <!-- titre de mon formulaire pour l'encadrement -->
<table align="center"> <!-- mise en page de mon formulaire dans un tableau -->
<tr><td width="300"><h3>Tapez ici votre nom : </h3></td><td><input type="text" name="nom" /></td></tr>
<tr><td width="300"><h3>Tapez ici votre mot de passe : </h3></td><td><input type="text" name="password" /></td></tr>
<tr><td width="300">&nbsp;</td><td><input type="submit" value="CONNEXION" /></td></tr>
</table> <!-- fin de mon tableau -->
</fieldset> <!-- fin de mon encadrement -->
</form> <!-- fin de mon formulaire -->
</body>
</html>
```

Notre formulaire est sauvegardé sous le nom «mon\_formulaire.html».

Le programme PHP qui va récupérer les données saisies dans le formulaire peut ressembler à ceci\* :

```
<html>
<head>
<title>MON PHP</title>
</head>
<body>
<?php
echo "<h1 align='center'>BIENVENUE</h1>";
echo "VOTRE NOM EST : $_POST[nom]<br />"; // $_POST[nom] récupère la valeur donnée à "nom"
echo "VOTRE MOT DE PASSE EST : $_POST[password]"; // $_POST[password] récupère la valeur donnée à "password"
?>
</body>
</html>
```

\*je dis «peut» car il y a d'autres possibilités, mais je n'ai pas réussi à obtenir le résultat que je désirais jusqu'à maintenant alors pour cette brève introduction à PHP, je pense que ce petit programme est suffisant...

De plus, pour un travail plus élaboré, plus avancé, je pense que l'utilisation de Java s'impose mais c'est mon opinion propre.

A vous de voir lorsque vous aurez effectué votre premier programme PHP... ;+)  
et votre premier programme Java, bien entendu...

# JSP - JAVA SERVER PAGES™

Nous y revoilà... Voici l'autre visage du langage JAVA. JSP est le moyen le plus simple d'écrire un site dynamique. Tout comme PHP, JSP est une application web s'exécutant côté serveur. On utilise le conteneur web Tomcat pour compiler les pages JSP. On peut également utiliser un serveur plus spécifique, mais ce n'est pas l'objet de cette section. Pour ceux d'entre vous qui s'intéressent aux serveurs java, lisez la documentation disponible sur la plateforme de la Gazette. Contrairement à PHP, JSP est plus efficace, plus riche, plus robuste et plus facile à utiliser.

Une page JSP peut ressembler à celle-ci :

```
<html>
<head><title>Ma Première Page JSP</title></head>
<body>
<%
// début du code Java
out.print("<h1>Ceci est ma page JSP</h1>"); // affichage à l'écran de la chaîne
%> <!-- Fin du code Java -->
</body>
</html>
```

Ceci est une page simple. Maintenant, nous allons lui apporter un peu plus de dynamisme en lui donnant un compteur de visite...

```
<% int i = 0; %> <!-- initialisation de la variable entière i -->
<html>
<head><title>Ma Première Page JSP</title></head>
<body>
<% i++; // incrémentation de i

if (i==1) { // si i strictement = 1 alors faire ; c'est une structure conditionnelle
out.println("Vous &ecirc;tes le premier visiteur"); // afficher la chaîne
}
else { // sinon faire
out.print("Visiteur numero "); // afficher la chaîne
out.println(i); // afficher la valeur de la variable i
}
%>
</body>
</html>
```

Sauvegardez ce nouveau code sous le nom "compteur.jsp" et testez-le. A chaque rafraichissement de la page, la variable i est incrémentée de 1. Amusant, non ?

Vous souvenez-vous de notre formulaire ? Vous savez celui que l'on a vu dans la partie PHP ? Et bien nous allons le reprendre... Sauf que cette fois-ci, la propriété "action" enverra à la page "my\_java.jsp" et non plus "mon\_php.php" (voir formulaire page 29).

Voici donc la récupération et l'utilisation des valeurs saisies en Java (le fichier sauvegardé se nomme "my\_java.jsp") :

```
<html>
<head><title>ma page jsp</title></head>
<body>
<center><h1>MA PAGE JSP</h1></center>
<%
String nom = request.getParameter("nom");
String password = request.getParameter("password");
// récupération des données saisies par l'utilisateur
if (nom == null || "".equals(nom)) // si le 'nom' n'est pas saisi alors faire
{
out.print("Entrez votre nom !"); // affichage de la chaine
out.print("<br />");
out.print("<a href='mon_formulaire.html'>");
out.print("Retour"); // retour au formulaire
out.print("</a>");
}
else if (password == null || "".equals(password)) // si le 'password' n'est pas saisi alors faire
{
out.print("Entrez votre mot de passe !"); // affichage de la chaine
out.print("<br />");
out.print("<a href='mon_formulaire.html'>");
out.print("Retour"); // retour au formulaire
out.print("</a>");
}
else if (password == "jimiko" || "jimiko".equals(password)) // vérification du mot de passe
{
out.print("<h2>");
out.print("BONJOUR ");
out.print(request.getParameter("nom"));
out.println("</h2>");
}
else
{
out.print("<h2>");
out.print("YOU ARE NOT ALLOWED TO SEE THIS PAGE");
out.print("</h2>");
out.print("<a href='mon_formulaire.html'>");
out.print("Retour"); // retour au formulaire
out.print("</a>");
}
%>
</body>
</html>
```

Dans cette partie, le mot de passe a été défini par le programmeur. Il faut taper "jimiko" pour pouvoir accéder à la page JSP et ce, quelque soit le 'nom' de l'utilisateur saisi. Bien entendu, il est possible de créer un programme exigeant, à la fois, un nom d'utilisateur précis et le mot de passe correspondant. C'est d'ailleurs ce que l'on fera après avec JSTL...

Cet exemple est, certes, très simple. Mais, ainsi, il est plus facile pour un débutant de comprendre JSP et de le comparer à son concurrent PHP. Par exemple, l'instruction "OUT.PRINT" est plus compréhensible que "ECHO". N'êtes-vous pas d'accord ? Maintenant, passons aux choses sérieuses avec JSTL...

# JSTL™

JSTL signifie JSP Standard Tag Library. Nous allons nous en servir afin de simplifier notre code JSP mais aussi de l'améliorer en performance.

Voici donc ce que nous allons faire : créer un programme en JSTL qui exigera à la fois un nom d'utilisateur et le mot de passe correspondant à cet utilisateur... Le formulaire origine est toujours celui vu à la page 29 et il renvoie toujours à la page "my\_java.jsp" qui désormais ressemblera à ceci :

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<html>
<head><title>Ma Page JSTL</title></head>
<body>

<!-- Première condition -->
<c:if test='${param.password == "jimiko" and param.nom == "ODAE EDO"}'>
<%-- vérification des données saisies, on les compare à "jimiko" et "ODAE EDO"
      si les données entrées par l'utilisateur correspondent, alors exécuter ce qui suit..
      [ceci est un commentaire java] --%>
<center>
<h1>BONJOUR ${param.nom}</h1>
</center>
</c:if>

<!-- seconde condition au cas où l'on ne saisit rien -->
<c:if test='${param.password == "" and param.nom == ""}'>
<center>
<h1>Il vous faut un nom d'utilisateur et un mot de passe<br />
pour pouvoir voir cette page<br />
</h1>
<a href="mon_formulaire.html">Retour</a>
</center>
</c:if>

<!-- pour tous les autres cas, mauvais nom d'utilisateur, mauvais mot de passe ...
      tel quel, le programme n'affichera rien... Il y a juste une simple page vierge... -->

</body>
</html>
```

Ce n'est qu'un simple exemple... Vous êtes libre de choisir vos propres instructions... Pour ma part, lorsque les conditions auraient été remplies, j'aurais redirigé l'utilisateur vers une autre page, correspondant à son nom d'utilisateur. Et pour les autres cas non prévus, j'aurais créé un code HTML avertissant l'utilisateur. Ce qui est tout de même mieux qu'une page blanche comme dans notre exemple... ;+)

## QUELQUES COMMENTAIRES SUR JSP ET JSTL :

Cela ne l'a, peut être, pas été clairement dit, mais pour développer en JSP, JSTL et les autres dérivés de Java, il est nécessaire d'installer le JDK de Sun Microsystems. Je vous renvoie à la partie JAVA pour cela.

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

Cette première instruction, dans notre code JSTL, indique l'utilisation d'une bibliothèque d'actions ou plutôt d'une bibliothèque de tags.

“<c:if>” est une action définie dans cette bibliothèque, et, est alors utilisable dans la page.

JSTL permet de séparer Java et HTML.

Ainsi, le développeur web n'a pas à se préoccuper du code Java et peut se concentrer sur la mise en page du code HTML...

L'utilisation d'une feuille de style CSS est, d'ailleurs, vivement recommandée.

## HISTORIQUE DE L'INFORMATIQUE

1945 : Alan Turing travaille sur les premiers ordinateurs modernes

1952 : John Von Neumann participe à la création du premier ordinateur utilisant un programme mémorisé : MANIAC I. L'architecture dont il est à l'origine 'la machine de Von Neumann' sera la base de tous les ordinateurs modernes.

1962 : Création d'un réseau de communication militaire par l'U.S. Air Force qui se base sur un système décentralisé.

1964 : Paul Baran met au point un réseau hybride d'architectures étoilées et maillée

1969 : Création du réseau expérimental ARPANET  
par l'Advanced Research Projects Agency  
Création du langage PASCAL par N. WIRTH  
Création du système d'exploitation UNIX  
IBM crée le SGML (Structured Generalized Markup Language)

1970 : Apparition du langage C, conçu par D.Ritchie et B.W.Kernighan  
Invention du modèle relationnel par E.F. Codd

1972 : Création du courrier électronique par Ray Tomlinson

**Microsoft**

1975 : Création de Microsoft™ Corporation par Bill Gates et Paul Allen

1976 : Création de Apple™ par Steve Jobs et Steve Wozniak



1977 : Invention du premier langage 'SQL' par IBM (IBM SEQUEL)

1978 : Création du langage TEX par Donald E. Knuth  
Création de la méthode MERISE

1981 : IBM crée le Personal Computer.  
Création du système MS-DOS par Microsoft qui se base sur QDOS inventé par Seattle Computer Products que Bill Gates rachète

1982 : Création de LaTeX, par Leslie Lamport, basé sur Tex

1986 : Création de PEARL (Practical Extraction and Report Language)  
par Larry Wall  
Naissance du standard SQL qui est normalisé par l'ANSI

1989 : Amélioration du langage C, création du C++ par Bjarne Stroustrup

1982 : Création de Sun Microsystems™ par Andy Bechtolsheim, Bill Joy,  
Vinod Khosla et Scott McNealy



1985 : Création de la société NeXT Computer par Steve Jobs



1989 : Proposition d'un réseau hypertexte par le Centre Européen de la Recherche Nucléaire, basé à Genève

1990 : Mise en fonction du réseau conçu par le CERN  
Naissance du World Wide Web fonctionnant sur NeXT  
Naissance du HTML basé sur le SGML



1991 : Création du langage Java par Sun Microsystems  
Création du premier système d'exploitation LINUX par Linus Torvalds  
qui se base sur le système UNIX



1993 : Création de MOSAIC, le premier navigateur web tournant sur UNIX

1994 : Rasmus Lerdorf invente le PHP



1995 : Netscape et Sun Microsystems créent le Javascript  
Création du W3C chargé de standardiser le World Wide Web

1996 : Microsoft crée le langage ASP (Active Server Pages)  
Création des feuilles de style CSS

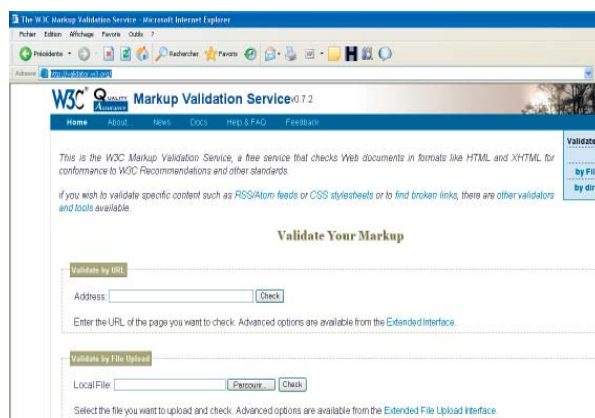


# LA VALIDATION HTML

Comme nous vous l'avons déjà dit, HTML a été normalisé par le W3C qui, pour une harmonie parfaite des pages disponibles sur le réseau, recommande aux programmeurs web de respecter les normes du langage. En effet, une bonne page est une page qui respecte la syntaxe grammatical du HTML, voir du XHTML. Mais, comment savoir si l'on respecte la syntaxe du langage ? La réponse est très simple! Le W3C met à disposition des web-masters un service de validation HTML/XHTML. Ainsi, il est possible de vérifier que notre code HTML est conforme, ou non, à la norme du 'HTML 4.01', du 'XHTML 1.0', ou bien du 'XHTML 1.1' qui est la version la plus rigoureuse du HTML. La page principale de la plateforme de la Gazette du DEUST est d'ailleurs conforme au 'XHTML 1.1'.

Voici donc comment faire pour valider vos pages:

Tout d'abord, connectez-vous à l'adresse suivante : <http://validator.w3.org/>



Le W3C vous propose trois possibilités pour vérifier la validité de vos documents...

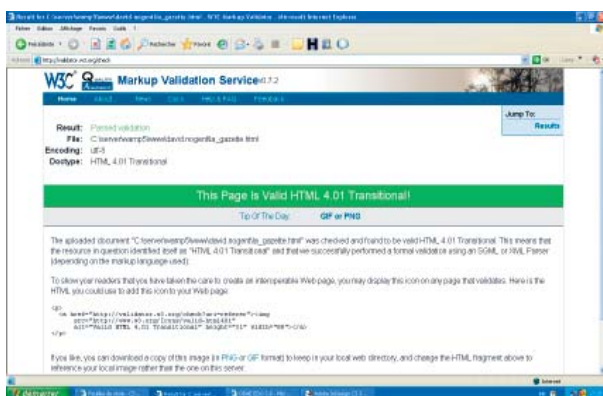
1°) si votre page HTML est en ligne, vous devez saisir l'adresse de votre page dans la section "VALIDATE BY URL" puis cliquez sur le bouton "Check". Le résultat apparaît alors!

2°) si votre fichier est sur votre ordinateur, vous pouvez "l'uploader" pour qu'il soit analysé. Pour cela, dans la section "VALIDATE BY FILE UPLOAD", cliquez sur le bouton "Parcourir", sélectionnez votre fichier, puis cliquez sur le bouton "Check". Le résultat s'affiche au bout de quelques secondes.

3°) la troisième possibilité de tapez votre code dans la section "VALIDATE BY DIRECT INPUT". Tapez votre code HTML dans la fenêtre, puis cliquez sur le bouton "Check". Le résultat s'affiche un instant plus tard.

Si votre code est conforme, le W3C vous propose d'insérer son logo comme gage de qualité.

Si votre code n'est pas conforme, le W3C vous énumère les erreurs de syntaxes. Cela peut s'avérer très fastidieux mais la qualité de votre code vaut bien cela...



Si votre fichier est conforme, une fenêtre comme celle-ci s'affiche...



Si votre fichier n'est pas conforme, une fenêtre comme celle-ci s'affiche...



# TRUCS ET ASTUCES

## Question de sécurité !

Pour assurer la sécurité d'un serveur, il est recommandé d'insérer un fichier intitulé "index.html" à la racine de chaque dossier du serveur. En effet, par défaut, lorsqu'une adresse internet pointe vers un dossier, il affiche la page "index.html". Si cette page "index.html" est inexistante et qu'il n'y a pas d'autres configurations possibles, le contenu du dossier visé par l'adresse internet s'affichera dans le navigateur. Et ceci peut poser quelques soucis de sécurité au cas où un pirate passerait par là...

## Le Serveur du W3C...



Le W3C utilise son propre serveur web. Il s'agit d'un serveur http Java intitulé "JIGSAW", né d'un projet Open Source du W3C débuté en Mai 1996. En juillet 2005, la version 2.2.5 a été réalisée. Il peut être téléchargé sur le site du W3C à l'adresse suivante : <http://www.w3.org/Jigsaw/>. L'objectif du W3C est d'augmenter la portabilité de Java sur le réseau car peu de serveurs compatibles java sont disponibles gratuitement.

## J2SE 6

Sun Microsystems travaille actuellement à l'élaboration d'une nouvelle mouture de son JDK. La version Java 2 Standard Edition 6, appelé "Mustang", devrait remplacer la version 5.

Pour en savoir plus, <http://java.sun.com/javase/6/>.

Une version beta est d'ailleurs proposée en téléchargement.

## PHP6

A peine on passe à PHP5 qui tarde à s'imposer comme standard PHP, le groupe PHP travaille sur l'élaboration de son successeur PHP6. Je rappelle, pour mémoire, que le logiciel EasyPHP ne fonctionne, actuellement, qu'avec la version 4 de PHP. Espérons que la prochaine version du logiciel prendra directement en charge la version 6. Histoire de pas trop être en retard.

## VALIDATION HTML...

Si vous souhaitez valider vos codes HTML, vous devez impérativement indiquer le type d'encodage. Il se déclare ainsi "`<meta http-equiv='Content-Type' content='text/html; charset=shift_jis'>`". "Charset" peut aussi être égale à "utf-8".

## Où en est-on avec le nouveau Windows ?

On peut dire que le nouveau Windows attise les passions. En effet, Microsoft avait d'abord annoncé la sortie du remplaçant de Windows XP pour les fêtes de Noël 2005, puis pour le début de l'année 2006. Jusqu'alors, seule la version Beta 2 de Windows Vista est sortie. Ceux qui ont eu la chance de la tester ont tous été séduit par le nouveau design du bureau du système d'exploitation.



Les gadgets de Microsoft ajoutés au système sont parfois attrayant comme un calendrier autonome, un anti-spyware, un logiciel P2P afin de faciliter le travail en collaboration et à distance et une mise à jour toujours plus intelligente. Toutefois, la beta 1 et 2 du futur Windows ont un petit problème... Durant les tests que nous avons effectués sur les deux OS, nous avons rencontré un léger problème: impossibilité d'installer le moindre anti-virus. Espérons que la version finale sera compatible avec nos anti-virus actuels car s'il faut changer de logiciel à chaque nouveau Windows, cela risque de nous revenir plus cher que prévu... La réponse à cette question arrivera prochainement. En attendant, Microsoft a déjà fait connaître sa stratégie commerciale...

Windows Vista sera disponible en 5 éditions (plus une sixième réservée aux pays émergents): 3 versions à destination du Grand Public (Home Basic, Home Premium, et Ultimate) et 2 à destination des professionnels (Business et Enterprise). "Home Basic" constitue la version bas de gamme de Windows. Curieusement, les versions qui ont le plus de fonctionnalités sont les versions "Home Premium" et "Ultimate" au détriment également des versions professionnelles qui semblent bridées.

## INTERNET ET LES LANGUES COMPLEXES!

Vous avez peut-être remarqué, en naviguant sur le web, que les caractères complexes, comme le Chinois, le Coréen, le Japonais, le Thaï, l'Hébreu, l'Arabe et j'en passe, ne passaient pas toujours correctement sur les pages HTML.

Si comme moi, vous devez insérer des caractères asiatiques ou autres dans vos pages HTML, il existe une solution efficace pour éviter d'éventuels désagréments. Lancez le logiciel Word. Créez un nouveau document. Tapez vos textes asiatiques ou autres et sauvegardez votre document au format HTML. Automatiquement, le logiciel va convertir votre saisie complexe en utilisant UNICODE. Une fois votre document sauvegardé, ouvrez le document HTML avec le bloc-note (Notepad) afin de pouvoir lire le code source. En accédant au code source, vous allez découvrir que votre texte est converti en code ISO, de la forme "&#2344;". Ainsi, le texte chinois 受欢迎的 est converti en "&#21463; &#27426; &#36814; &#30340;". Récupérez ces codes ISO et insérez-les dans votre code HTML... Cette fois, votre texte passera partout, car le code ISO est compréhensible par tous les navigateurs.

## LE MOT DE LA FIN...

Il me faut terminer ce numéro. J'ai essayé de présenter de façon simple les principaux langages de programmations pour le web. J'espère que, dans l'urgence, je n'ai pas été trop maladroit. L'erreur est facile surtout lorsqu'il est question de programmation. Les langages sont nombreux et très riches. Si bien qu'il n'est pas possible de tout voir en quelques pages.

Mais, si ces quelques pages vous ont donné envie d'apprendre sérieusement et d'aller plus loin, alors, j'ai rempli avec succès ma mission. La Gazette du DEUST TSIC a pour vocation d'aider les étudiants de la section dans leur formation, par l'apport de complément aux cours, mais également, en revenant sur des notions de cours. Lors des séances de regroupement, je sens que certains élèves sont perdus et que l'aide que peut leur apporter la Gazette est essentielle. La Gazette les aide à se remettre sur la bonne voie. J'espère que l'année prochaine marquera un tournant pour la Gazette, avec un contenu de plus en plus riche. Mais, également une participation forte des étudiants, à la fois, aux regroupements et à la Gazette. Bonne chance à toutes et à tous pour vos examens.