

Plan du cours



Conception d'un site WEB

- ◆ Historique
- ◆ HyperDocuments
- ◆ Site *Web*
- ◆ Atelier de développement
- ◆ Pages statiques : langage *HTML*
- ◆ Programmation *CGI* : langage *Perl*
- ◆ Pages dynamiques : *Javascript*

Jean DEMARTINI © 14.02.00

Historique



Conception d'un site WEB

- ◆ **Projet "World Wide Web" présenté en 1989, au CERN (Genève), par *Tim Berners Lee* : pour un *réseau mondial d'échange d'informations*.**
- ◆ **De nombreux éléments favorables :**
 - développement du réseau de transport *Internet*,
 - système distribué pour l'annuaire des machines (DNS),
 - besoin de la communauté mondiale des physiciens (10 000 chercheurs),
 - une idée simple,
 - compétences de l'initiateur du projet.

Jean DEMARTINI © 14.02.00

HyperDocuments



Conception d'un site WEB

- ◆ **Structure de document adaptée à la consultation informatique :**
 - petites pages,
 - liens de navigation entre pages,
 - exemples : documents Videotex, aides en ligne...
- ◆ **Etendue au Multimedia :**
 - liens vers des "pages sonores" ou des "pages video".
- ◆ **La consultation d'un hyperdocument nécessite des outils de navigation adaptés : butineurs (browsers), navigateurs, fureteurs ...**

Jean DEMARTINI © 14.02.00

Navigateurs célèbres



Conception d'un site WEB

- ◆ **Un des premiers : *Mosaic*.**
- ◆ **Le plus répandu : *Netscape navigator*.**
- ◆ **Le dernier en date : *Internet Explorer*.**
- ◆ **Un navigateur joue plusieurs rôles :**
 - présentation d'un hyperdocument,
 - accès aux pages distantes en suivant les liens.
- ◆ **De nombreux greffons de présentation : les plugins.**
 - documents à imprimer : formats .pdf ...
 - séquences sonores : formats .au, .wav ...
 - images fixes : formats .gif, .jpeg ...
 - images animées : formats .avi, .mov, .mpeg ...

Jean DEMARTINI © 14.02.00

WYSIWYG ? sûrement pas !

Conception d'un site WEB



- ◆ **Brisons immédiatement un mythe actuel.**
- ◆ **C'est la structure d'un hyperdocument qui est décrite et non pas sa présentation : langage *HTML* (*HyperText Mark Up Language*).**
- ◆ **Le navigateur interprète cette description puis compose le document "au mieux" compte tenu du contexte.**
 - résolution de l'écran,
 - couleurs disponibles,
 - polices disponibles,
 - taille des images à afficher,
 - taille de la fenêtre d'affichage.

Jean DEMARTINI © 14.02.00

Un type de document, une structure

Conception d'un site WEB



- ◆ **Un roman se lit :**
 - les pages se suivent,
 - quelques notes de bas de page,
 - une table des matières très sommaire.
- ◆ **Un rapport technique se consulte :**
 - les pages se suivent,
 - des notes de bas de page,
 - des renvois dans le document (texte, figures...),
 - des renvois en annexe,
 - une table des matières fournie,
 - un index,
 - une bibliographie (renvois hors du texte).

Jean DEMARTINI © 14.02.00

Documents interactifs

Conception d'un site WEB



- ◆ **Quelques constatations :**
 - un écran est petit (en général),
 - lire un écran est pénible,
 - changer d'écran est rapide.
- ◆ **Certains documents gagnent à être interactifs :**
 - notices techniques,
 - documents de présentation,
 - documentations commerciales.
- ◆ **Certains documents ne peuvent pas être interactifs :**
 - documents de réflexion.

Jean DEMARTINI © 14.02.00

Un journal, un site Web

Conception d'un site WEB



- ◆ **Un journal est conçu autour de sa *une* qui :**
 - identifie le journal (logo, typographie, mise en page),
 - attire l'oeil (couleurs, iconographie),
 - donne envie d'acheter le journal pour lire la suite des articles introduits "à la une".
- ◆ **Un service Web poursuit souvent les mêmes objectifs et sera structuré de la même façon.**
- ◆ **Un détail important :**
 - les frais de consultation, et donc l'assiduité du client, sont fonction du temps de chargement des pages.

Jean DEMARTINI © 14.02.00

Structure d'un hyperdocument

Conception d'un site WEB



- ◆ Un hyperdocument est articulé autour de sa page d'accueil : la *home page*.
- ◆ La *home page* représente un compromis difficile :
 - doit être attrayante, colorée, animée,
 - doit être claire,
 - doit se charger vite !
- ◆ Le document doit être structuré de telle sorte qu'on ne s'y perde pas :
 - danger de la présence de trop de liens externes.

Jean DEMARTINI © 14.02.00

Edition sur le Web

Conception d'un site WEB



- ◆ Le réseau peut simplement servir de support de diffusion pour un ouvrage :
 - pas de frais d'impression,
 - pas de frais de distribution.
- ◆ Le service correspondant a la structure d'une librairie virtuelle et ne propose qu'un catalogue d'ouvrages.
- ◆ Les documents "intéressants" sont sous une forme imprimable essentiellement :
 - format *postscript*,
 - format *acrobat*.

Jean DEMARTINI © 14.02.00

Localisation d'un document

Conception d'un site WEB

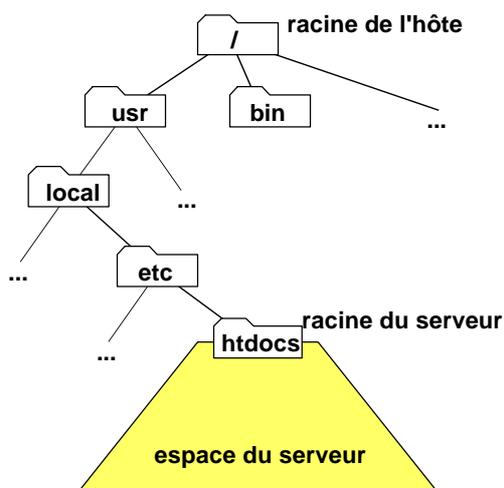


- ◆ Un *document Web* est repéré par son *URL* (Unique Resource Location).
- ◆ URL la plus simple :
`protocole://serveur/`
- ◆ URL courante :
`protocole://serveur/repert/.../document`
- ◆ Quelques exemples :
`http://www.inria.fr/`
`http://www.ti.com/`
`http://www.yahoo.com/`
`http://verdon.unice.fr/~jdem/`

Jean DEMARTINI © 14.02.00

Architecture de fichiers Web

Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Protection des documents

Conception d'un site WEB

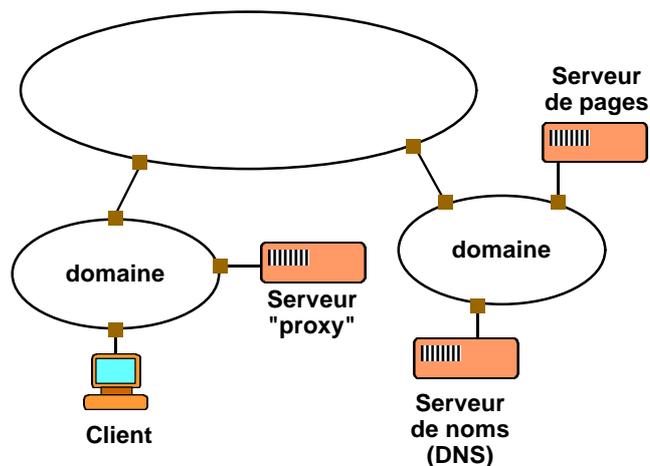


- ◆ **Le système de protection est double :**
 - protection de base des fichiers par le système d'exploitation de l'hôte (Unix, Windows NT ...),
 - protection des documents par le serveur lui-même.
- ◆ **Le serveur Web peut autoriser l'accès :**
 - aux lecteurs enregistrés : *nom, mot de passe*
 - aux lecteurs accédant depuis un domaine particulier (au sens TCP/IP),
 - aux lecteurs accédant depuis une machine particulière.
- ◆ **Attention. Nous verrons qu'il est possible, dans certains cas, de contourner ces protections.**

Jean DEMARTINI © 14.02.00

World Wide Web

Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Quelques protocoles utiles

Conception d'un site WEB



- HTTP :** protocole d'échange de documents HTML (HyperText Transfert Protocol)
- FTP :** protocole d'échange de fichiers quelconques (File Transfert Protocol).
- SMTP :** protocole d'échange de courrier électronique.
- TELNET :** protocole de connexion à distance.

Jean DEMARTINI © 14.02.00

Atelier de développement

Conception d'un site WEB



- ◆ **Tout au long de ce cours nous supposerons que nous nous trouvons dans la configuration suivante :**
 - le serveur est hébergé sur une machine *Unix*, le logiciel-serveur Web étant *Apache*.
 - le poste de développement est une machine en réseau de type PC sous *Windows 95* ou *Windows NT*.
- ◆ **Le poste de développement utilisé :**
 - un outil pour la conception de sitesWeb : Dreamweaver de Macromedia,
 - un outil de dessin pour la construction ou la mise en forme des images : Fireworks de Macromedia,
 - un outil de transfert de fichiers : WS_FTP,
 - un outil de connexion à distance : telnet,
 - un navigateur : Internet Explorer 5.0.

Jean DEMARTINI © 14.02.00

Outils HTML



Conception d'un site WEB

- ◆ **Il existe de nombreux éditeurs HTML :**
 - Front Page de Microsoft,
 - HoT Metal de SoftQuad,
 - Homesite de Allaire,
 - Dreamweaver de Macromedia, ...
- ◆ **Ceux qui permettent une édition WYSIWYG donnent des résultats très intéressants aujourd'hui.**
- ◆ **Il existe également des convertisseurs de format pouvant donner d'assez mauvais résultats.**
 - deux exceptions : LateX2HTML et RTF2HTML.

Jean DEMARTINI © 14.02.00

Outils de dessin



Conception d'un site WEB

- ◆ **Les images utilisées sous de type bitmap.**
- ◆ **Il existe de nombreux outils de dessin bitmap :**
 - Paint Shop Pro de JASC,
 - 3D Studio,
 - Fireworks de Macromedia, ...
- ◆ **Un outil de dessin doit être capable de :**
 - lire et produire différents formats d'image : .gif et .jpeg au moins,
 - de produire des dessins à fond transparent,
- ◆ **l'interfaçage avec un scanner (format twain par exemple) est utile.**

Jean DEMARTINI © 14.02.00

Outils de validation



Conception d'un site WEB

- ◆ **La présentation d'une page Web dépendant du navigateur, il est nécessaire de disposer de quelques navigateurs sous différentes versions pour tester le *look* d'une page :**
 - Netscape 3.0 et 4.x — les versions 4.x est la plus répandue,
 - Internet Explorer 4.0 et 5.0 — les versions antérieures sont beaucoup trop rustiques.
 - C'est la version Internet Explorer 5.x qui, aujourd'hui permet l'accès au plus grand nombre de sites.

Jean DEMARTINI © 14.02.00

Outils de transfert de fichiers



Conception d'un site WEB

- ◆ **Le service est construit sous la forme d'une maquette dans l'espace disque du poste de développement. Lorsqu'il est validé, il faut le transférer dans l'espace disque du serveur.**
- ◆ **WS_Ftp de IPSwitch donne d'excellents résultats.**

Jean DEMARTINI © 14.02.00

Outil de connexion à distance

Conception d'un site WEB



- ◆ Certains paramétrages et même certains développements (programmation CGI) ne peuvent se faire et/ou se valider que sur le serveur lui-même.
- ◆ Il est donc indispensable de pouvoir se connecter à distance.
- ◆ QVT/Net de QPC Software donne d'excellents résultats.

Jean DEMARTINI © 14.02.00

Le langage HTML

Conception d'un site WEB



- ◆ Ces quelques URL <http://...> vous donneront accès à une information complémentaire.
 - www.w3.org/
 - www.w3.org/TR/REC-html32.html
 - nphi.unice.fr/CoursHTML/
 - www.cc.ukans.edu/~acs/docs/other/HTML_quick.shtml
- ◆ Le langage HTML décrit la structure d'un document à l'aide de balises qui délimitent un élément du document
 - `<NOM_de_BALISE paramètres>...</NOM_de_BALISE>`
- ◆ Tout document HTML est délimité par :
 - `<HTML>...</HTML>`

Jean DEMARTINI © 14.02.00

Structure d'un document HTML

Conception d'un site WEB



- ◆ Un document HTML comprend toujours 2 éléments :
 - entête : `<HEAD>...</HEAD>`
 - corps : `<BODY paramètres>...</BODY>`
- ◆ La structure de base d'un document HTML est alors :

```
<HTML>
  <HEAD>
  ...
  </HEAD>
  <BODY paramètres>
  ...
  </BODY>
</HTML>
```

Jean DEMARTINI © 14.02.00

En-tête de document

Conception d'un site WEB



- ◆ Une en tête comprend deux parties :
 - un titre de la page,
 - une base de référence
- ◆ Le titre de la page est décrit par :
`<TITLE>...</TITLE>`
- ◆ La base de référence définit la racine d'accès par défaut à toutes les URL utilisées dans la page :
`<BASE HREF="URL">`
- ◆ Si la base de référence n'est pas définie, toutes les URL utilisées sont repérées par rapport au répertoire de la page en cours.

Jean DEMARTINI © 14.02.00

Ma première page HTML

Conception d'un site WEB



```
<HTML>
  <HEAD>
    <TITLE>Ma premi&egrave;re page</TITLE>
  </HEAD>
  <BODY>
    Ma premi&egrave;re page Web.
  </BODY>
</HTML>
```

définition du caractère "è"

Jean DEMARTINI © 14.02.00

Et pour la voir ?

Conception d'un site WEB



- ◆ **Tout d'abord, il faut taper le texte précédent en utilisant un éditeur de textes (pas un traitement de textes) :**
 - EDIT de DOS
 - NOTEPAD de Windows 3.x
 - WORDPAD de Windows 95 (en mode texte).
- ◆ **Cette page est, par exemple, la page d'accueil de votre futur service :**
 - la sauvegarder sous le nom `index.html` dans le répertoire qui servira de racine à la maquette de votre service.
- ◆ **Utiliser un navigateur, hors connexion, en simple consultation de fichier local.**

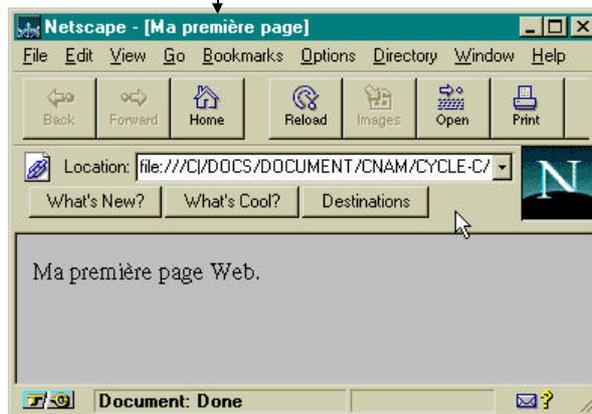
Jean DEMARTINI © 14.02.00

La voici !



Conception d'un site WEB

Son titre



Jean DEMARTINI © 14.02.00

Quelques précisions



Conception d'un site WEB

- ◆ Il est souvent utile d'inclure des commentaires dans votre description de page :
<!-- auteurs : Tom & Jerry -->
- ◆ Le texte en commentaire n'est pas interprété par le navigateur.
- ◆ Un premier pseudo-commentaire est utilisé par le navigateur pour déterminer la version du langage HTML utilisée dans la page :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
```

Jean DEMARTINI © 14.02.00

Caractères spéciaux

Conception d'un site WEB



- ◆ La plupart des alphabets comportent des caractères utilisant des signes diacritiques qui sont indisponibles sur le clavier utilisé. Ces caractères ont donc été codés :
- ◆ Exemples :

é	é	ô	ô
è	è	â	â
ê	ê	ê	ê
à	à	ï	ï
ü	ü	ë	ë
ç	ç		

Jean DEMARTINI © 14.02.00

Paragrapes...

Conception d'un site WEB



- ◆ En fait notre première page a été acceptée par le navigateur *par pure gentillesse* car elle contient un paragraphe non signalé. Il faudrait la décrire par :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2
Final//EN">

<HTML>
  <HEAD>
    <TITLE>Ma premi&egrave;re page</TITLE>
  </HEAD>
  <BODY>
    <P>Ma premi&egrave;re page Web.</P>
  </BODY>
</HTML>
```

Jean DEMARTINI © 14.02.00

...Paragrapes



Conception d'un site WEB

- ◆ La meilleure présentation d'un paragraphe est déterminée par le navigateur lorsqu'il compose la page.
- ◆ Pour faciliter la compréhension de la structure du document HTML, on présente, en général, les paragraphes sur une seule ligne (pouvant largement déborder de l'écran de saisie).

Jean DEMARTINI © 14.02.00

Ma deuxième page HTML



Conception d'un site WEB

- ◆ On remarque les indentations qui facilitent la lecture.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2
Final//EN">

<HTML>
  <HEAD>
    <TITLE>Ma premi&egrave;re page</TITLE>
  </HEAD>
  <BODY>
    <P>Ma premi&egrave;re page Web.</P>
    <P>Un deuxi&egrave;me paragraphe permet
de      la meubler un peu.</P>
  </BODY>
</HTML>
```

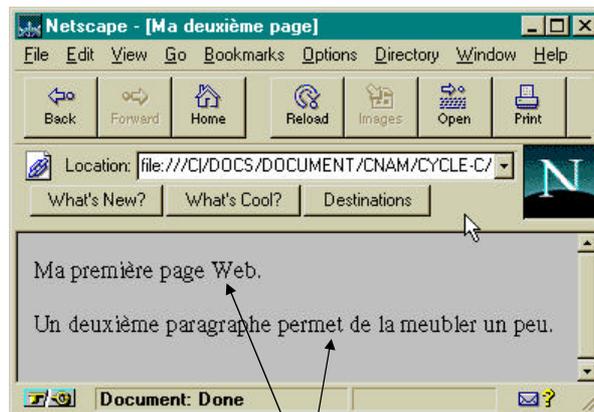
Jean DEMARTINI © 14.02.00

Que voici



Conception d'un site WEB

33



Les paragraphes

Jean DEMARTINI © 14.02.00

Éléments d'une page



Conception d'un site WEB

34

- ◆ Une page peut comporter :
 - des paragraphes,
 - des titres de paragraphes,
 - des images,
 - des ancres (liens),
 - des listes à puces, à numéros, descriptives,
 - des tableaux,
 - des formulaires,
 - des décorations.
- ◆ Des portions de textes peuvent être mises en évidence en utilisant différents styles.

Jean DEMARTINI © 14.02.00

Titres de paragraphes...

Conception d'un site WEB



◆ On dispose de 6 niveaux de titres :

`<Hn paramètres>...</Hn>` avec $n = 1..6$

```
<BODY>
  <H1 ALIGN="LEFT">Niveau 1</H1>
  <H2 ALIGN="CENTER">Niveau 2</H2>
  <H3 ALIGN="RIGHT">Niveau 3</H3>
  <H4 ALIGN="JUSTIFY">Niveau 4 : Ce titre est
    censé être justifié ; &circ;re
    justifié ;.</H4>
</BODY>
```

Jean DEMARTINI © 14.02.00

...Titres de paragraphes

Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Styles de paragraphes...

Conception d'un site WEB



◆ Les paragraphes peuvent être :

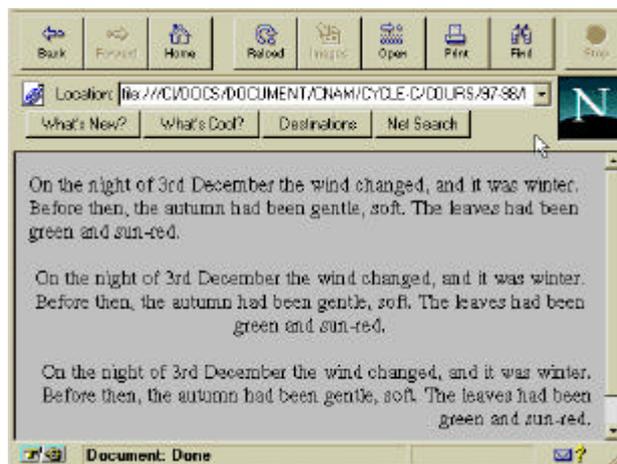
- alignés à gauche,
- alignés à droite,
- centrés.

```
<BODY>  
    <P ALIGN="LEFT">On the night of 3rd December  
the  
    <P ALIGN="CENTER">On the night of 3rd December  
th  
    <P ALIGN="RIGHT">On the night of 3rd December  
</BODY>
```

Jean DEMARTINI © 14.02.00

...Styles de paragraphes

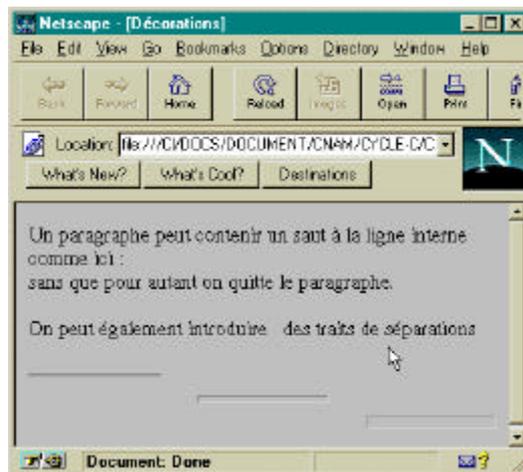
Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Ou bien cela !

Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Multicolonnage

Conception d'un site WEB



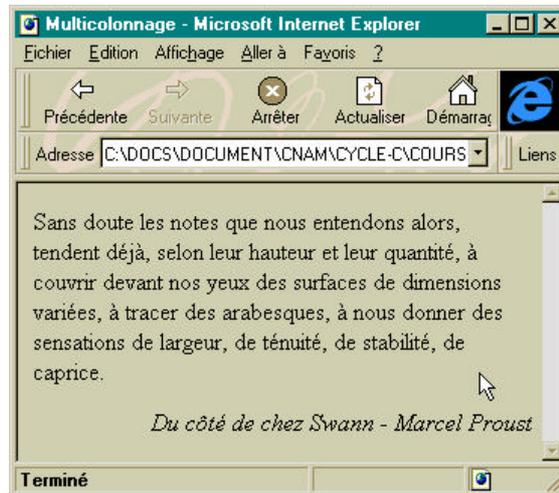
- ◆ Certains navigateurs permettent le multicolonnage du texte.

```
<BODY>  
<MULTICOL COLS="2" WIDTH="100%" GUTTER="10">  
<P ALIGN="JUSTIFY">Sans doute les notes que nous entendons  
<P ALIGN="RIGHT"><i>Du c&ocirc;rc;t&eacute;acut; de chez Swann - M  
</MULTICOL>  
</BODY>
```

Jean DEMARTINI © 14.02.00

Ce qui donne ceci...

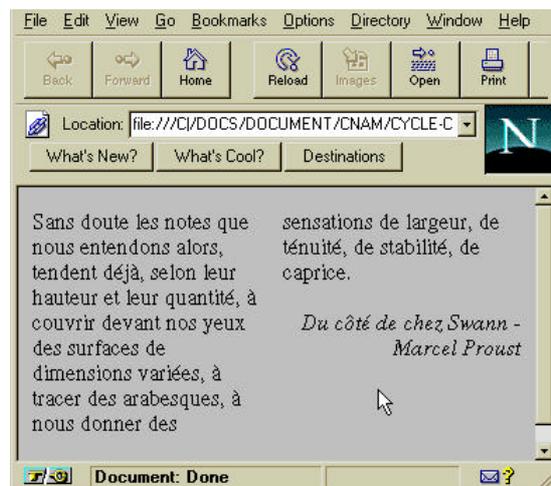
Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Ou bien cela

Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Styles physiques

Conception d'un site WEB



- ◆ Les styles physiques sont des conteneurs permettant de définir le style d'affichage d'une portion de texte :

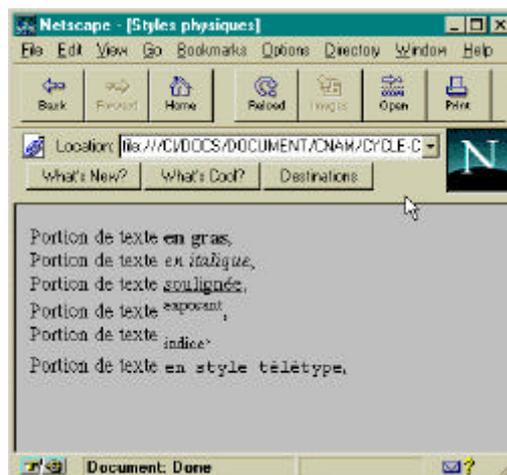
<code>...</code>	en gras,
<code><I>...</I></code>	en italique,
<code><U>...</U></code>	soulignée,
<code><SUP>...</SUP></code>	en exposant,
<code><SUB>...</SUB></code>	en indice,
<code><TT>...</TT></code>	en style télétype.

```
Portion de texte <B>en gras</B>,<BR>
Portion de texte <I>en italique</I>,<BR>
Portion de texte <U>souligné</U>,<BR>
Portion de texte <SUP>exposant</SUP>,<BR>
Portion de texte <SUB>indice</SUB>,<BR>
Portion de texte <TT>en style télétype</TT>.
```

Jean DEMARTINI © 14.02.00

Ce qui donne

Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Styles logiques

Conception d'un site WEB



- ◆ Les styles logiques sont des conteneurs permettant de définir un style d'affichage laissé à la discrétion du navigateur :

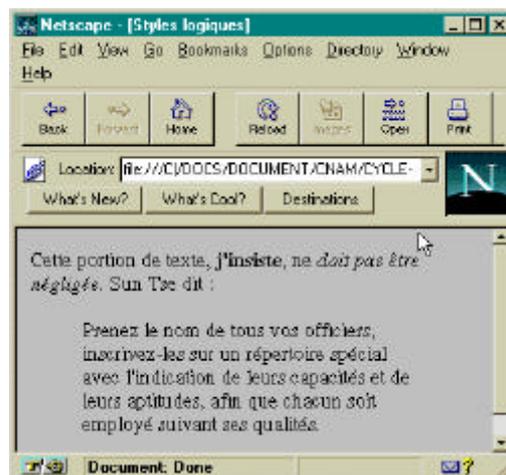
- `...` pour insister,
- `...` pour mettre en évidence,
- `<BLOCKQUOTE>...</BLOCKQUOTE>` une citation,
- `<PRE>...</PRE>` tel quel,
- `<ADRESS>...</ADRESS>` une adresse.

```
<P>Cette portion de texte, <STRONG>j'insiste</STRONG>, ne  
<EM>doit pas être négligée</EM>. Sun Tse dit  
</P>  
<P><BLOCKQUOTE>Prenez le nom de tous vos officiers, inscrivez-  
les sur un répertoire spécial avec l'indication de leurs  
capacités et de leurs aptitudes, afin que chacun soit  
employé suivant ses qualités.</BLOCKQUOTE></P>
```

Jean DEMARTINI © 14.02.00

Présentation Netscape 3.0

Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Les listes



Conception d'un site WEB

◆ HTML permet la définition de 3 types de listes :

- listes à puce : `......`
- listes numérotées : `......`
- listes descriptives : `<DL>...<DT>...<DD>...</DL>`

◆ Les puces peuvent être choisies parmi :



◆ Les numérotations peuvent être choisies parmi :

- 1. 2. 3. ...
- a. b. c. ...
- A. B. C. ...
- i. ii. iii. ...
- I. II. III. ...

Jean DEMARTINI © 14.02.00

Listes à puces



Conception d'un site WEB

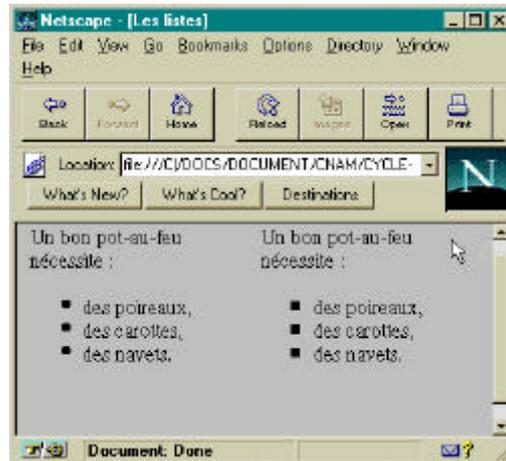
```
<MULTICOL COLS="2" WIDTH="100%" GUTTER="10">
  <P>Un bon pot-au-feu nécessite :</P>
  <UL TYPE="DISC">
    <LI>des poireaux,
    <LI>des carottes,
    <LI>des navets.
  </UL>
  <BR>
  <P>Un bon pot-au-feu nécessite :</P>
  <UL TYPE="SQUARE">
    <LI>des poireaux,
    <LI>des carottes,
    <LI>des navets.
  </UL>
</MULTICOL>
```

Jean DEMARTINI © 14.02.00

Présentées ainsi



Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Les listes numérotées



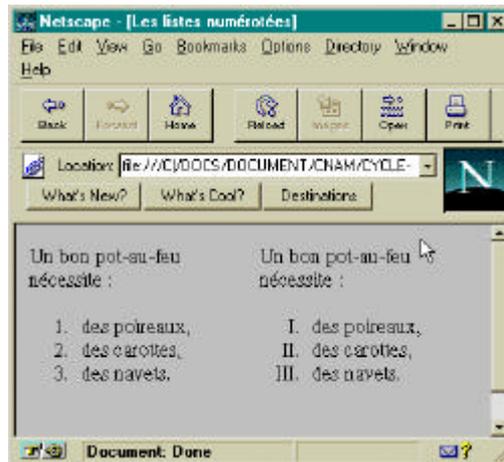
Conception d'un site WEB

```
<MULTICOL COLS="2" WIDTH="100%" GUTTER="10">
  <P>Un bon pot-au-feu nécessite :</P>
  <OL TYPE="1">
    <LI>des poireaux,
    <LI>des carottes,
    <LI>des navets.
  </OL>
  <BR>
  <P>Un bon pot-au-feu nécessite :</P>
  <OL TYPE="1">
    <LI>des poireaux,
    <LI>des carottes,
    <LI>des navets.
  </OL>
</MULTICOL>
```

Jean DEMARTINI © 14.02.00

Présentées ainsi

Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Les listes de définitions

Conception d'un site WEB



- ◆ Ce type de liste est particulièrement indiqué pour construire un glossaire ou un dictionnaire.

```
<DL>
  <DT>FLAMBARD<DD>n. m. <I>Fam.</I> Fanfaron, orgueill
  <DT>FLAMBEAU<DD>n. m. Torche, chandelle de cire ou
  <DT>FLAMB&Eacute;E<DD>n. f. Feu clair de menu bois.
</DL>
```

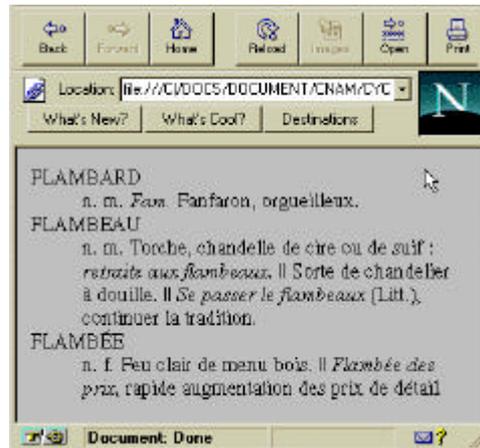
- ◆ Si les termes à définir sont courts, on peut la compacter.

```
<DL COMPACT>
  <DT>N<DD>Newton.
  <DT>Nm<DD>Newtonm&egrave;tre.
  <DT>Oe<DD>Oersted.
</DL>
```

Jean DEMARTINI © 14.02.00

Présentées ainsi

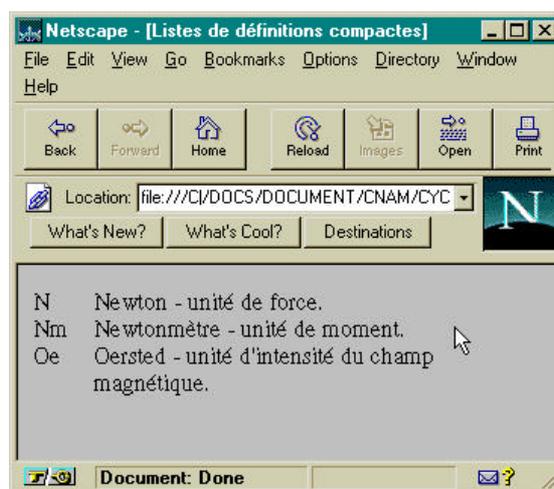
Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Ou ainsi

Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Listes imbriquées

Conception d'un site WEB



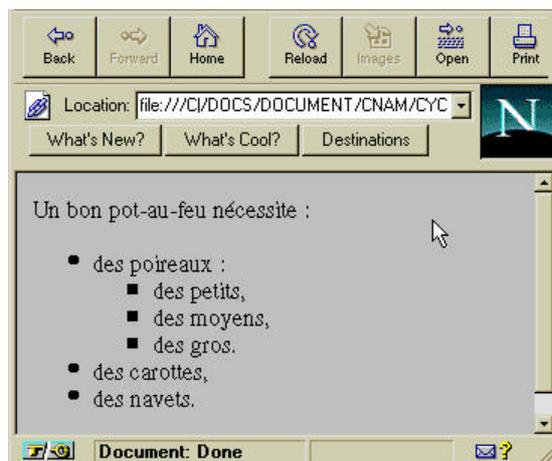
- ◆ Des listes de différentes natures peuvent être imbriquées

```
<P>Un bon pot-au-feu nécessite :</P>
<UL TYPE="DISC">
  <LI>des poireaux :
    <UL TYPE="SQUARE">
      <LI>des petits,
      <LI>des moyens,
      <LI>des gros.
    </UL>
  <LI>des carottes,
  <LI>des navets.
</UL>
```

Jean DEMARTINI © 14.02.00

Ce qui donne

Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Les images



Conception d'un site WEB

- ◆ Les pages Web sont souvent parsemées de petits éléments décoratifs obtenus par inclusion d'images.

``

- ◆ La balise d'inclusion d'images comporte de très nombreux paramètres :

- ALIGN= TOP | MIDDLE | BOTTOM | LEFT | RIGHT
- HSPACE= *marge horizontale (pixels)*
- VSPACE= *marge verticale (pixels)*
- WIDTH= *largeur de l'image (pixels)*
- HEIGHT= *hauteur de l'image (pixels)*
- BORDER= *épaisseur de l'encadrement (pixels)*

Jean DEMARTINI © 14.02.00

Inclusion d'une image



Conception d'un site WEB

```
<P> <IMG SRC="Image16.gif"
WIDTH=200
HEIGHT=145
BORDER=0
ALIGN="LEFT"
HSPACE=20
VSPACE=10>
```

Sans doute les notes que nous entendons alors, tendent à couvrir devant nos yeux des surfaces de dimensions variées, à tracer des arabesques, à nous donner des sensations de largeur, de ténacité, de stabilité, de caprice.

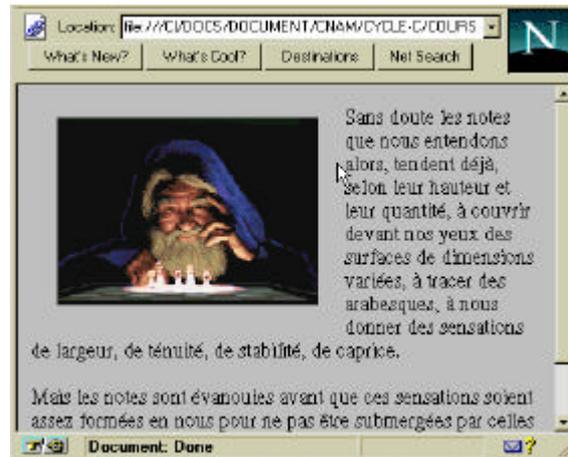
Mais les notes sont évanouies avant que ces sensations soient assez formées en nous pour ne pas être submergées par celles qu'elles veulent couvrir; les notes suivantes ou simultanées.

Jean DEMARTINI © 14.02.00

Aspect visuel



Conception d'un site WEB



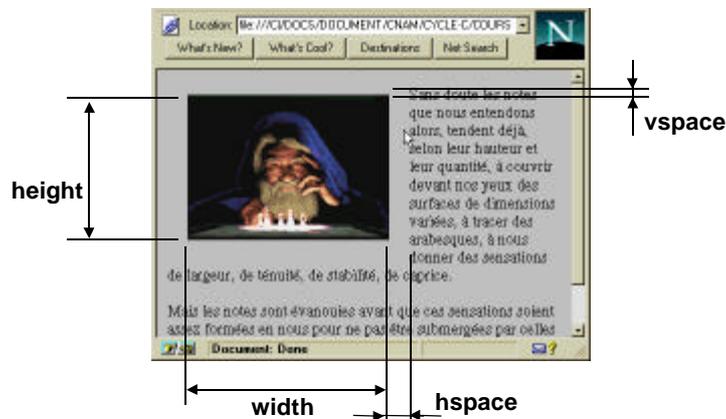
Jean DEMARTINI © 14.02.00

Paramètres d'inclusion



Conception d'un site WEB

◆ Interprétation des paramètres d'inclusion :



Jean DEMARTINI © 14.02.00

Paramètres d'alignement

Conception d'un site WEB



```
<P ALIGN="CENTER">
  <IMG SRC="Image17a.gif"></P>
<P>Cette image
  <IMG SRC="Image17b.gif" ALIGN="TOP">
  est alignée;
  <IMG SRC="Image17b.gif" ALIGN="MIDDLE">
  de différentes;
  <IMG SRC="Image17b.gif" ALIGN="BOTTOM">
  façons.</P>
<P>On remarque que son fond a été rendu transparent.</P>
```

Jean DEMARTINI © 14.02.00

Effets de présentation

Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Fond d'écran

Conception d'un site WEB



- ◆ Une image peut également servir de fond d'écran.

```
<BODY BACKGROUND="Image18.gif">  
<P ALIGN="CENTER"><IMG SRC="Image17a.gif"  
<P><B>Cette image <IMG SRC="Image17b.gif"  
<P><B>On remarque que son fond &agrave; &eacu  
</BODY>
```

Jean DEMARTINI © 14.02.00

Image de fond et transparence

Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Les ancres



Conception d'un site WEB

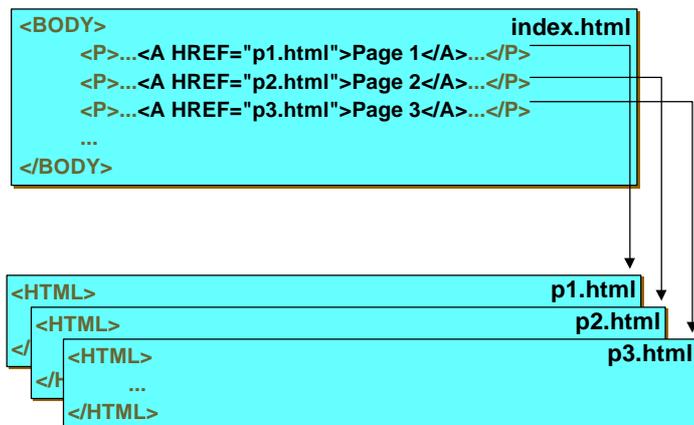
- ◆ Les ancres indiquent des *extrémités* de liens vers d'autres documents.
- ◆ Un lien peut pointer vers :
 - un autre document : *lien externe*,
 - un autre endroit de la page en cours : *lien interne*.
- ◆ Une ancre peut ainsi définir un *point de départ* :
`cliquable`
- ◆ Elle peut également signaler un *point d'arrivée* :
`non cliquable`

Jean DEMARTINI © 14.02.00

Schéma de liens externes simples



Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Schéma de liens internes

Conception d'un site WEB



```
<BODY>
  <P>...<A HREF="#p1">Page 1</A>...</P>
  ...
  <P>...<A HREF="#p2">Page 2</A>...</P>
  ...
  <P>...<A HREF="#p3">Page 3</A>...</P>
  ...
  <P><A NAME="p1"></A>...</P>
  ...
  <P><A NAME="p2"></A>...</P>
  ...
  <P><A NAME="p3"></A>...</P>
  ...
</BODY>
```

Jean DEMARTINI © 14.02.00

Liens doubles

Conception d'un site WEB



- ◆ Les liens doubles définissent à la fois un point de départ et un point d'arrivée.

```
<BODY>
  <P>...<A HREF="#p1" NAME="r1">Page 1</A>...</P>
  ...
  <P><A HREF="#r1" NAME="p1">Retour</A>...</P>
  ...
</BODY>
```

Jean DEMARTINI © 14.02.00

Une page avec des liens

Conception d'un site WEB



```
<H2 ALIGN="CENTER">Tout sur la fougasse</H2>
<P ALIGN="CENTER">
  <A HREF="chap-1.html">Chapitre 1 : La farine</A><BR>
  <A HREF="chap-2.html">Chapitre 2 : Les anchois</A></P>
<HR ALIGN="CENTER">
<P ALIGN="CENTER">Aimez-vous la fougasse ?</P>
<P ALIGN="CENTER">
  <A HREF="chap-1.html"><IMG SRC="bout-1.gif"></A>
  <A HREF="chap-2.html"><IMG SRC="bout-2.gif"></A></P>
```

Jean DEMARTINI © 14.02.00

Qui s'affiche

Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Les Tableaux



Conception d'un site WEB

- ◆ Le tableau est l'élément d'organisation le plus puissant de HTML.
- ◆ C'est lui qui permet de disposer dans une page de différentes zones de texte ou d'images.
- ◆ un tableau est un pavage de cellules (CELLS), associées à des cellules de titre (HEADING CELLS) le tout associé à une légende (CAPTION).
- ◆ Sa description est assez complexe, nous l'aborderons donc pas à pas.

Jean DEMARTINI © 14.02.00

Un tableau élémentaire



Conception d'un site WEB

Un tableau élémentaire

	Roues	Volant	Remorque(s)
Voiture	4	1	0 ou 1
Vélo	2	0	0
Camion	4 et plus	1	1 à 10*

* en Australie, un *road train* peut comporter jusqu'à 10 remorques attelées à un tracteur.

Annotations: CAPTION (points to the title), HEADING CELLS (points to the header row), CELLS (points to the data rows).

Jean DEMARTINI © 14.02.00

Structure d'un tableau

Conception d'un site WEB



- ◆ Un tableau est introduit par le conteneur :
`<TABLE paramètres>...</TABLE>`
- ◆ Le titre d'un tableau est introduit par le conteneur :
`<CAPTION paramètres>...</CAPTION>`
- ◆ Une ligne du tableau (ROW) est introduite par le conteneur :
`<TR paramètres>...</TR>`
- ◆ Une ligne est composée de cellules* :
`<TH paramètres>...</TH>`
`<TD paramètres>...</TD>`

* pouvant contenir tout ce que l'on veut et même un tableau.

Jean DEMARTINI © 14.02.00

Un tableau élémentaire 4x4

Conception d'un site WEB



```
<TABLE paramètres>
  <CAPTION paramètres>...</CAPTION>
  <TR>
    <TD>...</TD><TH>...</TH><TH>...</TH><TH>...</TH>
  </TR>
  <TR>
    <TH>...</TH><TD>...</TD><TD>...</TD><TD>...</TD>
  </TR>
  <TR>
    <TH>...</TH><TD>...</TD><TD>...</TD><TD>...</TD>
  </TR>
  <TR>
    <TH>...</TH><TD>...</TD><TD>...</TD><TD>...</TD>
  </TR>
</TABLE>
```

Jean DEMARTINI © 14.02.00

Principaux paramètres

Conception d'un site WEB



- ◆ On dispose de nombreux paramètres :
 - alignement horizontal,
 - alignement vertical,
 - couleur de fond,
 - couleur de texte,
 - image de fond
 - ...
- ◆ Lorsqu'un paramètre n'est pas défini, sa valeur est héritée de la structure qui l'englobe.
- ◆ Malheureusement, tous ces paramètres ne sont pas traités par les navigateurs, ils peuvent ne pas avoir d'effet.

Jean DEMARTINI © 14.02.00

Paramètres de <TABLE ...>

Conception d'un site WEB



	Roues	Volant	Remorque(s)
Voiture	4	1	0 ou 1
Vélo	2	0	0
Camion	4 et plus	1	1 à 10*

Diagram illustrating the parameters of a table: CELLPADDING, CELLSPACING, and BORDER.

Jean DEMARTINI © 14.02.00

Paramètres de <CAPTION

Conception d'un site WEB



ALIGN = LEFT | CENTER | RIGHT

Un tableau élémentaire ←

	Roues	Volant	Remorque(s)
Voiture	4	1	0 ou 1
Vélo	2	0	0
Camion	4 et plus	1	1 à 10*

← VALIGN = TOP | BOTTOM

Jean DEMARTINI © 14.02.00

Superposition de cellules

Conception d'un site WEB

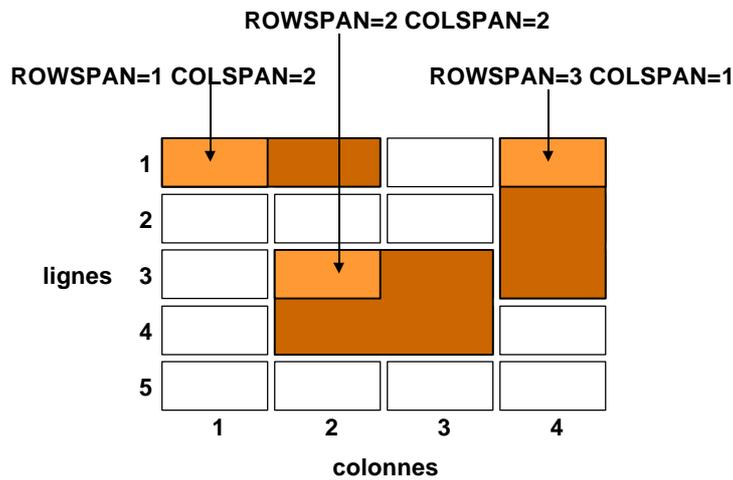


- ◆ On peut faire en sorte qu'une cellule en recouvre d'autres donnant l'impression que des cellules ont été regroupées.
- ◆ L'extension d'une cellule sur ses voisines de droite correspond au paramètre
COLSPAN = nombre de cellules
- ◆ L'extension d'une cellule sur ses voisines du dessous correspond au paramètre
ROWSPAN = nombre de cellules

Jean DEMARTINI © 14.02.00

Paramètres de superposition

Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Une belle table des matières

Conception d'un site WEB



AOL PRESS **Table of Contents**

<u>Introduction</u>				
Basics	Creating Pages	Organizing Your Pages	Tips & Tricks	Glossary
<ul style="list-style-type: none">◆ Opening a Page◆ Creating a Page◆ Save to Your Computer◆ Publishing to the Web	<ul style="list-style-type: none">◆ Creating Text◆ Creating Lists◆ Creating Tables◆ Adding Images◆ Creating Links◆ Creating Forms◆ Creating Frames			

 [[Home](#) | [Glossary](#) | [Index](#)]
Copyright © 1997 America Online, Inc.

Jean DEMARTINI © 14.02.00

Une belle page



Conception d'un site WEB

83

Tous les savoirs du monde :
L'encyclopédisme en Chine

1. Une tradition encyclopédique de 2000 ans
2. Un traité d'agriculture
3. Une encyclopédie en la Chine

1. Une tradition encyclopédique riche de deux mille ans d'histoire

La tradition encyclopédique chinoise est une tradition ancienne qui remonte au **début du 1er millénaire** avant notre ère. Elle y apparaît dotée d'une vigueur particulière, alimentée par une série de facteurs convergents : goût pour la citation, pensée par catégorie, mode de recrutement de l'intelligentsia impliquant la composition littéraire académique.

Léguer à son fils une corbeille remplie d'or est moins précieux que lui léguer un Classique.
(dicton chinois)

Son histoire peut se raconter en plusieurs épisodes.

- En 239 av. J.-C., **Lǐ Bùwēi**, un riche marchand devenu premier ministre de l'État de **Qin**, invita trois mille lettrés à la Cour et les pria d'écrire ce qu'ils avaient entendu. Il rassembla leurs propos en huit chapitres d'observations, six chapitres d'entretiens et douze chapitres de recueils de notes,



*Le premier répertoire de mots classés, IV^e-II^e siècle avant notre ère
Paris, Bouf*

Jean DEMARTINI © 14.02.00

Images cliquables



Conception d'un site WEB

84

- ◆ Il est possible de définir des **zones sensibles** sur une image bitmap et d'associer des liens à ces zones.
- ◆ Nous ne verrons que la technique la plus récente.
- ◆ Les zones sensibles définies sur une image sont regroupées dans une carte (MAP) associée à l'image.

Jean DEMARTINI © 14.02.00

Carte des zones sensibles

Conception d'un site WEB



```
<IMG SRC="Image16.gif" WIDTH="200" HEIGHT="145"
  ISMAP
  USEMAP="#Image16" BORDER="0">
<MAP NAME="Image16">
  <AREA SHAPE="CIRCLE" HREF="chap-1.html"
    COORDS="86,64,20">
  <AREA SHAPE="RECT" HREF="chap-2.html"
    COORDS="61,101,92,138">
</MAP>
```

Jean DEMARTINI © 14.02.00

Les formulaires

Conception d'un site WEB



- ◆ C'est le formulaire qui sert d'interface entre le monde du Web et le monde de l'informatique.
- ◆ Il permet de construire des interfaces de présentation à des logiciels d'application.
- ◆ Il utilise le mode CGI (Common Gateway Interface) pour fonctionner et son utilisation nécessite la coopération d'au moins 2 acteurs :
 - Côté client : Le formulaire HTML,
 - Côté serveur : un programme d'interface.

Jean DEMARTINI © 14.02.00

Dialogue Client-Serveur

Conception d'un site WEB

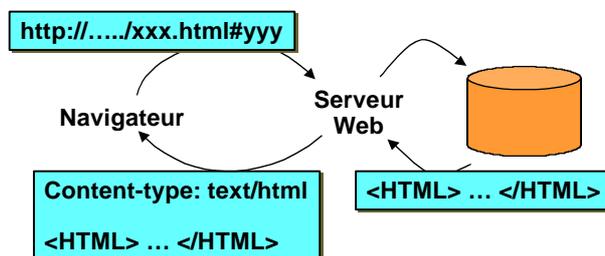


- ◆ Le côté *client* est représenté par le navigateur qui interprète la page HTML qui comporte des balises pouvant se traduire en requêtes.
- ◆ Le côté *serveur* est représenté par le serveur Web qui interprète les requêtes provenant du navigateur et qui lui transmet des pages HTML.
- ◆ Nous avons rencontré deux balises se traduisant en requête :
 - `<A ... >...` : requête pour une page externe,
 - `` : requête pour une image externe.

Jean DEMARTINI © 14.02.00

Schéma de dialogue HTML

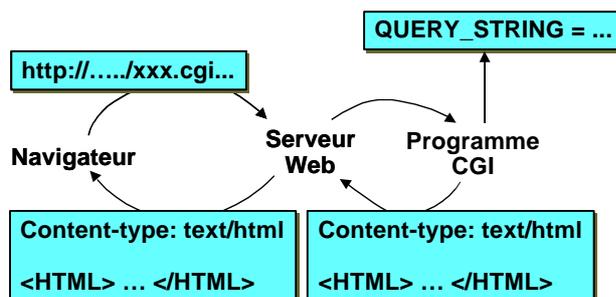
Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Schéma de dialogue CGI

Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Formulaire

Conception d'un site WEB



- ◆ Un formulaire est introduit par le conteneur :
`<FORM paramètres> ... </FORM>`
- ◆ Un formulaire est constitué d'un groupe d'outils de saisie décrits par des conteneurs de la forme :
`<INPUT paramètres> ... </INPUT>`
`<SELECT paramètres> ... </SELECT>`
- ◆ Un formulaire comporte 2 boutons :
`<INPUT TYPE=SUBMIT paramètres>`
`<INPUT TYPE=RESET paramètres>`

Jean DEMARTINI © 14.02.00

Exemple 1



Conception d'un site WEB

91

```
<FORM ACTION="query.cgi" ...  
  METHOD= "GET" | "POST">  
<P><BIG><BIG>P</BIG></BIG>our faire une bonne mayonnai  
<INPUT TYPE="radio" NAME="liquide" VALUE="eau">eau,  
<INPUT TYPE="radio" NAME="liquide" VALUE="huile">huile,  
<INPUT TYPE="radio" NAME="liquide" VALUE="lait">lait  
</FORM>
```

Concours du meilleur cuisinier

Pour faire une bonne mayonnaise, il faut :

eau, huile, lait

Jean DEMARTINI © 14.02.00

Exemple 2



Conception d'un site WEB

92

```
<FORM ACTION="query.cgi" ...  
  METHOD= "GET" | "POST">  
...  
<P><BIG><BIG>I</BIG></BIG> faut aussi :</P>  
<INPUT TYPE="checkbox" NAME="oeufs" VALUE="on">oeufs,  
<INPUT TYPE="checkbox" NAME="sel" VALUE="on">sel,  
<INPUT TYPE="checkbox" NAME="mouta" VALUE="on">moutarde,  
<INPUT TYPE="checkbox" NAME="confit" VALUE="on">confiture  
</FORM>
```

Pour faire une bonne mayonnaise, il faut :

eau, huile, lait

Il faut aussi :

oeufs, sel, moutarde, confiture

Jean DEMARTINI © 14.02.00

Exemple 3



Conception d'un site WEB

93

```
<FORM ACTION="query.cgi"
  METHOD= "GET" | "POST">
...
<P><BIG><BIG>O</BIG></BIG>n utilise :
<SELECT NAME="recipient" SIZE="3">
  <OPTION SELECTED>assiette
  <OPTION>bol
  <OPTION>bouteille
  <OPTION>casserole
  <OPTION>marmite</SELECT> </P>
</FORM>
```

Il faut aussi :

oeufs, sel, moutarde, confiture

On utilise :

Jean DEMARTINI © 14.02.00

Exemple 4



Conception d'un site WEB

94

```
<FORM ACTION="query.cgi" METHOD= "GET" | "POST">
...
<P><BIG><BIG>D</BIG></BIG>&eacute;crivez, en quelques lignes
les principales qualit&eacute;s di&eacute;t&eacute;tiques de
la mayonnaise :<BR>
<TEXTAREA NAME="dissertation" ROWS="5" COLS="20">
</TEXTAREA>
</FORM>
```

Décrivez, en quelques lignes les principales qualités diététiques de la mayonnaise :

Jean DEMARTINI © 14.02.00

Conclusion provisoire

Conception d'un site WEB



- ◆ Nous avons vu l'essentiel de ce qu'il était nécessaire de connaître de HTML pour réaliser de belles pages Web.
- ◆ N'hésitez pas à examiner le source des pages que vous consultez. Vous y découvrirez comment réaliser des effets époustouflants.
- ◆ Il faut, à présent, étudier ce qui se passe du côté du serveur (programmation CGI) et c'est la suite de ce cours.

Jean DEMARTINI © 14.02.00

Programmes CGI

Conception d'un site WEB



- ◆ Un programme CGI doit être capable de composer une page HTML et de l'émettre sur sa sortie standard "stdout".
- ◆ Ce programme reçoit ses arguments sous différentes formes que nous examinerons.
- ◆ Attention : un programme CGI mal conçu crée des trous de sécurité dans le système qui l'héberge et pour en savoir plus sur les problèmes de sécurité dans le Web consulter :

<http://www.w3.org/Security/Faq/www-security-faq.html>

Jean DEMARTINI © 14.02.00

Programmation CGI

Conception d'un site WEB

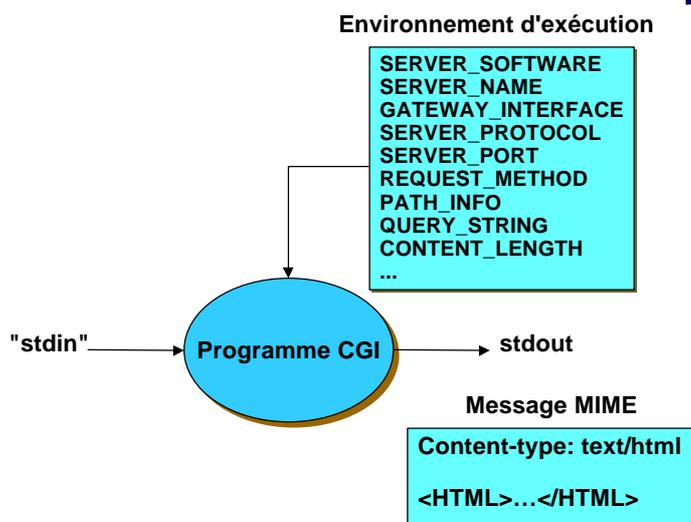


- ◆ Un programme CGI peut être écrit dans n'importe quel langage.
- ◆ Cependant certains langages sont plus commodes que d'autres.
- ◆ On utilise typiquement :
 - des langages interprétés tels *sh*, *csh*, ... et surtout *Perl*,
 - des langages compilés tels *C*, *C++*, *FORTRAN* et maintenant *Visual Basic*® dans un environnement Windows NT ®.
- ◆ Nous allons introduire, dans ce cours, le langage le plus utilisé actuellement : *Perl*.

Jean DEMARTINI © 14.02.00

Entrées - Sorties CGI

Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Invocation d'un programme CGI

Conception d'un site WEB



- ◆ On dispose principalement de 3 méthodes pour invoquer un programme CGI :
 - invocation directe en tant que cible d'un lien :
...
 - invocation par soumission d'un formulaire avec la méthode GET :
<FORM ACTION="foo.cgi" METHOD="GET" ...>
... </FORM>
 - invocation par soumission d'un formulaire avec la méthode POST :
<FORM ACTION="foo.cgi" METHOD="POST" ...>
... </FORM>

Jean DEMARTINI © 14.02.00

Extension de l'URL

Conception d'un site WEB



- ◆ On peut passer les arguments d'un programme CGI en étendant son URL :
...
- ◆ C'est cette technique qui est utilisée lors de la soumission d'un formulaire utilisant la méthode GET.
 - L'extension de l'URL est effectuée par le navigateur lui-même.
- ◆ Les arguments (empaquetés dans une grande chaîne de caractères) peuvent être obtenus en analysant les variables d'environnement SCRIPT_NAME, PATH_INFO et QUERY_STRING.

Jean DEMARTINI © 14.02.00

Entrée standard



Conception d'un site WEB

- ◆ Lorsqu'un formulaire est soumis en utilisant la méthode POST, les arguments sont empaquetés dans une grande chaîne de caractères présentée sur l'entrée stdin.
- ◆ Cette grande chaîne ne se termine pas par EOF !
- ◆ Sa longueur est placée dans la variable d'environnement CONTENT_LENGTH.

Jean DEMARTINI © 14.02.00

Empaquetage des arguments



Conception d'un site WEB

- ◆ Un argument est représenté sous la forme d'un couple nom=valeur.
- ◆ Une valeur est une chaîne de caractères,
- ◆ Chaque argument est séparé du précédent et du suivant par le caractère &.
- ◆ Les espaces sont remplacés par le caractère + et les caractères spéciaux sont remplacés par leur code ASCII(hex) %xx.
- ◆ Exemple :
liquide=eau&œufs=on&sel=on&recipient=bol&dissertation=Et+bien,la+mayonnaise,+c'est+tr%E8s+bon...&nom=Mickey+Mouse

Jean DEMARTINI © 14.02.00

En résumé



Conception d'un site WEB

- ◆ **Un programme CGI doit :**
 - avoir un nom dont l'extension est .cgi,
 - être capable de consulter des variables d'environnement,
 - être capable de lire une chaîne de caractères présentée sur stdin,
 - être capable de l'analyser,
 - être capable de composer une page HTML au format MIME et de la présenter sur stdout.
- ◆ **Il doit pouvoir, également, être capable de communiquer avec d'autres applications.**
- ◆ **Le langage *Perl* permet de faire tout ça très commodément.**

Jean DEMARTINI © 14.02.00

Le langage *Perl*



Conception d'un site WEB

- ◆ **Le langage *Perl* (Practical Extraction and Report Language) ne peut pas être considéré comme un langage de programmation orthodoxe. Sa seule ambition est d'être *commode*.**
- ◆ **Nous n'en donnerons pas une présentation exhaustive et nous ne l'introduirons qu'à partir d'exemples significatifs.**
- ◆ **Pour en savoir plus, consulter :**
 - <http://www2.ec-lille.fr/~seguy/francais/perl/>

Jean DEMARTINI © 14.02.00

Un programme Perl simplissime

Conception d'un site WEB



```
#!/usr/local/bin/perl

print "Content-type: text/html

<HTML>
<HEAD>
  <TITLE>Première page
scriptés</TITLE>
</HEAD>
<BODY>
  <P>Bonjour à tous.</P>
</BODY>
</HTML>
";
```

- ◆ Bien que très simple, ce premier programme nous apprend déjà plein de choses.

Jean DEMARTINI © 14.02.00

Les commentaires

Conception d'un site WEB



- ◆ Un commentaire est une portion de texte commençant par le caractère # et s'étendant jusqu'à la fin de la ligne.
- ◆ Exemples :
 - # Que ce programme est beau et bien écrit !
 - print 'bonjour tout le monde !'; # qu'est-ce donc ?

Jean DEMARTINI © 14.02.00

Localisation de l'interprète

Conception d'un site WEB



- ◆ Dans un environnement Unix, tous les fichiers interprétés (au moins pour tous les "shell" que je connais) indiquent, en première ligne, le chemin d'accès à leur interprète.
- ◆ Cette indication prend la forme du pseudo-commentaire :

`#!/usr/local/bin/perl`

- ◆ C'est cette indication qui permet de considérer un tel fichier comme un programme exécutable standard.

Jean DEMARTINI © 14.02.00

Les instructions

Conception d'un site WEB



- ◆ Une instruction est constituée :
 - soit d'une commande et de ses paramètres,
 - soit de l'affectation d'une variable à la valeur d'une expression.
- ◆ Une instruction se termine toujours par le caractère ;
- ◆ Ce premier programme ne comporte que l'instruction
 - `print "...";`
- ◆ Comme on peut s'y attendre, la commande `print` imprime (ici sur `stdout`).

Jean DEMARTINI © 14.02.00

Quote '...' & quasi-quote "..."

Conception d'un site WEB



- ◆ Une portion de texte qui ne doit pas être interprétée est citée entre quotes :

'bonjour tout le monde !'

- ◆ Lorsqu'une telle portion de texte est répartie sur plusieurs lignes, elle inclut les "RC" invisibles.
- ◆ Lorsqu'on cherche à construire des portions de texte, il est commode d'y inclure des variables qui seront évaluées. La portion de texte est alors citée entre quasi-quotes :

"Je voudrais \$n pizzas"

"Je voudrais une \${c}croute garnie"

Jean DEMARTINI © 14.02.00

Formats MIME

Conception d'un site WEB



- ◆ Le texte envoyé au navigateur par le serveur doit se conformer à la norme MIME (Multipurpose Internet Mail Extension) qui préconise d'indiquer en en-tête le type de ce qui suit, ici :

Content-type: text/html

- ◆ Cet en-tête doit être suivi d'une ligne vide !
- ◆ D'autres types MIME :

application/pdf
application/x-compress
audio/basic
image/gif

...

Jean DEMARTINI © 14.02.00

Variables



Conception d'un site WEB

- ◆ **Perl supporte 3 types de variables :**
 - *scalaires* : nombres et chaînes de caractères,
 - *tableaux* de scalaires dont les éléments sont indexés par un nombre entier,
 - *tableaux associatifs* dont les éléments sont indexés par un symbole.
- ◆ **Noms de variables, les noms de variables sont préfixés :**
 - par \$ pour les scalaires,
 - par @ pour les tableaux,
 - par % pour les tableaux associatifs.
- ◆ **Les caractères \$, @ et % définissent, plus généralement, le contexte d'interprétation des opérateurs Perl.**

Jean DEMARTINI © 14.02.00

Contextes d'interprétation



Conception d'un site WEB

- ◆ **Les différents contextes d'interprétation peuvent être illustrés par ces exemples :**

● \$jour	simple variable scalaire
● \$jours[28]	29ème élément du tableau @jours
● \$nombre_de_jours{'mars'}	élément associé au symbole 'mars' du tableau associatif %nombre_de_jours
● \$#jours	dernier indice dans le tableau @ jours
● @ jours	tableau dans son ensemble
● @ jours[3,4,5]	tableau constitué des éléments numéros 3, 4 et 5 du tableau @ jours
● @ jours[3..5]	même tableau que ci-dessus
● @ nombre_de_jours{'mars','avril'}	tableau constitué des éléments associés aux symboles 'mars' et 'avril' du tableau associatif % nombre_de_jours
● % nombre_de_jours	tableau associatif dans son ensemble

Jean DEMARTINI © 14.02.00

Petite généralisation

Conception d'un site WEB



- ◆ La composition d'une page HTML est décrite plus généralement par :

```
#!/usr/local/bin/perl

$title = ...;
$body = ...;

print "Content-type: text/html

<HTML>
<HEAD><TITLE>$title</TITLE></HEAD>
<BODY>$body</BODY>
</HTML>
";
```

Jean DEMARTINI © 14.02.00

Les constantes

Conception d'un site WEB



- ◆ Les constantes sont utilisées soit pour initialiser des variables, soit pour participer à l'évaluation d'une expression.
- ◆ Nombres :
18-14 3.14 0.6e-3 0x1E
- ◆ Portions de texte :
'bonjour tout le monde.' "il est \$b l'heure."
- ◆ Listes (indiquable comme un tableau) :
(18, 'est un divisible par', 2, 3, 'et ', 6)

Jean DEMARTINI © 14.02.00

Affectation



Conception d'un site WEB

- ◆ Comme toutes les opérations *Perl*, l'affectation présente quelques magies.

`$i = 3;` comme d'habitude.
`$i = '0001';` est intéressant si on veut produire un compteur conservant ses 0 à gauche.
`@jours = ('lundi', 'mardi', 'mercredi', 'jeudi', 'vendredi', 'samedi', 'dimanche');`
la liste a servi à construire un tableau.
`@week_end = @jours[5,6];`
`@jours_ouvres = jours[0..4];`
`($a, $b, $c) = (3, 4, 5);`
`($a, $b, @des_jours) = @jours;`

- ◆ Et bien d'autres magies que je vous laisse découvrir.

Jean DEMARTINI © 14.02.00

Quelques opérations scalaires



Conception d'un site WEB

<code>\$a = 1 + 2;</code>	addition
<code>\$a = 2 - 4;</code>	soustraction
<code>\$a = 5 * 6;</code>	multiplication
<code>\$a = 7 / 2;</code>	division
<code>\$a = 9 ** 3.2;</code>	puissance (pas forcément entière)
<code>\$a = 5 % 2;</code>	reste euclidien
<code>++\$a;</code>	pré-incrémentation
<code>\$a++;</code>	post-incrémentation
<code>\$a = 'bonjour ' . 'tout le monde';</code>	concaténation
<code>\$a = 'je bégaie ' x 8;</code>	répétition
<code>\$a += 3;</code>	équivalent à <code>\$a = \$a + 3</code>

Jean DEMARTINI © 14.02.00

Opérations sur les tableaux

Conception d'un site WEB



```
@fruits = ('pomme', 'poire');
```

```
push(@fruits, 'banane', 'abricot');
```

modifie @fruits qui vaut alors ('pomme', 'poire', 'banane', 'abricot').

```
pop(@fruits);
```

modifie @fruits qui vaut alors ('pomme', 'poire', 'banane') et rend 'abricot'.

Jean DEMARTINI © 14.02.00

Fonctions

Conception d'un site WEB



- ◆ Comme dans tous les langages de programmation, la fonction est un outil de structuration puissant.
- ◆ Déclaration (sub) et appel (&) d'une fonction (*Perl* ne connaît pas le λ) :
- ◆ Les arguments sont passés dans le tableau implicite @_.
- ◆ La valeur rendue est celle de la dernière expression évaluée.

```
sub moyenne {  
    (@_[0] + @_[1]) / 2;  
}  
  
print &moyenne(1,2);
```

Jean DEMARTINI © 14.02.00

Manipulation des fichiers

Conception d'un site WEB



- ◆ Voici un programme élémentaire qui manipule un fichier (pas trop gros) :

```
#!/usr/local/bin/perl
open( FICHER, 'foo.txt');    # ouverture* du fichier et
                             # création de son descripteur

@lignes = <FICHER>;         # lecture du fichier ligne
                             # par ligne et construction
                             # d'un tableau de ces lignes

close(FICHER);              # fermeture du fichier
print @lignes;              # et affichage
```

* les fichiers STDIN, STDOUT et STDERR sont ouverts a priori.

Jean DEMARTINI © 14.02.00

Variables d'environnement

Conception d'un site WEB



- ◆ Les variables d'environnement sont empaquetées dans le tableau associatif prédéfini %ENV.
- ◆ On y accède par leur nom :

```
$ENV{'REQUEST_METHOD'}
$ENV{'SCRIPT_NAME'}
$ENV{'PATH_INFO'}
$ENV{'QUERY_STRING'}
$ENV{'CONTENT_LENGTH'}
```

Jean DEMARTINI © 14.02.00

On entre dans le vif du sujet

Conception d'un site WEB



```
...
if ($ENV{'REQUEST_METHOD'} eq "POST") {
    read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
} else {
    $buffer = $ENV{'QUERY_STRING'};
}
...
```

◆ On note 3 nouveautés :

- la structure de contrôle :
if <prédicat> {<conséquence>}
else {<alternative>}
- l'opérateur de comparaison : eq
- la fonction de lecture d'un bloc de caractères dans un fichier : read().

Jean DEMARTINI © 14.02.00

Manipulation des chaînes

Conception d'un site WEB



◆ L'exemple précédent laisse en suspens deux problèmes :

- comment être sûr que la chaîne "POST" est bien en majuscules ? ou plus généralement composée de caractères de la même casse.
 - Comment analyser le contenu de la variable \$buffer qui contient typiquement le résultat de la saisie d'un formulaire (cf. diapo 104).
- ## ◆ La manipulation des chaînes de caractères en Perl utilise (comme beaucoup d'outils Unix tel *regexp*) des motifs (patterns) décrits à l'aide d'*expressions régulières*.

Jean DEMARTINI © 14.02.00

Expressions régulières

Conception d'un site WEB



- ◆ Une expression régulière est la description d'un motif textuel à l'aide d'une *formule* placée entre / et /.
- ◆ Une expression régulière est construite à partir :
 - de constantes : *caractères* (signes typographiques),
 - d'opérateurs : dénotés par des *caractères spéciaux* dont l'interprétation dépend du contexte.
- ◆ Le coin du théoricien : Les expressions régulières ont été introduites pour décrire les langages réguliers (T1) et leurs analyseurs qui sont des machines à états finis.

Jean DEMARTINI © 14.02.00

Classes de caractères

Conception d'un site WEB



- ◆ Une classe de caractères est l'ensemble des caractères qui peuvent apparaître en tel ou tel emplacement d'un motif.
- ◆ Une classe de caractères est définie entre [et].
- ◆ Dans la définition d'une classe de caractères, les signe - et ^ (placé en tête) signifient "entre" et "sauf".
- ◆ Exemples :

[abc]	les 3 caractères a, b et c
[a-zA-Z0-9]	n'importe quelle lettre ou chiffre
[^abc]	sauf les caractères a, b ou c

Jean DEMARTINI © 14.02.00

Caractères spéciaux

Conception d'un site WEB



- ◆ Certains caractères sont spéciaux car on ne peut pas les taper au clavier (ils ont un effet immédiat), exemples :
 - `\n` nouvelle ligne
 - `\t` tabulation
- ◆ D'autres servent de raccourcis syntaxique, exemples :
 - `.` : n'importe quel caractère = `[^\n]`
 - `\w` : un caractère alpha-numérique = `[a-zA-Z0-9]`
 - `\W` : sauf un caractère alpha-numérique = `[^a-zA-Z0-9]`
 - `\d` : un chiffre = `[0-9]`
 - `\D` : sauf un chiffre = `[^0-9]`
 - `\s` : un espacement = `[\t\n]`
 - `\S` : sauf un espacement = `[^\t\n]`

Jean DEMARTINI © 14.02.00

Concaténation, option et répétition

Conception d'un site WEB



- ◆ L'opération de base est la concaténation dénotée par la simple juxtaposition.
 - `/[A-Z][0-9]/` dénote des motifs de la forme B3, Z8, ...
- ◆ Un caractère peut être optionnel dans un motif :
 - `/pommes?/` dénote le mot "pomme" éventuellement au pluriel.
 - `/par(t|c)/` dénote les mots "part" ou "parc"
- ◆ Un caractère peut être dupliqué dans un motif :
 - `/b+/` dénote des motifs de la forme "b", "bb", "bbb", "bbbb", ...
 - `/b*/` = `(b+)?`

Jean DEMARTINI © 14.02.00

Parenthésage et échappement

Conception d'un site WEB



- ◆ Pour définir ce sur quoi porte une opération, il peut être nécessaire de parenthéser certains termes de l'expression :

`/(bon)+/` dénote des motifs de la forme "bon", "bonbon", "bonbonbon", ...

- ◆ Pour introduire un caractère qui joue normalement le rôle d'un opérateur, il faut le faire précéder de `\` :

`\|` `\[` `\]` `\(` `\)` `*` `\+` ...

- ◆ Il est toujours possible d'introduire un caractère spécial par son code ASCII :

`\x82` dénote (sur PC sous DOS) le caractère "é"

Jean DEMARTINI © 14.02.00

Appariement

Conception d'un site WEB



- ◆ Une chaîne est appariée à un motif si on peut y trouver ce motif.

- ◆ La relation d'appariement est notée `=~`

- ◆ Exemples :

<code>'le petit chat' =~ /le/</code>	vaut "vrai"
<code>'le petit chat' =~ /Le/</code>	vaut "faux"
<code>'un bon saucisson' =~ /s+/</code>	vaut "vrai"

- ◆ Cette relation d'appariement sera souvent utilisée en tant que prédicat dans une structure "if (...) else {...}"

Jean DEMARTINI © 14.02.00

Contexte de l'appariement

Conception d'un site WEB



- ◆ L'appariement sur une partie du motif peut être imposé en début de texte ou en fin de texte.

^ le terme qui suit est confronté au début du texte
\$ le terme qui précède est confronté à la fin du texte.

- ◆ Exemples :

`/^(bon)(.*)s$/` s'apparie avec tous les textes qui commencent par "bon" et qui se terminent par la lettre "s".

Jean DEMARTINI © 14.02.00

Substitution

Conception d'un site WEB



- ◆ Une *expression de substitution* permet de remplacer, dans un texte, un motif par un autre, elle vaut le nombre des substitutions effectuées.

- ◆ Une expression de substitution est de la forme :

`s/<reg-exp>/<expression>/<paramètres>`

- ◆ Exemples :

`s/saucisse/banane/`
remplace la première occurrence du mot "saucisse" par le mot "banane".

`s/saucisse/saucisson/g`
remplace toutes les occurrences du mot "saucisse" par le mot "banane".

Jean DEMARTINI © 14.02.00

Utilisations de la substitution (1)

Conception d'un site WEB



- ◆ Modification partielle d'une portion de texte :

```
$texte = "  
une bonne saucisse sèche est  
une saucisse qui a bien séché.";  
  
$texte =~ s/saucisse/banane/;
```

- ◆ Que devient la variable \$texte ?

Jean DEMARTINI © 14.02.00

Utilisations de la substitution (2)

Conception d'un site WEB



- ◆ Modification globale d'une portion de texte :

```
$texte = "  
une bonne saucisse sèche est  
une saucisse qui a bien séché.";  
  
$texte =~ s/saucisse/banane/g;
```

- ◆ Que devient la variable \$texte ?

Jean DEMARTINI © 14.02.00

Utilisations de la substitution (3)

Conception d'un site WEB



- ◆ Sauf indication, la substitution porte sur la variable prédéfinie* \$_.
- ◆ Modification d'une portion de texte :

```
$_ = "  
une bonne saucisse sèche est  
une saucisse qui a bien séché.";  
  
$ns = s/saucisse/banane/g;
```

- ◆ Que deviennent les variables \$_ et \$ns ?

* ces variables prédéfinies introduisent quelques magies dans Perl. Nous en verrons quelques autres.

Jean DEMARTINI © 14.02.00

Traduction

Conception d'un site WEB



- ◆ Une *expression de traduction* permet de remplacer, dans un texte, des caractères par d'autres caractères ; elle vaut le nombre des traductions effectuées.
- ◆ Une expression de traduction est de la forme :
`tr/<classe de caractères>/<classe de caractères>/<paramètres>`
- ◆ Exemples :
`tr/a-z/A-Z/`
remplace toutes les lettres minuscules par les majuscules correspondantes. Toutes ?
`tr/a-z/b-za/`
utilise le code de César pour crypter un message.

Jean DEMARTINI © 14.02.00

Utilisations de la traduction (1)

Conception d'un site WEB



- ◆ Modification d'une portion de texte :

```
$texte = "  
une bonne saucisse sèche est  
une saucisse qui a bien séché.";  
  
$texte =~ s/a-zèé/b-zaiâ/;
```

- ◆ Que devient la variable \$texte ?

Jean DEMARTINI © 14.02.00

Utilisations de la traduction (2)

Conception d'un site WEB



- ◆ Sauf indication, la traduction porte sur la variable prédéfinie \$_.
◆ Modification d'une portion de texte :

```
$_ = "  
une bonne saucisse sèche est  
une saucisse qui a bien séché.";  
  
$ntr = tr/a/a/;
```

- ◆ Que deviennent les variables \$_ et \$ntr ?

Jean DEMARTINI © 14.02.00

Mémoire de l'appariement (1)

Conception d'un site WEB



- ◆ Un motif d'appariement peut être scindé en plusieurs sous-motifs par parenthésage :
`/^(.)(.)*(.)$/` définit 3 sous-motifs
- ◆ Lorsque ce motif est apparié avec succès à une portion de texte, ce qui s'est apparié à chacun des sous-motifs est conservé dans des variables nommées `\1`, `\2`, ... dans la définition de la substitution et `$1`, `$2`, ... dans le reste du programme.

Jean DEMARTINI © 14.02.00

Mémoire de l'appariement (2)

Conception d'un site WEB



- ◆ Exemple :
`/^(.)(.)*(.)$/` confronté au texte "bonne nuit les petits" conduit à :
`\1 = $1 = "b"`
`\2 = $2 = "onne nuit les petit"`
`\3 = $3 = "s"`
- ◆ Que fait donc la portion de programme suivante ?

```
$_ = "bonne nuit les petits";  
s/^(.)(.)*(.)$/\3\2\1/;  
print "$1 $2 $3";
```

Jean DEMARTINI © 14.02.00

On retourne dans le vif du sujet

Conception d'un site WEB



```
$ENV{'REQUEST_METHOD'} =~ tr/a-z/A-Z;/

if ($ENV{'REQUEST_METHOD'} eq "POST") {
    read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
} else {
    $buffer = $ENV{'QUERY_STRING'};
}
...

```

- ◆ A présent, peu importe que le nom de la méthode soit écrit en minuscule ou en majuscule.
- ◆ Il ne reste plus qu'à exploiter le contenu de la variable \$buffer.

Jean DEMARTINI © 14.02.00

Deux fonctions très utiles

Conception d'un site WEB



- ◆ Très fréquemment il est nécessaire d'éliminer le dernier caractère d'une portion de texte (typiquement \n) :
 - chop (\$texte) élimine le dernier caractère du texte contenu dans \$texte.
 - chop élimine le dernier caractère du texte contenu dans \$_.
- ◆ Encore plus fréquemment il est nécessaire de tronçonner une portion de texte (typiquement pour analyser une QUERY_STRING) :
 - split(<motif>,\$texte) rend le tableau constitué des "tronçons" découpés dans \$texte, <motif> définissant le séparateur.
 - split(<motif>) comme ci-dessus sur la variable \$_.

Jean DEMARTINI © 14.02.00

Un tronçonnage utile

Conception d'un site WEB



```
$buffer = " liquide=eau&œufs=on&...&nom=Mickey+Mouse"  
@paires = split(/&/,$buffer);  
...
```

- ◆ Le tableau contient, maintenant, des termes de la forme :

```
$paires[0] = "liquide=eau"  
$paires[1] = "œuf=on"  
...  
$paires[..] = "nom=Mickey+Mouse"
```

Jean DEMARTINI © 14.02.00

Pour aller plus loin

Conception d'un site WEB



```
foreach $paire (@paires) {  
    ($nom, $valeur) = split(/=/, $paire);  
    $valeur =~ tr/+/ /;  
    $valeur =~ s/%(..)/pack("C", hex($1))/eg;  
    ...  
}
```

- ◆ Quatre nouveautés :
 - la structure d'itération
 foreach <scalaire> (<tableau>) {...}
 - la fonction hex()
 - la fonction pack()
 - l'option "e" dans "s/.../.../ge"

Jean DEMARTINI © 14.02.00

Structure d'itération en Perl

Conception d'un site WEB



◆ On trouve les structure d'itération suivantes :

```
foreach $var (@tab) {...}
for(<initialisation>;<test>;<incrémentation>) {...}
while (<prédicat>) {...}
do {...} while(<prédicat>)
```

◆ Les prédicats sont construits à partir ds relations suivantes :

==	égalité numérique
!=	différence numérique
eq	égalité lexicographique
ne	différence lexicographique
=~	appariement
!~	désappariement

Jean DEMARTINI © 14.02.00

Expressions logiques

Conception d'un site WEB



◆ Les expressions logiques sont construites à partir des connecteurs suivants :

!	négation logique (pas)
&&	conjonction logique (et)
	disjonction logique (ou)

◆ Ces connecteurs sont évalués en *mode paresseux*. Le 2^{ème} opérande n'est évalué qu'en cas de besoin :

&&	le 2 ^{ème} opérande n'est évalué que si le 1 ^{er} opérande s'évalue à "vrai",
	le 2 ^{ème} opérande n'est évalué que si le 1 ^{er} opérande s'évalue à "faux".

Jean DEMARTINI © 14.02.00

Fonction hex() & pack()

Conception d'un site WEB



◆ Fonction hex :

hex (\$texte) rend la valeur entière correspondant à la chaîne \$texte supposée représenter un nombre exprimé en notation hexadécimale.

hex comme ci-dessus sur la variable \$_.

◆ Fonction pack :

pack("C", <expression à valeur entière>) rend la chaîne contenant les caractères dont le code ASCII est la valeur de l'expression.

- ◆ La fonction pack() présente de très nombreuses options toutes plus obscures les unes que les autres ; donc consultez la documentation.

Jean DEMARTINI © 14.02.00

L'option "e" de la substitution

Conception d'un site WEB



- ◆ L'option "e" indique que le motif de remplacement est le résultat de l'évaluation d'une expression Perl.

◆ Exemple :

Option d'évaluation

s/%(..)/pack("C", hex(\$1))/eg

Expression dont l'évaluation produit le motif de substitution

A diagram showing the mapping of labels to parts of a Perl substitution command. The command is 's/%(..)/pack("C", hex(\$1))/eg'. An arrow points from 'Option d'évaluation' to 'eg'. Another arrow points from 'Expression dont l'évaluation produit le motif de substitution' to 'hex(\$1)'.

Jean DEMARTINI © 14.02.00

Finalemment, une fonction bien utile

Conception d'un site WEB



```
sub read_input {
  local ($buffer, @paires, $paire, $nom, $valeur, %FORM);
  $ENV{'REQUEST_METHOD'} =~ tr/a-z/A-Z/;
  if ($ENV{'REQUEST_METHOD'} eq "POST") {
    read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
  } else {
    $buffer = $ENV{'QUERY_STRING'};
  }
  @paires = split(/&/, $buffer);
  foreach $paire (@paires) {
    ($nom, $value) = split(/=/, $paire);
    $valeur =~ tr/+ / /;
    $valeur =~ s/%%(..)/pack("C", hex($1))/eg;
    $FORM{$nom} = $valeur;
  }
  %FORM;
}
```

Jean DEMARTINI © 14.02.00

Quelques considérations

Conception d'un site WEB



- ◆ Il est souvent nécessaire de tester un script CGI hors connexion (économies de téléphone).
- ◆ Nous allons donc voir comment écrire deux petits scripts *Perl* de test pour :
 - une communication selon la méthode GET,
 - une communication selon la méthode POST.
- ◆ Ces deux petits scripts peuvent être utilisés aussi bien dans un contexte Unix que dans un contexte Windows 95 ou NT.

Jean DEMARTINI © 14.02.00

Test de la méthode GET

Conception d'un site WEB



Message à transmettre tel qu'il viendrait d'un formulaire.

```
$message = 'liquide=lait&recipient=bol&...';  
$ENV{'REQUEST_METHOD'} = 'GET';  
$ENV{'QUERY_STRING'} = $message;  
system('perl script.pl');
```

Script à tester

On utilise la commande *system* qui crée un processus fils qui hérite des variables d'environnement.

Jean DEMARTINI © 14.02.00

Test de la méthode POST

Conception d'un site WEB



Message à transmettre tel qu'il viendrait d'un formulaire.

```
$message = 'liquide=lait&recipient=bol&...';  
$longueur = length($message);  
$ENV{'REQUEST_METHOD'} = 'POST';  
$ENV{'CONTENT_LENGTH'} = $longueur;
```

```
open(CGI,'|perl script.pl');  
print CGI $message;  
close CGI;
```

On utilise l'ouverture d'un *pipe* qui permet au script testé d'hériter à la fois des variables d'environnements et de recevoir le message via STDIN.

Script à tester

Jean DEMARTINI © 14.02.00

Un petit exemple de script

Conception d'un site WEB



```
Sub read_input {  
    ...  
}  
  
%formulaire = &read_input();  
  
print "  
liquide      : $formulaire{'liquide'}  
recipient    : $formulaire{'recipient'}  
nom          : $formulaire{'nom'}  
";
```

Jean DEMARTINI © 14.02.00

Rôle des scripts CGI

Conception d'un site WEB



- ◆ **Les scripts CGI interviennent très souvent dans le cadre d'un site Web :**
 - Site de recherche : gestion d'une base de donnée,
 - Site de commerce électronique : parcours du catalogue, construction du "panier", élaboration de la facture, déclenchement du paiement,
 - Dispositifs variés :
 - compteurs d'accès,
 - livre d'or (guestbook),
 - ...
 - Scripts d'analyse du trafic :
 - pays, domaines, machines d'origine,
 - navigateur utilisé,
 - ...

Jean DEMARTINI © 14.02.00

Deux exemples un peu réalistes

Conception d'un site WEB



- ◆ Nous allons en terminer sur notre présentation du langage *Perl* par les deux exemples suivants :
 - petit moteur de recherche : utilisation intensive du traitement des chaînes de caractères de *Perl*,
 - constructeur de "paniers" : utilisation systématique des "magic cookies".
- ◆ Les "magic cookies" ou plus simplement "cookies" est une technique introduite par Netscape (à partir de la version 3.0 de son navigateur) pour pallier au comportement purement combinatoire du protocole HTTP.

Jean DEMARTINI © 14.02.00

Un petit moteur de recherche

Conception d'un site WEB



- ◆ Nous nous contenterons d'une *recherche par mots clé hiérarchisés*, non optimisée, dans un domaine bien ciblé :

Recherche gastronomique
- ◆ Nous n'en verrons que les grandes lignes (les détails seront développés sur machine) :
 - structure du fichier des informations,
 - formulaire de référencement,
 - formulaire d'interrogation,
 - moteur de recherche,
 - présentation des résultats de la recherche.

Jean DEMARTINI © 14.02.00

Le fichier des informations

Conception d'un site WEB



- ◆ Nous nous contenterons d'un fichier unique dans lequel chaque enregistrement correspond à une ligne.
- ◆ Les enregistrements seront rangés dans l'ordre où ils auront été introduits.
- ◆ Ce fichier s'appellera BASE.
- ◆ Cette structure peut suffire pour de petites bases de données, elle conduirait, autrement, à des temps de consultation prohibitifs.

Jean DEMARTINI © 14.02.00

Un enregistrement

Conception d'un site WEB



- ◆ Un enregistrement est constitué de champs-clé et de champs-information séparés par le signe "|" ; ils sont séparés entre eux par le signe ":".
- ◆ Champs clé (dans cet ordre) :
 - ville,
 - spécialité : locales, asiatiques, viandes, poissons,
 - catégorie de prix : 50-100, 100-200, 200-300, >300.
- ◆ Champs information (dans cet ordre) :
 - photo (fichier JPEG),
 - nom, sous titre,
 - adresse,
 - prix1, menu1, prix2, menu2.

Jean DEMARTINI © 14.02.00

Pages associées

Conception d'un site WEB



◆ Deux pages sont nécessaires :

- formulaire d'interrogation pour rechercher le restaurant qui nous convient,
- formulaire de référencement pour permettre à un restaurateur de référencer son restaurant dans la base.

Jean DEMARTINI © 14.02.00

Formulaire d'interrogation

Conception d'un site WEB





Où manger ?

Ce soir,
Demain,
Pendant les vacances.

Où (ville)	Spécialités	Prix de base
<input type="text" value="Lyon"/>	<input checked="" type="checkbox"/> cuisine locale <input type="checkbox"/> cuisine asiatique <input type="checkbox"/> viandes <input type="checkbox"/> poissons	<input type="checkbox"/> de 50 à 100 F <input type="checkbox"/> de 100 à 200 F <input type="checkbox"/> de 200 à 300 F <input type="checkbox"/> plus de 300 F

Jean DEMARTINI © 14.02.00

Formulaire de référencement

Conception d'un site WEB



Faites-vous connaître !

 Restaurant :
Sous-titre :
Adresse :

Où (ville)	Spécialités	Prix de base
<input type="text" value="Lyon"/>	<input checked="" type="radio"/> cuisine locale <input type="radio"/> cuisine exotique <input type="radio"/> viandes <input type="radio"/> poissons	<input type="radio"/> de 50 à 100 F <input type="radio"/> de 100 à 200 F <input type="radio"/> de 200 à 300 F <input type="radio"/> plus de 300 F

Menu 1	Menu 2
Prix : <input type="text" value="4000"/> F Composition : <input text"="" type="text" value="1000"/> F Composition : <input button"="" type="text" value="Retourner"/> <input type="button" value="Créer"/>	

Jean DEMARTINI © 14.02.00

Modèles de Page associées

Conception d'un site WEB



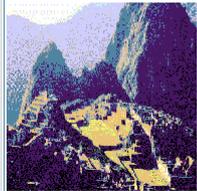
- ◆ **Un modèle de page est nécessaire :**
 - modèle de présentation pour afficher le résultat d'une recherche,
- ◆ **Le modèle de présentation ne constitue pas une page à proprement parler, il sert simplement de modèle à la page que le script d'interrogation doit produire.**

Jean DEMARTINI © 14.02.00

Modèle de présentation

Conception d'un site WEB



	<p>La Boustifaille Spécialités lyonnaises</p> <p>21, rue Mercière 69000 Lyon</p>	<p>Menu 1 : 140 F</p> <ul style="list-style-type: none">• Charcuteries lyonnaises• Pot-au-feu et ses petits légumes• St marcellin aënés• Sorbet aux artichaux chauds	<p>Menu 2 : 180 F</p> <ul style="list-style-type: none">• Salade lyonnaise aux truffes• Haricot de canard aux navets• Epoisse au marc de Bourgogne• Evarois aux pommes
---	---	--	--

Jean DEMARTINI © 14.02.00

Format de l'enregistrement

Conception d'un site WEB



- ◆ L'enregistrement est sur une seule ligne telle que :

```
ville:specialite:categorie|  
photo:nom:soustitre:adresse:prix1:menu1:prix2:menu2
```

Le champ *photo* correspond au nom du fichier JPEG, la photo sera parvenue par un autre canal que celui du formulaire (La poste par exemple).

Jean DEMARTINI © 14.02.00

Les repères des formulaires

Conception d'un site WEB



◆ Les différents champs de l'enregistrement correspondent aux entités de formulaire ayant les noms suivants (sauf photo) :

- ville : ville - clé n°1
- specialite : spécialité - clé n°2
- categorie : catégorie - clé n°3
- nom : nom du restaurant - information
- soustitre : complément de nom - information
- adresse : adresse - information
- prix1 : prix du menu n°1 - information
- menu1 : menu n°1 - information
- prix2 : prix du menu n°2 - information
- menu2 : menu n°2 - information

Jean DEMARTINI © 14.02.00

Les repères des scripts

Conception d'un site WEB



◆ Les différents champs de l'enregistrement correspondent aux entités de présentation suivantes :

- ville : \$ville
- specialite : \$specialite
- categorie : \$categorie
- photo : \$photo
- nom : \$nom
- soustitre : \$soustitre
- adresse : \$adresse
- prix1 : \$prix1
- menu1 : @menu1
- prix2 : \$prix2
- menu2 : @menu2

Jean DEMARTINI © 14.02.00

Script d'interrogation : structure

Conception d'un site WEB



- ◆ Le script d'interrogation comporte trois parties décrites par trois fonctions :

```
%formulaire = &read_input();
```

construction du tableau associatif correspondant au formulaire d'interrogation.

```
@resultats = &interrogation($cle1,$cle2,$cle3);
```

interrogation du fichier des données et construction du tableau de la partie information des enregistrements trouvés.

```
do page(@resultats);
```

construction de la page de présentation.

Jean DEMARTINI © 14.02.00

Script d'interrogation : source

Conception d'un site WEB



```
#!/usr/local/bin/perl

sub read_input {...}
sub interrogation {...}
sub page {...}
sub compose_motif {...}
sub compose_menu {...}

%formulaire = &read_input();
$cle1 = $formulaire{'ville'};
$cle2 = $formulaire{'specialite'};
$cle3 = $formulaire{'categorie'};
@resultats = &interrogation($cle1, $cle2, $cle3);
do page(@resultats);
```

Jean DEMARTINI © 14.02.00

Fonction d'interrogation

Conception d'un site WEB



```
sub interrogation {
  local ($cle1, $cle2, $cle3) = @_;
  open(BASE, "<base.rst") || die 'fichier inconnu';
  foreach $enregistrement (<BASE>) {
    ($cles, $info) = split(/\|/, $enregistrement);
    ($lc1, $lc2, $lc3) = split(/:/, $cles);
    if (($cle1 eq $lc1) &&
        ($cle2 eq $lc2) && ($cle3 eq $lc3)) {
      push(@RESULTATS, $info);
    }
  }
  close BASE;
  @RESULTATS;
}
```

Jean DEMARTINI © 14.02.00

Composition de la page

Conception d'un site WEB



	\$nom	\$prix1	\$prix2
\$photo	\$soustitre	\$menu1	\$menu2
	\$adresse		

- ◆ La présentation de la page est obtenue en combinant des motifs construits à l'aide de 3 tables.
- ◆ Il y a autant de motifs que d'éléments dans le tableaux @resultats.

Jean DEMARTINI © 14.02.00

Fonction de composition de la page

Conception d'un site WEB



```
sub page {
  print "
Content-type: text/html

<HTML>
<HEAD><TITLE>R&eacute;sultats</TITLE></HEAD>
<BODY BGCOLOR="White">";
  foreach $resultat (@_) {
    print "
<TABLE WIDTH="100%" BORDER="2"
        CELLSPACING="2" CELLPADDING="2">";
    do compose_motif($resultat);
    print "</TABLE>";
  }
  print "</BODY></HTML>";
}
```

Jean DEMARTINI © 14.02.00

Fonction de composition du motif

Conception d'un site WEB



```
sub compose_motif {
  local ($photo,$nom,$soustitre,$adresse,
        $prix1,$menu1,$prix2,$menu2)
    = split(/:/, @_ [0]);
  print "<TR>
<TD><IMG SRC=" $photo "></TD>
<TD ALIGN="CENTER">
  <FONT SIZE="+3"><B>$nom</B></FONT><BR>
  <B>$soustitre</B><HR>$adresse</TD>
<TD ALIGN="CENTER" VALIGN="TOP">
  <B>Menu 1 : $prix1 F</B><HR> ";
  do compose_menu(split(/\./,$menu1));
  print "</TD>";
  ....
  print "</TR>";
}
```

Jean DEMARTINI © 14.02.00

Fonction de composition des menus

Conception d'un site WEB



```
sub compose_menu {
  print "<TABLE>";
  foreach $item (@_) {
    print " <TR>
          <TD VALIGN=\"TOP\">
            <IMG SRC=\"blueball.gif\"></TD>
          <TD VALIGN=\"TOP\">$item</TD>
        </TR>
      ";
  }
  print "</TABLE>";
}
```

Jean DEMARTINI © 14.02.00

Script de référencement : source

Conception d'un site WEB



```
#!/usr/local/bin/perl

sub read_input {...}
sub référencement {...}

do référencement(%formulaire);
```

- ◆ Cette version est minimale, on peut, bien sûr, la compléter :
 - On pourrait, par exemple, renvoyer une page analogue à la page de présentation,
 - ...

Jean DEMARTINI © 14.02.00

Fonction de référencement

Conception d'un site WEB



```
sub referencement {
  local ($ville, $specialite, $categorie, ...) =
    (@_[0]){ville}, @_[0]){specialite}, ...);
  open(BASE,">>base.rst") || die 'fichier inconnu';
  print BASE "$ville:$specialite: ... |$photo: ... \n";
  close BASE;
}
```

- ◆ Voici une version élémentaire de la fonction de référencement.
- ◆ Il reste quelques petits problèmes à résoudre :
 - traitement des caractères spéciaux,
 - permettre le déréférencement,
 - éviter les déréférencements "sauvages" ...

Jean DEMARTINI © 14.02.00

Le "panier de la ménagère"

Conception d'un site WEB



- ◆ Lorsqu'on navigue sur un site, en soumettant des formulaires à diverses reprises, le serveur ne peut pas conserver chez lui la trace de ce qui a été saisi ; *il n'a aucun moyen d'identifier qui l'appelle car les adresses IP sont presque toujours flottantes.*
- ◆ Pour ce faire, le serveur peut :
 - Provoquer l'écriture d'une information de son choix dans un fichier de la machine client.
 - Relire l'information qu'il a fait écrire.
- ◆ Ces portions d'informations s'appellent des "magic cookies".

Jean DEMARTINI © 14.02.00

Maintenir un "état"

Conception d'un site WEB



- ◆ L'utilisation de cookies n'est pas la seule technique possible.
- ◆ On peut, également, utiliser les champs "cachés" des formulaires :
`<INPUT TYPE="HIDDEN" NAME="..." VALUE="...">`
- ◆ Ces champs ne sont pas affichés, ils permettent au serveur d'inclure dans un formulaire une information qui lui sera retransmise.
- ◆ On peut ainsi créer des formulaires "à tiroirs".

Jean DEMARTINI © 14.02.00

Mise en place d'un cookie

Conception d'un site WEB



- ◆ Un cookie est une portion de l'en-tête du document renvoyé par le serveur (script CGI) et reçu par le navigateur.
- ◆ Le cookie est rangé dans un fichier cookies.txt par le navigateur Netscape, dans un répertoire Cookies par Internet Explorer.

```
Content-type: text/html
Set-Cookie: <Nom>=<Valeur>; expires=<Date>;
            path=<Chemin>; domain=<Domaine>;
            secure

<HTML> ... </HTML>
```

Jean DEMARTINI © 14.02.00

Paramètres des cookies (1)

Conception d'un site WEB



◆ <Nom>=<Valeur>

obligatoire — chaînes de caractères (sans point-virgule, virgule ou espace qui peuvent être URL-encodés : %..) définissant le nom et la valeur du cookie.

compte=1

◆ expires=<Date>

facultatif — indique la date à partir de laquelle le cookie doit être détruit.

Si ce paramètre n'est pas fourni, le cookie est détruit *au moment où on quitte le navigateur.*

expires=Tuesday,19-May-1998 18:35:40 GMT

Jean DEMARTINI © 14.02.00

Paramètres des cookies (2)

Conception d'un site WEB



◆ domain=<Domaine>

facultatif — nom du domaine de la machine vers lequel le cookie peut être renvoyé. Ce nom doit comporter au moins 2 points ".", La comparaison est effectuée par la fin.

domain=.unice.fr

Ce nom de domaine correspond aux machines verdon.unice.fr, hermes.unice.fr, i3S.unice.fr ...

Si ce paramètre n'est pas fourni, le cookie ne peut être renvoyé *qu'à la machine* qui l'a mis en place.

Jean DEMARTINI © 14.02.00

Paramètres des cookies (3)

Conception d'un site WEB



◆ path=<Chemin>

facultatif — tête du chemin qui doit être indiqué dans l'URL requise pour déclencher l'envoi du cookie.

path=/~jdem

Cette tête de chemin correspond au répertoire ~jdem ainsi qu'à tous ses sous-répertoires.

Si ce paramètre n'est pas fourni, le cookie ne peut être renvoyé *qu'en suivant exactement le même chemin* que celui de la requête à l'origine de l'installation du cookie.

◆ secure

facultatif — le cookie ne peut être transmis qu'à un serveur *sécurisé*.

Jean DEMARTINI © 14.02.00

Renvoi d'un cookie

Conception d'un site WEB



- ◆ Lorsqu'une URL est requise, le navigateur recherche tous les cookies concernés et renvoi vers le serveur un message de la forme :

Cookie: <Nom1>=<Valeur1>; <Nom2>=<Valeur2>; ...

- ◆ Cette chaîne de caractères (sans "Cookie:") est accessible dans la variable d'environnement HTTP_COOKIE
- ◆ Le traitement de HTTP_COOKIE est analogue à celui de QUERY_STRING (cf. diapo 149).

Jean DEMARTINI © 14.02.00

De plus ...



Conception d'un site WEB

- ◆ On peut envoyer plusieurs cookie dans le même en-tête.
- ◆ Un client peut toujours supprimer un cookie avant que le navigateur ne le détruise en éditant le fichier cookies.txt.
- ◆ Pour supprimer un cookie, il suffit de renvoyer un cookie de même définition qui *expire dans le passé*.
- ◆ Les cookies traversent les proxy-serveurs.
- ◆ Limitations :
 - pas plus de 300 cookies,
 - pas plus de 4Ko par cookie,
 - pas plus de 20 cookies par serveur ou domaine.

Jean DEMARTINI © 14.02.00

Exemple de transaction



Conception d'un site WEB

Client ← Serveur
Set-cookie: client=Donald; path=/; expires=Monday,25-May-1998 23:59:59 GMT

Client → Serveur
Cookie: client=Donald

Client ← Serveur
Set-cookie: legumes=carottes; path=/

Client → Serveur
Cookie: client=Donald; legumes=carottes

Client ← Serveur
Set-cookie: laitages=yaourts; path=/

Client → Serveur
Cookie: client=Donald; legumes=carottes; laitages=yaourts

Jean DEMARTINI © 14.02.00

Quelques références



Conception d'un site WEB

- ◆ **La définition des cookies est en pleine évolution. Pour plus d'informations, on peut consulter :**

http://home.netscape.com/newsref/std/cookie_spec.html

bonne introduction (par les inventeurs) à l'utilisation des cookies.

<http://www.illuminatus.com/cookie.fcgi>

bonne réflexion sur les risques encourus (très peu) et l'utilisation des cookies.

<http://www.gh-interactive.com/WEBMASTERREPORT2.HTML>

magazine à parution (relativement) périodique traitant de différents sujets concernant l'administration d'un serveur Web.

Jean DEMARTINI © 14.02.00

A présent, changeons de côté



Conception d'un site WEB

- ◆ **Faisons le point :**

	Côté client	Côté serveur
Pages statiques	Langage HTML	
Pages dynamiques	Scripts Javascript*	Scripts CGI Perl

* ... et Applets Java (ou Active-X). Nous ne présenterons que le langage Javascript beaucoup plus utilisé (et admis pour cet usage) que le langage Java.

Jean DEMARTINI © 14.02.00

Javascript



Conception d'un site WEB

- ◆ **JavaScript** est une extension de HTML qui permet d'inclure, dans une page HTML, des fragments de programmes interprétés par le navigateur.
- ◆ **JavaScript** n'est pas du *Java* simplifié. C'est un langage de programmation à part entière, possédant des possibilités très modernes, normalisé sous la référence ECMA-262.
- ◆ Quelques références :
 - <http://www.ecma.ch/>
 - <http://www.dpunkt.de/javascript/>
 - <http://developer.netscape.com/>
 - <http://www.imaginet.fr/ime/javascr.htm>

Jean DEMARTINI © 14.02.00

Mon premier script



Conception d'un site WEB

```
<HTML>
<HEAD><TITLE>JavaScript - le&ccedil;on
1</TITLE></HEAD>
<BODY BGCOLOR="White">
  <P>Texte HTML normal.<BR>
  <SCRIPT LANGUAGE="JavaScript">
    document.write("Ceci provient d'un script JavaScript.")
  </SCRIPT><BR>
  Et voici, &agrave; nouveau, du texte HTML.</P>
</BODY></HTML>
```

- ◆ On remarque une nouvelle balise-conteneur qui peut être placée n'importe où dans la description d'une page HTML :
`<SCRIPT paramètres>...</SCRIPT>`

Jean DEMARTINI © 14.02.00

Une petite précaution

Conception d'un site WEB



```
...  
<SCRIPT LANGUAGE="JavaScript">  
<!-- caché aux vieux navigateurs.  
      document.write("Ceci provient d'un script JavaScript.")  
!-->  
</SCRIPT>  
...
```

- ◆ Les navigateurs antérieurs à Netscape 2.0 et Internet Explorer 3.0 ne reconnaissent pas *Javascript*. Il est donc prudent d'en cacher les instructions en simulant des commentaires.

Jean DEMARTINI © 14.02.00

Les événements

Conception d'un site WEB

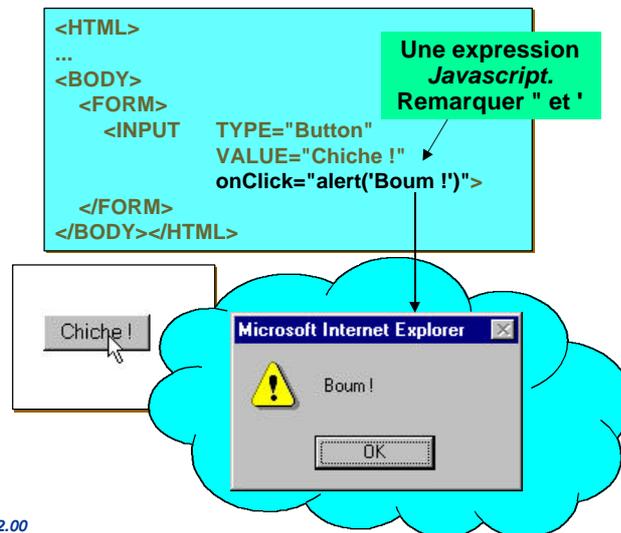


- ◆ *Evénements* et *gestionnaires d'événements* sont les 2 mamelles de la programmation *Javascript*.
- ◆ La programmation *Javascript* est donc essentiellement *réactive*.
- ◆ Un événement est une situation fugitive qui est apparaît dans certaines conditions :
 - onLoad on vient de charger une nouvelle page,
 - onClick on vient de cliquer sur un bouton,
 - onMouseOver la souris vient d'entrer dans la zone d'un lien,
 - onMouseOut la souris vient de sortir d'une zone de lien,
 - ...

Jean DEMARTINI © 14.02.00

Une première réaction

Conception d'un site WEB



Jean DEMARTINI © 14.02.00

"Objets" Javascript

Conception d'un site WEB



- ◆ Le langage *Javascript* est construit autour d'un modèle-objet simple.
- ◆ Toutes les entités que permettent de manipuler les instructions de ce langage sont des atomes ou des objets caractérisés par des attributs :
 - certains définissent ses *propriétés* — qui sont d'autres objets ou atomes,
 - d'autres définissent ce que l'on peut lui faire : ce sont ses *méthodes*.
- ◆ Nous verrons que les objets peuvent être définis et créés, par contre, il n'existe aucun mécanisme d'héritage.

Jean DEMARTINI © 14.02.00

Entités prédéfinies

Conception d'un site WEB



- ◆ **Javascript prédéfinit les types d'atomes suivants :**
 - nombres : 3 ou 3.14 ou 0.031e2
 - booléens : true ou false
 - chaînes : "coucou" ou 'coucou'
 - fermetures : fonction et son environnement de définition
- ◆ **Le modèle objet de Javascript est construit autour de la structure de dictionnaire symbolique (tableau associatif) dont la seule constante est this. Ce dictionnaire symbolique est instanciable et les objets Javascript seront tous des instances de this.**

Jean DEMARTINI © 14.02.00

Accès aux attributs et aux méthodes

Conception d'un site WEB



- ◆ **Un objet étant un dictionnaire, l'accès aux attributs (propriétés ou méthodes) peut donc naturellement être :**
 - `Math["sin"]` fonction sinus
 - `Math["PI"]` valeur du nombre Pi
- ◆ **Cette écriture étant assez lourde, on lui a associé un sucre syntaxique (opérateur "." qui s'associe à gauche) :**
 - `Math.PI` `Math.sin`
- ◆ **Exemple :**
 - `Math.sin(1.5)` calcule le sinus d'un angle de 1.5 radians

Jean DEMARTINI © 14.02.00

Une page HTML

Conception d'un site WEB

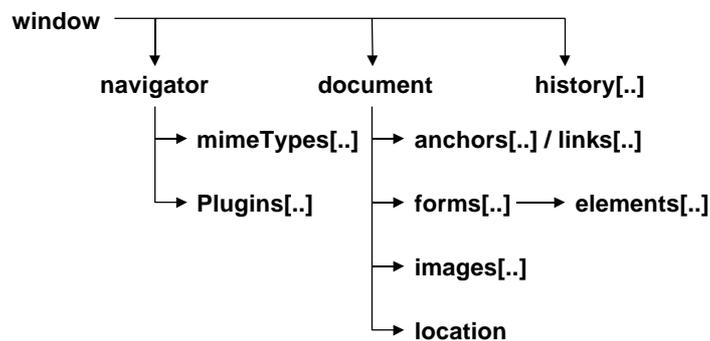


- ◆ Lorsqu'une fenêtre (contenant une page HTML) est construite par le navigateur, une hiérarchie d'objets Javascript est construite parallèlement.
- ◆ Les scripts Javascript pourront accéder et manipuler les éléments de cette fenêtre (donc de la page HTML) à travers cette hiérarchie.
- ◆ Seront manipulables (entre autres) :
 - les ancres ou liens,
 - les images,
 - les formulaires.

Jean DEMARTINI © 14.02.00

Hiérarchie des objets Javascript

Conception d'un site WEB



Jean DEMARTINI © 14.02.00

Composantes d'une page (1)

Conception d'un site WEB



- ◆ La hiérarchie des objets associés à l'objet document est construite au fur et à mesure que le texte HTML est interprété.
- ◆ Ainsi :
 - Les images seront repérées par :
document.images[0]
document.images[1]
...
 - Les liens seront repérés par :
document.links[0]
document.links[1]
...
 - etc...

Jean DEMARTINI © 14.02.00

Composantes d'une page (2)

Conception d'un site WEB



- ◆ La hiérarchie des objets associés à l'objet document est plus facile à parcourir si on a nommé ses éléments.
- ◆ Ainsi :
 - L' image définie par :
–
sera repérée par : document.Chiche
 - Le lien défini par :
– ...
sera repéré par : document.lesson_1
 - etc...

Jean DEMARTINI © 14.02.00

Un exemple plus conséquent

Conception d'un site WEB



```
<FORM NAME="somme">
  <INPUT TYPE="Text" NAME="x" VALUE="0"><BR> +
  <INPUT TYPE="Text" NAME="y" VALUE="0"><BR>
  <INPUT TYPE="Button"
    VALUE=" = "
    onClick=
      "document.somme.z.value =
        String(parseInt(document.somme.x.value) +
          parseInt(document.somme.y.value))">
  <INPUT TYPE="Text" NAME="z" VALUE="0">
</FORM>
```

123
+ 45
= 168

Jean DEMARTINI © 14.02.00

Structure d'un programme

Conception d'un site WEB



- ◆ A la base, on a des *atomes*, des *objets*, des *variables* et des *opérateurs*.
- ◆ Une variable est, a priori, un *nom* utilisé pour désigner la valeur d'un atome ou d'un objet.
- ◆ Les atomes, les objets, les variables, les opérateurs et les *expressions* peuvent être regroupés pour former des *expressions*.
- ◆ Un *programme* est une *expression*.

Et c'est tout !

- ◆ Nous avons déjà vus certains atomes et certains objets, voyons les opérateurs.

Jean DEMARTINI © 14.02.00

Nom de variable



Conception d'un site WEB

- ◆ Un nom de variable est un symbole uniquement conçu pour ne pas être confondu avec autre chose (un nombre par exemple).
- ◆ En *Javascript*, un nom de variable est formé de lettres et de chiffres et commence par une lettre.
 - Le signe `_` est considéré comme une lettre.
- ◆ On distingue les lettres majuscules des lettres minuscules.

`<nom> ::= [a-zA-Z_] [0-9a-zA-Z_]*`

Jean DEMARTINI © 14.02.00

Opérateurs



Conception d'un site WEB

- ◆ A la base, l'opérateur représente une fonction au sens mathématique.

+	addition
-	différence
*	produit
/	division
- ◆ Certains opérateurs ne produisent que des effets :

<code>var .. = ..</code>	définition d'une variable
<code>function .. (..) ..</code>	définition d'une fermeture
<code>write</code>	écriture — cf. <code>document.write(...)</code>
- ◆ Un opérateur est appliqué à ses arguments lors de l'évaluation de l'expression qui le contient.
- ◆ Une fermeture est un opérateur défini par le programmeur.

Jean DEMARTINI © 14.02.00

Deux stratégies d'évaluation

Conception d'un site WEB



Soit $f(x) = 1 + x.(2 + 3.x)$
et $g(y) = y.y$

Que vaut $f(g(5))$?

$$\begin{aligned} f(g(5)) &= 1 + g(5).(2 + 3.g(5)) \\ &= 1 + 5.5.(2 + 3.5.5) \\ &= 1926 \end{aligned}$$

Evaluation en
ordre normal
les arguments sont
passés non évalués

$$\begin{aligned} f(g(5)) &= f(5.5) \\ &= f(25) \\ &= 1 + 25.(2 + 3.25) \\ &= 1926 \end{aligned}$$

Evaluation en
ordre applicatif
les arguments sont
passés évalués

Jean DEMARTINI © 14.02.00

Laquelle choisir ?

Conception d'un site WEB



- ◆ Les théoriciens (théorèmes de Church-Rosser) ont montré que l'évaluation en ordre normal aboutit toujours.
- ◆ Mais l'évaluation en ordre applicatif est plus rapide.
- ◆ Alors :
 - on évalue en ordre applicatif les expressions qui le peuvent : *formes standards*,
 - on évalue en ordre normal les expressions* que ne peuvent pas être évaluées en ordre applicatif : *formes spéciales*.

* ces expressions sont en petit nombre et bien connues. Heureusement !

Jean DEMARTINI © 14.02.00

Expressions vs Instructions

Conception d'un site WEB



- ◆ Certains opérateurs produisent essentiellement un "effet". Par exemple :
 - read(..)
 - write(..)
- ◆ Lorsqu'une expression ne produit qu'un effet ou lorsque sa valeur n'est pas utilisée, on l'appelle, en général, une *instruction*.
- ◆ Les instructions ne sont réellement indispensables que lorsqu'il s'agit de décrire un *comportement d'interface*. Il peut, cependant, être commode d'en utiliser dans d'autres contextes.

Jean DEMARTINI © 14.02.00

Séquences

Conception d'un site WEB



- ◆ En ordre applicatif, l'ordre dans lequel on évalue les arguments d'un opérateur n'a aucune importance.
- ◆ Par contre, l'enchaînement des effets produits va dépendre de l'ordre dans lequel les instructions sont évaluées.
- ◆ Il existe donc un opérateur qui garantit l'ordre dans lequel on évalue ses arguments : la séquence.
 - expression-séquence : (<expression>, <expression>...)
 - instruction-séquence : {<instruction>; <instruction>...}
- ◆ La valeur d'une expression-séquence est celle de la dernière expression évaluée.

Jean DEMARTINI © 14.02.00

Opérateurs standards

Conception d'un site WEB



- ◆ Un opérateur standard est tel que ses arguments peuvent être évalués dans n'importe quel ordre sans que son comportement en soit affecté.

Opérateurs "bit à bit"		Opérateurs arithmétiques	
&	"et" bit à bit	+	Addition (concaténation)
	"ou" bit à bit	-	soustraction
^	"ou-exclusif" bit à bit	*	multiplication
<<	décalage à gauche	/	division
>>	décalage à droite	%	reste
>>>	décalage à droite		

Opérateurs de relation			
==	égaux	!=	différents
<	inférieur	>	supérieur
<=	inférieur ou égal	>=	supérieur ou égal

Jean DEMARTINI © 14.02.00

Expressions standards

Conception d'un site WEB



- ◆ La syntaxe et la sémantique des expressions standards de Javascript est identique* à celles utilisées par le langage C.

- ◆ Exemples :

3	x
3 + 4	3 * 4 + 5
(x < y) (x < z)	(n == 0)
(3 + x) * (5 + z)	factorielle(6)

- ◆ La priorité des opérateurs est la même et les parenthèses sont utilisées de la même façon.

* pour ce que j'ai pu constater jusqu'à présent.

Jean DEMARTINI © 14.02.00

Expressions spéciales (1)

Conception d'un site WEB



- ◆ La syntaxe et la sémantique des expressions spéciales de Javascript est *presque* identique à celles utilisées par le langage C.
- ◆ Expressions conditionnelles :

```
<prédicat> ? <conséquence> : <alternative>  
if (<prédicat>) {<conséquence>} else { <alternative>}  
if (<prédicat>) {<conséquence>}
```
- ◆ Opérateurs d'itération :

```
while (<prédicat>) {...}  
do {...} while (<prédicat>) {...}  
for (<expr-i>; <expr-f>; <expression>) {...}
```

Jean DEMARTINI © 14.02.00

Expressions spéciales (2)

Conception d'un site WEB



- ◆ Définir une variable, c'est lier un nom à une valeur.
- ◆ Définition d'une variable :

```
<nom> = <expression>
```
- ◆ Lorsqu'un nom qui n'a été lié à aucune valeur apparaît dans une expression, on dit que c'est une variable libre. Sinon, la variable est dite liée.
- ◆ Une expression contenant des variables libres n'a pas de valeur, on dit que sa valeur est undefined (la non-valeur).
- ◆ Une expression ne contenant aucune variable libre est dite fermée.

Jean DEMARTINI © 14.02.00

Lieurs de variables



Conception d'un site WEB

- ◆ Le signe = est un lieu de variable :
l'expression (x = 2, x+5) vaut 7
- ◆ Lorsqu'une variable est liée, on peut la renommer librement (presque).
- ◆ La définition d'une fonction est une autre manière de lier des variables :
function <nom> (<nom>, <nom>...) {...}
Paramètres : noms liés dans la séquence
- ◆ L'opérateur fonction construit en fait une *fermeture*.

Jean DEMARTINI © 14.02.00

Opérateurs "paresseux"



Conception d'un site WEB

- ◆ Certains opérateurs pourraient très bien être considérés comme standards.
- ◆ Pourtant, il est commode (plus efficace) de les considérer comme spéciaux car tous leurs arguments ne sont pas nécessaires à leur évaluation.
- ◆ On peut procéder ainsi pour toutes les fonctions incomplètes.

Opérateurs booléens	
&&	conjonction
	disjonction
!	complémentation

Jean DEMARTINI © 14.02.00

Objet statiques ...



Conception d'un site WEB

- ◆ Pour représenter certaines situations réelles, la variable définie *statiquement* — lien nom-valeur éternel — comme nous venons de le voir n'est pas très commode.
- ◆ On acceptera (avec réticence) de perdre les possibilités du *raisonnement* au profit des possibilités de la *description*.

Ce qui revient à avoir l'attitude un peu illogique * suivante : décrivons déjà, on verra bien après ce qu'on peut en faire.

* souvenez-vous de proverbe "Shaddock" : il n'est pas nécessaire de regarder où on va, il sera bien temps de voir quand on y sera !

Jean DEMARTINI © 14.02.00

... ou objets vivants ?



Conception d'un site WEB

- ◆ On admettra alors les deux opérations complémentaires suivantes :
 - attacher un nom à une valeur : *définir* une variable,
 - attacher une valeur à un nom : *affecter* une variable.
- ◆ Le lien nom-valeur n'est plus éternel et la variable peut avoir une *histoire*.
 - $x = 3$ *définition* de la variable x
 - $x = 4$ *affectation* de la variable x qui, à présent, vaut 4
 - $x = x + 2$ *ré-affectation* de la variable x
- ◆ Ce qui est à droite d'un signe = fait référence au *passé* tandis que ce qui est à gauche représente le *futur*.

Jean DEMARTINI © 14.02.00

Opérateurs d'affectation

Conception d'un site WEB



- ◆ Ces opérateurs sont, en fait, des sucres syntaxiques :

$x += 3$ au lieu de $x = x + 3$
 $x -= 5$ au lieu de $x = x - 5$

- ◆ La syntaxe générale des expressions (instructions ?) utilisant ces opérateurs est :

$x <op>= <expression>$ au lieu de $x = x <op> <expression>$

- ◆ Une expression d'affectation a une valeur, l'expression suivante a donc un sens :

$x = y = z = 3$ l'opérateur = s'associe à droite.

Jean DEMARTINI © 14.02.00

Opérateurs d'inc(dec)rémentation

Conception d'un site WEB



- ◆ Certains opérateurs décrivent une affectation implicite :

$++$ incrémentation
 $--$ décrémentation

- ◆ L'incrément (décrément) peut être effectuée avant ou après l'évaluation de l'expression contenant l'opérateur.

- ◆ Préincrément (prédécrément) :

$(x++ + 2)$ on évalue $x+2$ puis x est incrémentée
 $(++x + 2)$ on incrémente x puis on évalue $x + 2$

Jean DEMARTINI © 14.02.00

Les objets *Javascript*

Conception d'un site WEB



- ◆ Un objet *javascript* est décrit par sa fonction-constructeur qui configure l'objet universel *this* en lui définissant des attributs et des méthodes.
- ◆ Exemple — un nombre rationnel possède 2 attributs, son numérateur et son dénominateur :

```
function Rationnel(n,d) {  
    this.numerateur = n;  
    this.denominateur = d;  
}
```

Jean DEMARTINI © 14.02.00

Instanciation d'un objet

Conception d'un site WEB



- ◆ L'instanciation d'un objet est effectuée par l'opérateur standard *new*.
- ◆ Instancier un objet, c'est en fabriquer un nouvel exemplaire à partir de sa description.
- ◆ Exemple :

```
q12 = new Rationnel(1,2)  
q34 = new Rationnel(3,4)
```

- ◆ Accès aux attributs :

```
q12.numerateur    vaut    1  
q12.denominateur  vaut    2
```

Jean DEMARTINI © 14.02.00

Les méthodes d'un objet (1)

Conception d'un site WEB



- ◆ En définissant la fermeture ad'hoc, on peut associer une méthode d'addition aux nombres rationnels.

```
function rat_somme(q) {  
  n1 = this.numerateur;  
  d1 = this.denominateur;  
  n2 = q.numerateur;  
  d2 = q.denominateur;  
  return new Rationnel(n1*d2+n2*d1,d1*d2);  
}
```

- ◆ On remarque l'utilisation de l'objet this qui représente l'objet qui a reçu le message.

Jean DEMARTINI © 14.02.00

Les méthodes d'un objet (2)

Conception d'un site WEB



- ◆ Il ne reste plus qu'à rajouter la méthode aux propriétés de l'objet.

```
function Rationnel(n,d) {  
  this.somme = rat_somme;  
  this.numerateur = n;  
  this.denominateur = d;  
}
```

- ◆ On peut alors créer des nombres rationnels et effectuer des opérations à leur sujet :

```
q12 = new Rationnel(1,2);  
q34 = new Rationnel(3,4);  
qs = q12.somme(q34);
```

↑
On "envoie le message
somme" à l'objet q12.

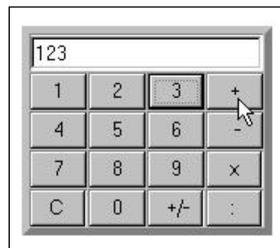
Jean DEMARTINI © 14.02.00

Un exemple plus conséquent

Conception d'un site WEB



- ◆ L'objet de cet exemple est d'introduire une calculette dans une page Web.
- ◆ Une calculette un peu simplifiée, utilisant une notation polonaise post-fixée, dont l'allure est la suivante :



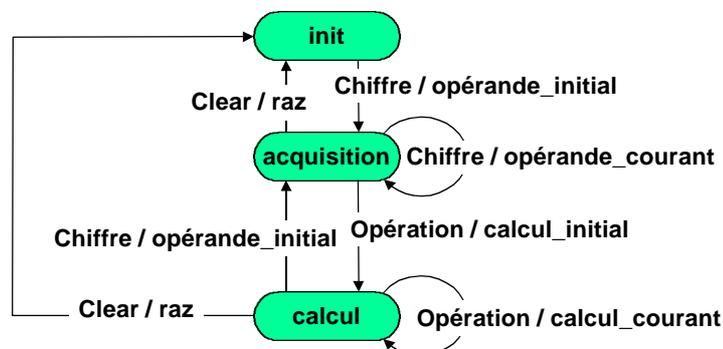
Jean DEMARTINI © 14.02.00

Description de son comportement

Conception d'un site WEB



- ◆ C'est une machine à états finis qui permet de le décrire le plus facilement :



Jean DEMARTINI © 14.02.00

L'automate



Conception d'un site WEB

◆ L'automate qui décrit la machine à états finis comporte :

- 3 états : "init", "acquisition" et "calcul"
- 3 événements : "Clear", "Chiffre" et "Opération"
- 5 actions : "raz", "opérande_initial", "opérande_courant", "calcul_initial", "calcul_courant"

◆ L'interface utilisateur est construite autour d'un formulaire nommé "calcullette" qui comporte :

- des boutons pour engendrer les événements,
- une zone de texte nommée "n" pour réaliser les affichages de l'opérande (en mode "acquisition") et du résultat (en mode "calcul").

Jean DEMARTINI © 14.02.00

Le formulaire "calcullette"



Conception d'un site WEB

```
<FORM NAME="calcullette">
<TABLE BORDER="5">
  <TR>
    <TD COLSPAN="4" ROWSPAN="1">
      <INPUT TYPE="Text" NAME="n" VALUE="0"></TD></TR>
  <TR>
    <TD><INPUT TYPE="Button" ... onClick="transition('chiffre','1')"></TD>
    <TD><INPUT TYPE="Button" ... onClick="transition('chiffre','2')"></TD>
    <TD><INPUT TYPE="Button" ... onClick="transition('chiffre','3')"></TD>
    <TD><INPUT TYPE="Button" ...
onClick="transition('operation','add')"></TD>
  </TR>
  <TR>...</TR>
  <TR>...</TR>
  <TR>
    <TD><INPUT TYPE="Button" ... onClick="raz()"></TD>
    <TD><INPUT TYPE="Button" ... onClick="transition('chiffre','0')"></TD>
    <TD><INPUT TYPE="Button" ...
onClick="transition('operation','neg')"></TD>
    <TD><INPUT TYPE="Button" ... onClick="transition('operation','div')"></TD>
  </TR>
</TABLE></FORM>
```

Jean DEMARTINI © 14.02.00

L'état de la calculette

Conception d'un site WEB



◆ L'état de la calculette peut être représenté par 4 variables :

etat valant "init", "acquisition" ou "calcul",
operande valant la valeur numérique de l'opérande en cours d'utilisation,
resultat valant le résultat de l'accumulation de tous les calculs effectués depuis le dernier "Clear".
document.calculette.n.value
- chaîne des chiffres de l'opérande en mode
"acquisition",
- chaîne des chiffres du résultat en mode
"calcul".

Jean DEMARTINI © 14.02.00

La fonction de transition

Conception d'un site WEB



```
function transition (e,v) {  
  if (etat == "init") {  
    if (e == "chiffre") {operande_initial(v);etat = "acquisition"}  
    return;  
  }  
  if (etat == "acquisition") {  
    if (e == "chiffre") {operande_courant(v)}  
    if (e == "operation") {  
      operande = parseInt(document.calculette.n.value);  
      calcul_initial(v);  
      etat = "calcul";}  
    return;  
  }  
  if (etat == "calcul") {  
    if (e == "chiffre") {operande_initial(v);etat = "acquisition"}  
    if (e == "operation") {calcul_courant(v)}  
    return;  
  }  
}
```

Jean DEMARTINI © 14.02.00

L'action "raz"

Conception d'un site WEB



```
function raz () {  
  resultat = 0;  
  operande = 0;  
  etat = "init";  
  document.calculette.n.value = "0";  
}
```

Jean DEMARTINI © 14.02.00

Les actions "operande_..."

Conception d'un site WEB



```
function operande_initial (v) {  
  document.calculette.n.value = v;  
}  
  
function operande_courant (v) {  
  if (document.calculette.n.value == "0") { ←  
    document.calculette.n.value = v;  
  } else {  
    document.calculette.n.value += v;  
  }  
}
```

Pour ne pas avoir de "0" à gauche

Jean DEMARTINI © 14.02.00

Les actions "calcul_..."

Conception d'un site WEB



```
function calcul (v) {
  if (v == "add")    {resultat += operande;}
  if (v == "sub")    {resultat -= operande;}
  if (v == "mul")    {resultat *= operande;}
  if (v == "div")    {resultat /= operande;}
  document.calculette.n.value = resultat;
}
function calcul_initial (v) {
  if (v == "neg")    {resultat = -operande;}
  calcul(v);
}
function calcul_courant (v) {
  if (v == "neg")    {resultat = -resultat;}
  calcul(v);
}
```

Jean DEMARTINI © 14.02.00

Les objets Javascript d'usage courant

Conception d'un site WEB



- ◆ Les objets Javascript constituent une *hiérarchie* documentée dans tous les bons outils de développement.
- ◆ La référence Javascript peut être consultée à home.netscape.com/eng/mozilla/3.0/handbook/javascript
- ◆ Attention, Javascript est en pleine évolution, ce site est à consulter fréquemment.

Jean DEMARTINI © 14.02.00

En guise de conclusion

Conception d'un site WEB



- ◆ **Le but de ce cours était :**
 - de vous donner quelques éléments de base pour apprendre à monter un site Web,
 - de vous donner des pistes pour suivre les évolutions qui se dessinent dans le monde internet.
- ◆ **Ce cours est donc largement incomplet.**
- ◆ **Vous avez cependant constaté qu'il était déjà possible de construire des applications intéressantes avec peu de moyens.**