

Programmation web

PARTIE 1 : **Codes exécutés par les clients** **(HTML-CSS-JAVASCRIPT)**

Nhan LE THANH - Brice COVES

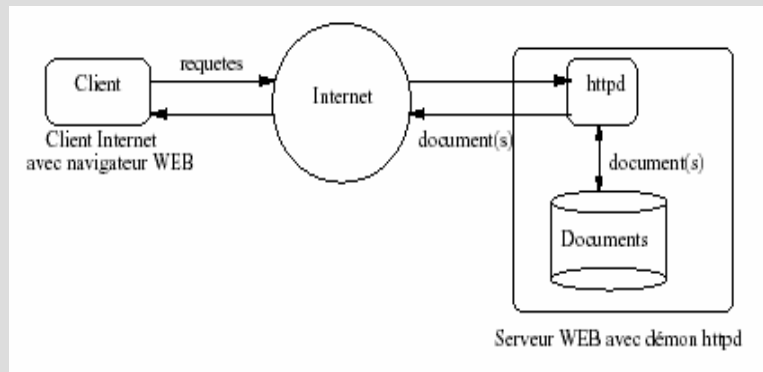
Université de Technologie de Nice

Plan

- **DEFINITIONS**
 - SERVICE HTTP ET NOTION URL
 - MODELE CLIENT/SERVEUR
 - MODELE CLIENT SERVEUR VIA LE SERVICE HTTP
 - ARCHITECTURE 3-TIERS
- **LANGAGE HTML**
 - INTRODUCTION
 - BALISES DE BASE
 - BALISES MULTIMEDIAS
 - IMAGES CLIQUABLES
 - FORMULAIRES
 - CADRES
- **FEUILLES DE STYLE CSS2**
 - INTRODUCTION
 - DEFINITIONS
 - CLASSES, IDENTIFIANT
 - UNITES
 - BOITES ET COUCHES
- **LANGAGE JAVASCRIPT**
 - INTRODUCTION
 - JAVASCRIPT ET HTML
 - ELEMENTS DE JAVASCRIPT
 - OBJETS DOM
 - AUTRES OBJETS
 - EVENEMENTS
 - DHTML

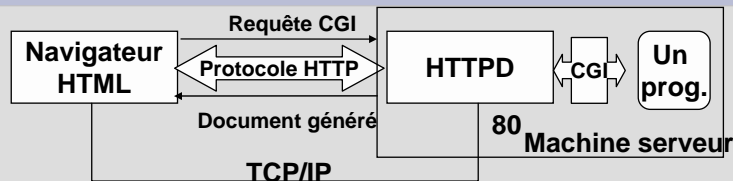
PROGRAMMATION WEB

Le service HTTP : modèle de base



PROGRAMMATION WEB

Modèle de document dynamique avec CGI



Le serveur HTTP dispose d'une interface CGI (Commun Gateway Interface), permettant d'invoquer l'exécution d'un programme sur le serveur :

- L'ordre sera transmis par le client (navigateur) sous la forme d'un hyperlien
- Ce lien contiendra des paramètres `www.test.com?parametre1=toto¶m2=titi`
- Le serveur HTTP reçoit cet ordre, il organisera ensuite l'exécution du programme demandé. La communication entre le programme et le serveur HTTPD est assurée par : les entrées/sorties standard (STDIN et STDOUT) et un ensemble de variables d'environnement du serveur HTTPD
- Le résultat de l'exécution sera envoyé au client web
- Protocoles web :
 - `http://` pour la consultation de pages web
 - `ftp://` pour la consultation de sites FTP
 - `telnet://` pour la connexion à un terminal distant
 - `mailto://` pour l'envoi d'un courrier électronique

PROGRAMMATION WEB

L' URL ou l'ADRESSE WEB

- **NOTION d'URL (Uniform Resource Locator) :**

- Un URL représente une identification d'un endroit où stocke une ressource sur le réseau Internet
- Une ressource peut-être stockée à plusieurs endroits, elle possède donc un ou plusieurs URL
- Format d'un URL : (note : le symbole [xxx] signifie que xxx est facultatif)

Protocole://	Hôte	[:Port]	Chemin	Nom	[#Ancre]	[Chemin privé]	[?Paramètres]
--------------	------	---------	--------	-----	----------	----------------	---------------

☞ Exemple 1 : une ressource statique est un fichier html

http://	www.site1.fr		/home/	index.htm?	#chapitre2		
---------	--------------	--	--------	------------	------------	--	--

☞ Exemple 2 : une ressource dynamique

http://	www.site2.fr	:8080	pages/	index.php		?nom=dupont&prenom=pierre
---------	--------------	-------	--------	-----------	--	---------------------------

PROGRAMMATION WEB

- **NOTION d'URI (Uniform Resource Identificator) :**

- Un URI représente une identification de l'origine d'une ressource sur le réseau Internet
- Une ressource peut-être stockée à plusieurs endroits, mais ces copies ont la même origine. **Une ressource a donc un et un seul URI**
- Format d'un URI : <type document | Propriétés | Origine>
- Exemple :

- La version 3.2 de HTML impose la spécification de l'URI de définition de ce langage (un DTD) dans la première ligne du document :

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

- La version XHTML impose la spécification de l'URI de définition de ce langage (un DTD) suivante :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/xhtml1/DTD/xhtml1-transitional.dtd">
```

PROGRAMMATION WEB

- Le passage de paramètres à une page de scripts :

Modes de passage de paramètres :

- **GET** : la chaîne de paramètres est envoyée avec l'URL après le caractère '?' et sera déposée dans une variable d'environnement, appelé QUERY_STRING du service HTTP (sur la machine serveur).
 - chaque paramètre comprend 2 opérandes un nom et une valeur les paramètres sont reliés par le symbole '&'
 - index.php?Nom-de-variable='Valeur-de-variable'
 - Convention : certaines règles de transformation automatique sont appliquées : le caractère d'espace (' ') est remplacé par %20
 - Avantage : simple, dans frames, liens cliquables
 - Inconvénients : limitée à 200 car., sécurité
- **POST** : la chaîne de paramètre sera envoyée indépendamment de l'URL et dirigé vers le fichier STDIN (Standard INPUT) du programme CGI.
 - Avantage : taille illimitée, traitement standard
 - Inconvénients: nécessite un formulaire
 - `<form action="url" name="nom" Method="post"> ... </form>`

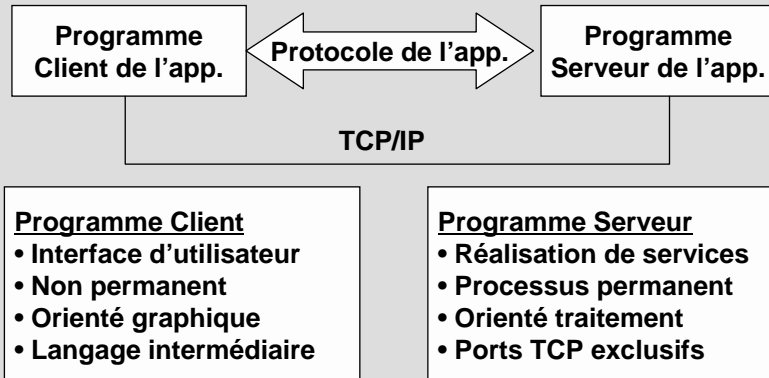
PROGRAMMATION WEB

- **Le retour de données depuis d'un programme CGI/script au serveur HTTP, et le retour du serveur vers le client**

- Format de données de retour : **Texte, HTML ou XHTML**
- Mode de passage du CGI vers le serveur HTTP : les données sorties du STDOUT (Standard OUTPUT) du programme CGI seront redirigées à l'entrée standard (stdin) du service HTTP qui les transmet au Client Web
- Ce résultat peut être n'importe quel document multimédia, depuis le simple texte ascii jusqu'à la vidéo. Dans le cas où la requête d'un client se limite à demander au serveur de lui fournir un fichier, le serveur se base sur l'extension de ce fichier pour déterminer son type (headers)
- Conformément au protocole HTTP, il faut alors transmettre ce type dans l'en-tête, avec la clause 'Content-type: *typeDocument*', pour que le navigateur sache comment décrypter les informations qui lui proviennent par la suite
- **Exemple** : Pour un fichier HTML par exemple, l'extension est le plus souvent *.html*, et la valeur de *typeDocument* est 'text/html'

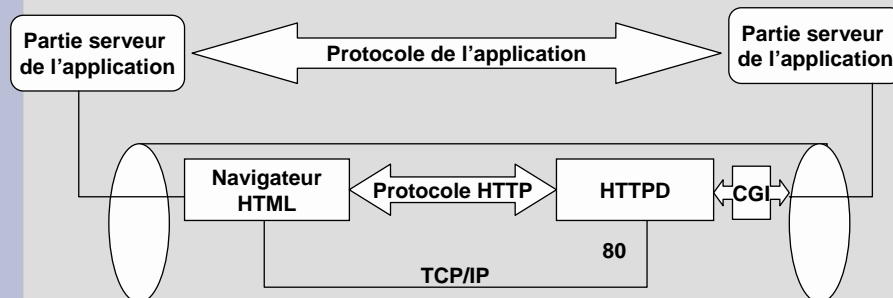
PROGRAMMATION WEB

- **Rappel du modèle de programmation Client/Serveur sur Internet :**
une application CL/SV sur le réseau Internet est constitué de 2 parties :
 - Un programme Client de l'application et
 - Un programme Serveur de l'application



PROGRAMMATION WEB

- **Première approche programmation CL/SV via web :**
 - Utilisation l'interface CGI pour activer la partie Serveur de l'application
 - Le programme Client est un script (X)HTML (HyperText Markup Language)
 - Le programme Serveur est un programme CGI
 - Le protocole de l'application doit respecter les restriction de HTTP



PROGRAMMATION WEB

- Les limites de la première approche

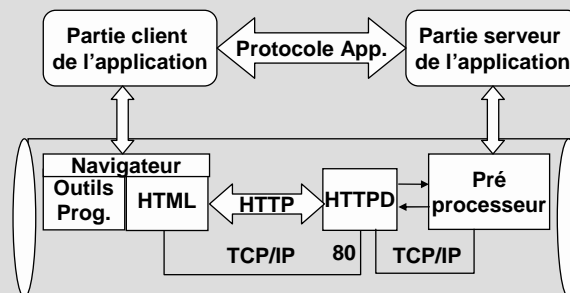
- Côté Client : absence des outils de programmation et les outils de présentation sont limités à la capacité du langage (X)HTML
- Côté Serveur : absence des outils de développement adaptés, la communication avec le serveur HTTP est à automatiser
- Côté Protocole : les paramètres sont passés en mode texte avec un format imposé (absence de sécurité et capacité limitée) et deux modes de récupération fixes (GET et POST)



PROGRAMMATION WEB

- **Modèle de programmation web à 2-tiers :**

- Tiers Client : Programme interface comprenant des outils de présentation et de programmation : (X)HTML, Plug-in, JavaScript, Style, Applet Java
- Tiers Serveur : Programme de services, appelé " objets de métiers ", développé depuis un environnement de développement normalisé, dit pré processeur de HTTP, Par exemple : PHP, JSP, ASP



PROGRAMMATION WEB

Modèle de programmation web à 2-tiers :

- Outils de présentation pour CLIENT :

- HTML/XML : Langage cadre permet d'une présentation de base et une intégration des différents modes de programmation et de présentation multimédia
- CSS2 : Langage de définition de style de présentation pour les balises (X)HTML. Il permet d'une présentation plus fine et paramétrable
- Plug-in : Des programmes d'interprétation des types de données multimédia permettant au navigateur d'afficher ces données (word, excel, ...)
- Flash animations dynamiques utilisant la technologie Macromédia

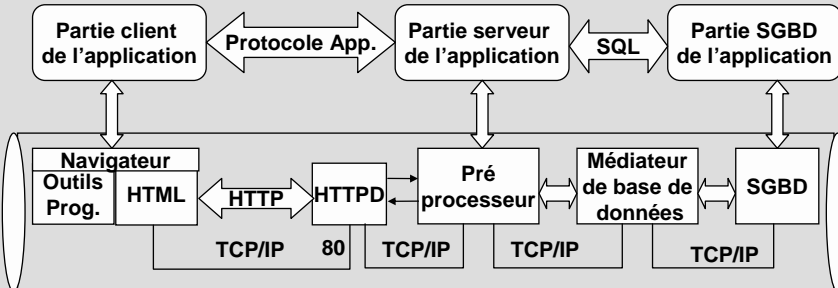
- Outils de programmation pour CLIENT :

- JavaScript : Langage de programmation intégré dans le (X)HTML, permettant de manipuler les objets documentaires de (X)HTML, de créer et capturer et manipuler des événements sur interface et d'effectuer des calculs
- Applets Java : des programmes java qui sont exécutés par une machine virtuelle Java intégrée dans le navigateur. Ces programmes échangent les données avec (X)HTML par les zones variables partagées.

PROGRAMMATION WEB

Modèle de programmation web à 3-tiers :

1er-Tiers de " Client " : Programmation interface d'utilisateur (html/css/js/flash/applet)	2e-Tiers d' "Objets de métier " : Programmation de services liés à l'application réelle (Langages : asp/php/servlet/cgi...)	3e-Tiers de "Bases de données " : Programmation des sources de données pour l'application (mysql,sql...)
---	--	--



PROGRAMMATION WEB

Langage HTML

• INTRODUCTION

- Langage à balises :

- chaque balise est une instruction de mise en forme de texte contenu dans cette balise
- Une balise peut contenir des textes et/ou des autres balises autorisées
- Permet d'explicitier les/la manière(s) de mettre en forme de textes

- Liens hypertexte :

- Chaque document HTML peut-être représenté par un URL comme une ressource sur Internet. `Cliquer ici`
- Un document HTML peut contenir des liens vers les autres documents HTML ou les autres sources de données (multimédia, programme)

PROGRAMMATION WEB

Langage HTML

• Introduction : Premier exemple de HTML

```
<HTML>
<HEAD>
  <TITLE> Mon premier document HTML </TITLE>
</HEAD>
<BODY BGCOLOR="blue">
  <CENTER>
    <H1>Mon premier document HTML</H1>
  </CENTER>
  <HR>
  Je suis étudiant(e) de l'Université de Nice et Sophia-Antipolis.
  <BR> Je travaille à l'IUT au sein du département Informatique.
</BODY>
</HTML>
```


PROGRAMMATION WEB

Langage HTML

BALISES DE BASE du langage HTML

● Format général de balise :

- ◆ Les balises conteneurs : les balises contiennent du texte et éventuellement des balises. Une balise conteneur est utilisée pour présenter ou structurer le texte et les balises qui lui appartiennent.

Exemple : `<CENTER> Mon document >/CENTER> <!-- Centrer le texte -->`

- ◆ Les balises vides : sont destinées à insérer un élément hors du texte dans le document : séparateur, saut paragraphe, saut de ligne, image, ...

Exemple : `<HR /> <!-- ajouter un séparateur horizontal -->`

Balises conteneur
`<nom_Balise {attribut='val'}>`
Corps_balise (texte et balises)
`</nom_Balise>`

Balises vides
`<nom_Balise {attribut='val'} />`

{attribut='val'} : liste de couples attribut='val' séparés par des espaces

PROGRAMMATION WEB

Langage HTML

QUELQUES BALISES DE BASE : balises de structuration

<code><HTML> ...</HTML></code>	Document HTML
<code><HEAD>...</HEAD></code>	Partie d'entête du document
<code><BODY>...</BODY></code>	Partie corps du document
<code><Hn>...</Hn>, n=1..6</code>	Titre dont la taille est dans l'ordre décroissant
<code><P></code> ou <code>
</code>	Ajout d'un paragraphe ou d'une nouvelle ligne
<code><HR></code>	Ajout d'une ligne horizontale
<code><PRE>...</PRE></code>	Texte pré formaté, préserver le format
<code>...</code>	Liste ordonnée (avec numéros latins ou arabes)
<code>...</code>	Liste non ordonnée (avec puces)
<code><DL>...</DL></code>	Liste des définitions
<code> </code>	Item d'une liste

PROGRAMMATION WEB

Langage HTML

Utilisation de HR :

- `<HR SIZE=nombre />` donne une épaisseur au trait.
- `<HR WIDTH=nombre|pourcentage />` donne la longueur du trait
- `<HR ALIGN=left|right|center />` le trait peut être aligné à droite, à gauche ou centré
- `<HR NOSHADE />` le trait n'a pas d'ombre
- `<HR ALIGN="left" size="1" />`

Utilisation de P et BR : (paragraphe/retour à la ligne)

- `<P ALIGN=CENTER|LEFT|RIGHT>`
- `<BR CLEAR=all|left|right>` CLEAR quand il a la valeur *left* va faire le même saut de ligne et fait un déplacement vers le bas en respectant la marge gauche. CLEAR avec la valeur *right* fait la même chose avec la marge droite. Ceci permet aussi de mettre des images autour de texte. L'attribut CLEAR peut être utilisé dans la balise `<P>...</P>`

PROGRAMMATION WEB

Langage HTML

• Balises de structuration (suite)

- `<OL TYPE=A|a||i|j|1 START=nb COMPACT>` Outre les numéros 1, 2, 3 etc.. vous pouvez accéder aux numéros A,B,C etc ... et pouvez commencer à un autre rang que le premier (START). COMPACT permet l'écriture de numéros plus compacts.
- `<UL TYPE=disc|circle|square>` Ainsi il est possible de décider que la puce sera un disque vide, un cercle ou un carré.
- Les listes peuvent être emboîtées en niveaux. Par exemple les lignes suivantes seront emboîtées:

Exemple :	<code></code>
	<code>Niveau 1</code>
• Niveau 1	<code></code>
• Niveau 2	<code>Niveau 2 </code>
• Niveau 2	<code>Niveau 2 </code>
• Niveau 1	<code></code>
	<code>Niveau 1</code>
	<code></code>

PROGRAMMATION WEB

Langage HTML

- **BALISES DE BASE :**
- **Balises entêtes : qui se mettent entre le couple de balises <HEAD> et </HEAD>.**
- Elles donnent des informations générales sur toute la page.
 - <ISINDEX PROMPT="chaîne"> permet d'afficher un champ recherche comme pour un formulaire mais de façon simplifiée. L'argument optionnel PROMPT permet d'afficher un texte avant le champs de recherche.
 - <TITLE> . . . </TITLE> donne le titre du document, ce titre apparaît dans le bandeau supérieur de la fenêtre de votre lecteur de Web. Cette balise doit être unique pour respecter la norme HTML 3.2 et elle est obligatoire
 - <LINK HREF=adresse rel=top|contents|index|glossary|copyright|next|previous|help|search rev=top|contents|index|glossary|copyright|next|previous|help|search title=valeur> établit un lien avec un autre document (de style en particulier)

PROGRAMMATION WEB

Langage HTML

- **Balises entêtes :**
 - <META> donne à votre document des informations qui seront lues par le serveur http. Ces informations sont généralement votre nom (NAME), le nom de l'auteur (AUTHOR), le contenu (CONTENT), une directive HTTP-EQUIV qui prend souvent la valeur Expires, Keywords, Reply_to. La directive *Expires* est utilisée par la plupart des navigateurs pour donner une date à partir de laquelle la page ne doit plus être conservée dans le cache de votre logiciel mais bien rechargée sur le serveur à chaque passage sur la page. **Utilisée pour les moteurs de recherche.**
 - <BASE HREF= ...> donne la base de l'adresse URL qui sera placée devant les références relatives dans le document, de façon à ce que hors contexte les fichiers soient cherchés à l'adresse : BASE adresse relative
- Exemple :
- <BASE HREF=http://nyx.unice.fr/> permettra de référencer les adresses du genre *toto.html* à l'adresse http:// nyx.unice.fr/toto.html

PROGRAMMATION WEB

Langage HTML

- Vous pouvez assigner les attributs suivants à la **balise BODY**
 - BACKGROUND=fichier - pour reproduire le fichier en motif de vos pages, les calculs du nombre d'images nécessaires seront faits automatiquement par votre logiciel lecteur de Web.
 - BGCOLOR=couleur - Définit la couleur du fond.
 - TEXT=couleur - Définit la couleur du texte.
 - LINK=couleur - Définit la couleur des URL non visités
 - VLINK=couleur - Définit la couleur des URL visités
 - ALINK=couleur - Définit la couleur des URL lorsqu'on clique dessus

PROGRAMMATION WEB

Langage HTML

- **QUELQUES BALISES DE BASE : balises de style entre <body> et </body>**

 ...	Gras
<I>...</I>	Italique
<center>...</center>	Centrer les éléments (textes, images)
<Blockquote>...</Blockquote>	afficher un bloc avec retrait à droite
<CITE>...</CITE>	Afficher des citations
<S>...</S>	Texte rayé
<U>...</U>	Texte souligné
...>STRONG>	Texte gras épais
<BLINK>...</BLINK>	Texte clignotant
^{...}	Texte exposant

PROGRAMMATION WEB

Langage HTML

• QUELQUES BALISES DE BASE : Police de caractères entre <body> et </body>

- Les polices de caractères peuvent être modifiées sur l'ensemble d'une page par la balise : <BASEFONT SIZE=valeur>
 - *Valeur* sert à changer la taille de la police par défaut. Six tailles sont disponibles et la valeur par défaut est 3, la plus grosse étant de valeur 7
- Une partie de texte dans la page peut voir sa taille modifiée si elle est entourée de la balise FONT qui contient un attribut **size** : ..
 - *Valeur* sert à spécifier, ici encore, 7 tailles de polices (le défaut étant 3)
 - La syntaxe SIZE=+i ou i peut être compris entre 1 et 7 est admise, elle permet de donner une taille relative par rapport à la taille en cours
- Exemple :
- permet de colorier les caractères (ici en rouge) donne comme résultat :

permet de colorier les caractères (ici en rouge)

PROGRAMMATION WEB

Langage HTML

• QUELQUES BALISES MULTIMEDIA : Liens et Ancres entre <body> et </body>

- **Liens** : Un lien permet de définir une région sensible au clic souris dans un document et en même temps, l'endroit où l'on va se retrouver après ce clic. Cette adresse peut être un document différent, par exemple, mais également un endroit précis (défini par une ancre) dans le document
- **Syntaxe d'un lien** :
texte
où les attributs rel, rev, et title sont peu utilisés et ont les mêmes significations que celles exposées dans la balise <LINK>
- **Syntaxe d'un Ancre** : définit un ancre avec le nom_ancree dans le document. Ex: chapitre 1
- Pour faire référence à une ancre : une ancre

PROGRAMMATION WEB

Langage HTML

- **QUELQUES BALISES MULTIMEDIA : Liens et Ancres entre <body> et </body>**

- Exemple d'un ancre dans un même document HTML :

- Définition : ``
- Référencer depuis d'un autre endroit du document :
`Chapitre 1`

- Exemple de référence à un autre document HTML :

- `Chapitre 1`

Rappels: une URL est toujours de la forme :

protocole ://host.domaine:port/pathname où protocole peut être :

file, http, news, gopher, telnet, wais ou mailto.

Les éléments port et pathname sont optionnels

- `cliquez ici pour charger le fichierA`

- Sur un PC on peut utiliser la ressource "file" pour référencer à un fichier, par exemple, mais cela n'est pas valable sur un serveur :

- ` fichier sur c:`

PROGRAMMATION WEB

Langage HTML

- **QUELQUES BALISES MULTIMEDIA : Images entre <body> et </body>**

- `` insérer l'image dans le document

- Attributs possibles

- ALT : pour afficher un texte si le lecteur de Web ne sait pas lire l'image
- ALIGN=left|right|top|texttop|middle|absmiddle|baseline|bottom|absbottom
- WIDTH=valeur HEIGHT=valeur – taille de l'image
- BORDER=valeur épaisseur de trait
- VSPACE contrôle l'espacement vertical en pixels au-dessus et en dessous de l'image. HSPACE contrôle l'espacement horizontal en pixels à droite et à gauche de l'image
- ISMAP et USEMAP pour utiliser l'image comme un ou des liens cliquables

PROGRAMMATION WEB

Langage HTML

- **QUELQUES BALISES MULTIMEDIA : Images entre <body> et </body>**

- Charger une image du serveur nyx à votre document

```
<IMG SRC="http://nyx.unice.fr/images/chato.jpg ">
```

- Charger une image de votre répertoire courant en spécifiant la taille

```
<IMG SRC="theatre.jpg" WIDTH=200 HEIGHT=500 Border = 10>
```

- Afficher en grande d'une image en cliquant sur l'image :

```
<A HREF="images/verdure.jpg ">
```

```
<IMG SRC="images/verdure.jpg " WIDTH=200 HEIGHT=500 Border = 0  
  ISMAP>
```

```
</A>
```

PROGRAMMATION WEB

Langage HTML

- **IMAGES CLIQUABLES entre <body> et </body>**

- Une image cliquable est une image contenant des zones servant les liens vers les autres documents

- Pour la mise en place d'une image cliquable, on doit composer 2 balises :

- une balise Image IMG pour charger l'image
- Une balise MAP pour définir les zones et les liens concernés
- la référence à la balise Map est faite par attribut **USEMAP**

- Balise MAP :

- Syntaxe : <MAP NAME=Nom>
<AREA SHAPE=valeur COORDS= "valeur_coordonnées" HREF=URL>
<AREA SHAPE=valeur COORDS= "valeur_coordonnées" HREF=URL>
<AREA SHAPE=valeur COORDS= "valeur_coordonnées" HREF=URL>
<AREA SHAPE=valeur COORDS= "valeur_coordonnées" HREF=URL>
</MAP>

PROGRAMMATION WEB Langage HTML

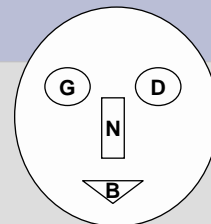
• IMAGES CLIQUABLES entre <body> et </body>

- Syntaxe de balise MAP (suite)
 - Les attributs de AREA prennent les valeurs suivantes :
 - SHAPE= rect|circle|poly|default la zone de définition est un rectangle, un cercle, un polygone ou le reste de la figure qui n'a pas été décrite. Ainsi :
 - rect : est défini par les coins opposés (par exemple "130,10,170,90")
 - circle : est défini par son centre et par le rayon (par exemple "50,50,10")
 - poly : est défini par un ensemble de points (par exemple "250,10 210,90 290,90")
 - default : est défini par les points non définis précédemment.
 - COORDS= des valeurs entre côtes et séparées par des virgules comme décrit précédemment.
 - HREF= l'URL qui sera atteinte après le clic.

PROGRAMMATION WEB Langage HTML

• IMAGES CLIQUABLES : Exemple

- G est un cercle et défini par 1 point (g1,g2) et un rayon r1
- D est un cercle et défini par 1 point (d1,d2) et un rayon r2
- N est un rectangle et défini par 2 points opposés : (n1;n2) et (n3,n4)
- B est un polygone et défini par 3 sommets : (b1,b2), (b3,b4), (b5,b6)



<http://nyx.unice.fr/images/tete.gif>

```
<IMG SRC="http://nyx.unice.fr/images/tete.gif" USERMAP="#unmap" >
<MAP Name="unmap"
<AREA SHAP=circle coords=" g1,g2,r1 " href=" œil-gauche.html >
<AREA SHAP=circle coords=" d1,d2,r2 " href=" œil-droit.html >
<AREA SHAP=rect coords=" n1,n2,n3,n4 " href=" nez.html >
<AREA SHAP=poly coords=" b1,b2,b3,b4,b5,b6 " href="bouche.html >
</MAP>
```


PROGRAMMATION WEB

Langage HTML

- **Tableau : présenter des données sous forme d'un tableau.**

Le tableau sera défini ligne par ligne. Un tableau est encadré par la balise TABLE

- `<TABLE ...></TABLE>` Cette commande permet de définir des tableaux, elle inclut toutes les commandes sur le nombre de lignes et le nombre de colonnes. D'autres attributs (traits, ombres) peuvent être définis.
- `<TR ...></TR>` Chaque balise TR définit une ligne du tableau. Donc pour trois lignes dans le tableau, on retrouvera trois balises TR.
- `<TD ...></TD>` Chaque balise TD définit une cellule à l'intérieur d'une ligne. On trouve autant de balises TD que de colonnes par ligne.
- `<TH ...></TH>` Ce couple de balises concerne les entêtes du tableau, c'est-à-dire la première ligne ou la première colonne. La différence avec une balise TD réside dans l'alignement, qui par défaut est centré et la police de caractères qui est en gras
- `<CAPTION ...></CAPTION>` La balise CAPTION peut être utilisée dans un texte délimité par les balises TABLE (mais pas dans ceux délimités par TR ou TD), elle sert à donner un titre au tableau et peut être en haut (`ALIGN=TOP`) ou en bas (`ALIGN=BOTTOM`)

PROGRAMMATION WEB

Langage HTML

- **Tableau : les attributs, entre `<body>` et `</body>`**

- `BORDER` : définit les bordures du tableau. Se met dans la balise TABLE pour indiquer que le tableau sera délimité. L'absence de l'attribut BORDER, équivalent à `BORDER=0`, n'affichera pas de bord au tableau. Par défaut `BORDER` vaut 1.
- `ALIGN=TOP|BOTTOM` Indique si la légende est en haut ou en bas. S'applique à la balise CAPTION
- `ALIGN=LEFT|RIGHT|CENTER` Dans les balises TR, TH et TD permet d'aligner les cellules au centre à gauche ou à droite
- `VALIGN=TOP|MIDDLE|BOTTOM` Dans les balises TR, TH et TD permet d'aligner le texte verticalement en haut, au centre ou en bas
- `NOWRAP` Dans les balises TH ou TD demande que les lignes dans les cellules ne soient pas coupées
- `COLSPAN` Dans les balises TH ou TD donne le nombre de colonnes à fusionner dans la cellule. Par défaut la valeur est 1. Avec une valeur supérieure on peut recouvrir plusieurs colonnes

PROGRAMMATION WEB

Langage HTML

- **Tableau : les attributs, entre <body> et </body>**

- ROWSPAN Dans les balises TH ou TD donne le nombre de lignes à mettre dans la cellule. Par défaut la valeur est 1. Avec une valeur supérieure on peut recouvrir plusieurs lignes
- CELLSPACING=valeur La valeur CELLSPACING est utilisable avec la commande TABLE et définit l'espacement inter cellule. La valeur par défaut est 2
- CELLPADDING=valeur La valeur CELLPADDING est utilisable avec la commande TABLE et définit la distance entre le bord de la cellule et son contenu
- WIDTH=pourcentage Cet attribut de la commande TABLE est utilisé pour donner la largeur du tableau en pourcentage de la largeur du document. Le champ pourcentage devra être de la forme "nb%"

PROGRAMMATION WEB

Langage HTML

- **Tableau : Exemple, entre <body> et </body>**

```
<TABLE BORDER=1 CELLSPACING=4
CELLPADDING=3>
<TR>
  <TD COLSPAN=2>
    AAAAAAAAAAAAAA
  </TD>
</TR>
<TR>
  <TD ROWSPAN=2 VALIGN=middle
    ALIGN=center>BBBBBBBB
  </TD>
  <TD ALIGN=right>123456</TD>
</TR>
<TR>
  <TD ALIGN=right>456</TD>
</TR>
</TABLE>
```

AAAAAAAAAAAAAA	
BBBBBBBB	123456
	456

PROGRAMMATION WEB

Langage HTML

- **FORMULAIRE : Généralités**

- Les formulaires sont les seuls outils permettant de récolter des informations en provenance des lecteurs de Web. Ils seront traités sur le serveur par des programmes appelés CGI
- Les balises FORM permettent de définir les interfaces d'échange d'informations entre un utilisateur et le serveur http. Cette interface est graphique et utilise les listes déroulantes, les boutons poussoirs, les boutons radios et tous les éléments graphiques présents dans les systèmes actuels. Plusieurs balises FORM peuvent être présentes dans un document HTML mais en aucun cas les balises FORM ne peuvent être emboîtées.
- La balise FORM enverra une requête sur le serveur http sur lequel l'utilisateur est connecté lorsque ce dernier appuiera sur le bouton d'envoi de la requête.

PROGRAMMATION WEB

Langage HTML

- **FORMULAIRE La syntaxe d'une balise FORM :**

```
<FORM ACTION="url" method="post" enctype="" name="form1"> ...  
</FORM>
```

(entre <body> et </body>)

- Les attributs suivants peuvent être utilisés :
 - ACTION fournit une adresse URL sur laquelle la requête issue de la FORM va être envoyée. Si le champ ACTION est absent, la requête sera envoyée sur le serveur courant.
 - METHOD est la méthode d'accès au serveur http. On trouve deux méthodes d'accès POST et GET.
 - ENCTYPE donne la méthode d'encodage pour la FORM. Cet attribut n'est pris en compte que si la méthode est POST. Dans les versions futures, ce champ permettra l'encryptage de données.
- Le texte encadré par le couple de balises <FORM> et </FORM> peut contenir un ensemble de balises:
 - INPUT
 - SELECT, OPTION
 - TEXTAREA

PROGRAMMATION WEB

Langage HTML

FORMULAIRE : Balise INPUT

- La commande INPUT est utilisée avec des attributs pour spécifier les caractéristiques de la commande clavier (ou souris) désirée. Les différents attributs que peut utiliser la balise INPUT sont :
 - TYPE qui peut prendre les valeurs :
 - text pour les entrées de type texte. Il est à noter que les caractères entrés par l'utilisateur sont convertis automatiquement avec les quotations (accents, champs &)
 - hidden l'entrée ne sera pas affichée dans la page HTML mais grâce à l'ordre VALUE la requête partira vers le serveur http avec la valeur donnée. (ceci permet de forcer une des valeurs qu'attend le serveur sans l'afficher)
 - password pour les entrées de type mot de passe, ou les caractères entrés seront cachés par des caractères astérisques.
 - checkbox pour les boîtes à cocher.
 - radio pour les boutons radios
 - submit un bouton poussoir qui provoque l'envoi de la requête FORM.
 - reset un bouton poussoir qui provoque la remise aux valeurs initiales de tous les champs de la FORM
 - image permet de remplacer le bouton par défaut par une image.
 - file permet de soumettre un fichier au serveur.

PROGRAMMATION WEB

Langage HTML

- FORMULAIRE : **Balise INPUT** (suite) : les **attributs**
 - NAME est le nom d'utilisation par l'auteur de la page, et qui n'est pas le nom affiché à l'utilisateur.
 - VALUE sert à spécifier la valeur par défaut d'un champ de saisie.
 - CHECKED indique que le bouton radio ou la boîte à cocher est sur ON.
 - SIZE est la taille du champ pour les champs caractères. La valeur par défaut est 20. Si la zone valeur de size est remplie avec deux valeurs séparées par une virgule, le champ texte aura une longueur déterminée par la première valeur et un nombre de lignes défini par la deuxième valeur.
 - MAXLENGTH est le nombre maximum de caractères qui sont acceptés pour les champs textuels. Cet attribut permet de gérer les défilements si MAXLENGTH est plus grand que SIZE.
 - SRC est l'URL de l'image dans le cas où TYPE=image dans ce cas l'attribut ALIGN peut prendre les valeurs LEFT|RIGHT|TOP|MIDDLE|BOTTOM. Dans ce cas, les paramètres x et y sont passés au serveurs sous la forme du nom de la variable donnée par NAME suivi par les extensions .x et .y
 - l'attribut ENCTYPE prend la valeur *multipart/form-data* dans le cas où TYPE=file

PROGRAMMATION WEB

Langage HTML

- **FORMULAIRE : Balise INPUT - exemple**

```
<FORM METHOD="POST"
ACTION="http://nyx.unice.fr/~nlt/test.cgi">
Nom <INPUT NAME="Nom"><P>
Prénom <INPUT NAME="Prenom">
<OL> <LI> <INPUT TYPE="checkbox" CHECKED
NAME=" Ecivil" VALUE="Marie">Marié
<LI> <INPUT TYPE="checkbox"
NAME=" nationalite" VALUE="francais"> francais
</OL>
Volontaire
<DL> <DD> <INPUT TYPE="radio" NAME=" choix"
VALUE="Oui" CHECKED> <I>Oui</I>
<DD> <INPUT TYPE="radio"
NAME=" choix" VALUE="Non"> <I>Non</I>
<DD> <INPUT TYPE="radio" NAME=" choix"
VALUE="Ne sait pas"> <I>Ne sait pas</I>
</DL>
<INPUT TYPE="submit" VALUE=" Envoyer">
<INPUT TYPE="reset" VALUE=" Annuler">
</FORM>
```

Nom

Prénom

- marié
- francais

Volontaire

Oui

Non

Ne sait pas

PROGRAMMATION WEB

Langage HTML

- **FORMULAIRE : La balise SELECT** permet de donner la liste des champs accessibles pour les menus déroulants. La commande SELECT est utilisée avec la syntaxe suivante :

```
<SELECT NAME="a-menu">
<OPTION VALUE=valeur> Option 1 </OPTION>
<OPTION VALUE=valeur> Option 2 </OPTION>
</SELECT>
```

– Les **attributs** de la commande SELECT sont les suivants :

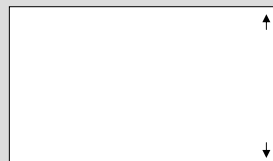
- **NAME** est le nom symbolique de l'élément. C'est-à-dire le nom du champ utilisé par l'auteur de la page HTML. Ce nom n'est pas visible pour les utilisateurs.
- **SIZE** donne le nombre d'éléments qui seront affichés dans le menu. Les autres valeurs seront accessibles au moyen d'un ascenseur.
- **MULTIPLE** indique que la commande SELECT pourra avoir plusieurs sélections.
- **SELECTED** indique que l'option est sélectionnée par défaut.
- **VALUE** est un paramètre optionnel permettant de donner à la variable en regard de NAME la valeur désirée

PROGRAMMATION WEB

Langage HTML

- **FORMULAIRE** : La balise **TEXTAREA** est utilisée avec la syntaxe suivante :

```
<TEXTAREA NAME=... ROWS=.. COLS=.. >  
valeur par défaut  
</TEXTAREA>
```



- La balise **TEXTAREA** contient les attributs :
 - **NAME** le nom symbolique de l'entrée
 - **ROWS** nombre de lignes du champ de saisie
 - **COLS** est la longueur de la ligne de saisie du champ texte.

PROGRAMMATION WEB

Langage HTML

- **LES CADRES : INTRODUCTION**

- Balises **FRAMESET** et **FRAME** permettent de définir des fenêtres dans une page HTML, chacune de ces fenêtres se comportant comme une page HTML indépendante. Ainsi on peut disposer d'un ascenseur pour chacune de ces zones, faire défiler des textes ou des images dans chacune d'entre elles.
- Les frames mettent en jeu deux éléments :
 - la balise **FRAMESET** définissant la taille des fenêtres qui remplace la balise **BODY**. Le document contenant les framesets doit être lancé en premier (fichier index). Les balises Framesets pourront être emboîtées permettant de définir une hiérarchie de fenêtres, dans laquelle les nœuds intermédiaires sont des fenêtres virtuelles et les nœuds feuilles sont des fenêtres réelles. **Un FRAMESET est un découpage qui contient des FRAME spécifiant l'url de la page du FRAME.**
Ex: FRAMESET contient deux frames un menu et une page accueil référencé par menu.htm et accueil.htm. Le frameset sera index.htm.
 - la balise **FRAME** définissant le contenu des fenêtres au niveau feuille de la hiérarchie de fenêtre.

PROGRAMMATION WEB

Langage HTML

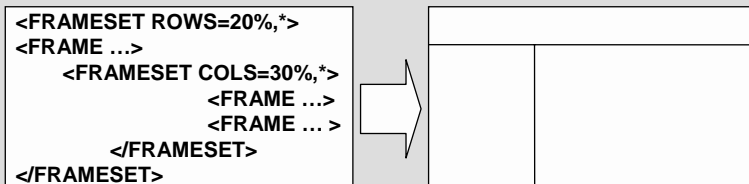
- **LES CADRES : Balise FRAMESET** remplace la balise BODY dans le document concerné, qui doit être chargé en premier :

- Syntaxe :

```
<FRAMESET> Corps </FRAMESET>
```

- La balise FRAMESET comprend deux attributs

- ROWS=hauteur1,hauteur2,...
- COLS=largeur1,largeur2,...
- Les valeurs peuvent être des pixels (par exemple 40) , des pourcentages (par exemple 20%) ou des résultantes (par exemple *).



PROGRAMMATION WEB

Langage HTML

- **LES CADRES : Balise FRAME** est une balise vide qui définit les caractéristique d'une fenêtre réelle :

- Syntaxe : <FRAME attributs >

- La balise FRAME comprend plusieurs attributs

- SRC=URL : donne l'URL à mettre dans la FRAME
- NAME=*nom de la fenêtre* : donne le nom de la fenêtre.
- MARGINWIDTH=*valeur* : donne la valeur (en pixels) des marges pour la fenêtre.
- MARGINHEIGHT=*valeur* : donne la valeur (en pixels) de l'en-tête et du bas de page de la fenêtre.
- SCROLLING="*yes/no/auto*" : permet les flèches déroulantes de défilement du texte.
- NORESIZE Ce flag indique que la taille de la fenêtre n'est pas redéfinissable par l'utilisateur.

- La balise <NOFRAMES> </NOFRAMES> permet de créer une page HTML visible par ceux qui ne disposent pas d'un outil compatible

PROGRAMMATION WEB

Langage XHTML

- **INTRODUCTION : Un premier exemple de XHTML**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title> Mon premier document xhtml </title>
</head>
<body bgcolor="white">
<center><h1>Mon premier document xhtml</h1></center>
<hr/>
Je suis étudiant( e) de la LPSIL de l'Université de Nice et
Sophia-Antipolis.
<br/> Je travaille à l'IUT Côte d'Azur, au sein du département
informatique.
</body>
</html>
```

PROGRAMMATION WEB

Langage XHTML

- **Description XHTML**

- Une version plus stricte du html qui est une reformulation de HTML4.01 en XML :
 - Les balises de HTML
 - La syntaxe de XML
- Un texte xhtml *valide* doit comporter
 - La déclaration de type de document (DTD pour data type definition): contient toutes les particularités qu'un document peut avoir concernant sa structure ou sa présentation. Elle permet d'indiquer au navigateur comment se comporter avec le code qu'il reçoit, et peut être plus ou moins souple.
 - Une balise html avec l'attribut xmlns : désigne un nom d'espace de nommage XML.
 - Des balises <head> définissent les entêtes du document : son titre, le type de données qu'il contient... **Seule la balise title est obligatoire.**
 - Des balises body : représentent le contenu proprement dit du texte.
- En xhtml, toute balise doit être saisies en **minuscules**.

PROGRAMMATION WEB

Langage XHTML

- **Les règles à appliquer pour valider du XHTML sont :**
 - La fermeture de toutes les balises
 - Ne pas laisser de balises se chevaucher
 - Ecrire en minuscule le nom des balises et des attributs
 - Ne pas laisser d'attributs présents sans valeur associée
 - Encadrer les valeurs appliquées aux attributs par des guillemets
 - Spécifier le standard utilisé pour l'écriture de la page : balise DOCTYPE
 - Ne plus utiliser les balises et les attributs obsolètes

PROGRAMMATION WEB

Langage XHTML

La fermeture des balises est obligatoire :

- soit par une balise fermante : `<balise>...</balise>`
- soit par la syntaxe `" />".`

Les balises n'ayant pas de tag de fermeture obligatoire s'écrivent :

INPUT `<input ...></input>` ou `<input ... />`
IMG `` ou ``
OPTION `<option ...> ... </option>`
HR `<hr ... />`
BR `<hr />`
LI `<li ... > ... `
DT `<dt ... > ... </dt>`
DD `<dd ... > ... </dd>`

PROGRAMMATION WEB

Langage XHTML

- Chevauchement des balises :
 - Cette nécessité vient à nouveau d'une cohérence syntaxique avec le XML. L'imbrication des tags doit être respectée : l'ordre de fermeture des tags doit s'effectuer dans l'ordre inverse de celui d'ouverture. En d'autres termes, le dernier tag ouvert doit être le premier fermé.
 - Exemples :
`<u>mauvaise écriture des tags</u>`
`<u>écriture correcte des tags</u>`
`<u>écriture correcte des tags</u>`
- Ecriture des balises et attributs en minuscules :
 - Il n'est plus possible en XHTML d'écrire par exemple :
`<INPUT TYPE="hidden" name="test">`
 - Il faut écrire les mots input et type en minuscule et toujours penser à bien écrire la fermeture de la balise:
`<input type="hidden" name="test" />`

PROGRAMMATION WEB

Langage XHTML

- Spécifier la valeur de chaque attribut :
 - Certaines des balises HTML possèdent des attributs sans valeurs. La balise `<td>` utilise par exemple l'attribut `nowrap` qui empêche les retours à la ligne dans la cellule. les attributs `checked` ou `disabled` ne prennent pas non plus de valeur.
 - Pour répondre à cette norme, il faut donner la valeur correspondante au nom de l'attribut.
Une zone de texte non éditable s'écrira alors :
`<input type="text" disabled="disabled" value="Texte définitif" />`
 - Toute valeur associée à un attribut doit être encadrée par des guillemets pour être valable. Il n'est donc pas possible d'écrire explicitement des guillemets dans la valeur d'un attribut. Pour contourner ce problème, il suffit d'écrire le code HTML associé (Tableau des caractères HTML).

PROGRAMMATION WEB

Langage XHTML

- **Les attributs doivent tous être renseignés :**

- Par exemple la syntaxe d'une textbox non modifiable est en HTML : `<input type="text" disabled>`
- Cette syntaxe est incorrecte en XHTML, son écriture correcte est : `<input type="text" disabled="disabled" />`
- Les attributs suivant ne nécessite pas de valeur en HTML. Pour qu'ils soient bien interprétés en XHTML il est nécessaire de leur donner comme valeur leur propre nom :
 - compact compact="compact"
 - checked checked="checked"
 - declare declare="declare"
 - readonly readonly="readonly"
 - disabled disabled="disabled"
 - selected selected="selected"
 - defer defer="defer"
 - ismap ismap=" ismap"
 - nohref nohref="nohref"
 - noshade noshade="noshade"
 - nowrap nowrap="nowrap"
 - multiple multiple="multiple"
 - noresize noresize="noresize"

PROGRAMMATION WEB

Langage XHTML

- **Caractères HTML spéciaux**

Á	Á	Lettre A
á	á	
Â	Â	
â	â	
à	à	
é	é	Lettre E
ê	ê	
è	è	
ô	ô	Lettre O
ö	ö	

 		espace
&	&	& commercial
€	€	euro
£	£	livre sterling
¥	¥	Yen
"	"	double quote
©	©	copyright
ç	ç	C cédille
®	®	registre
←	← ↓ ↑ →	les flèches
§	§	Paragraphe
¶	¶	Paragraphe
¿	¿	? inversé
»	"	double flèche droite
«	"	double flèche gauche

PROGRAMMATION WEB

Langage XHTML

- Caractères HTML spéciaux

α	α	Alpha
β	β	Beta
χ	χ	Chi
ψ	ψ	Psi
Ω	Ω	Grand Omega
ω	ω	Petit Omega

°	°	degré
‰	‰	Pour mille
′	'	Prime
″	"	Seconde
∞	∞	Infini
±	±	Plus ou Moins
<	<	inférieur à
>	>	supérieur à
¼	¼	fraction 1/4
½	½	fraction 1/2
¾	¾	fraction 3/4
¹	¹	Puissance 1
²	²	Puissance 2
³	³	Puissance 3