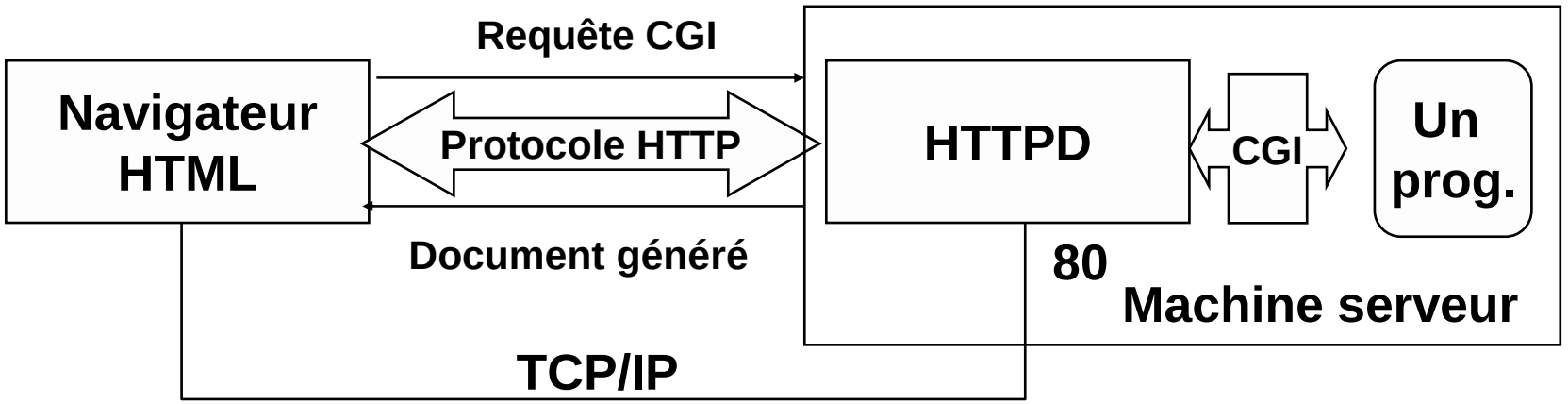
HTML / xhtml

Modèles de programmation via le service Web

- Rappel du service HTTP : Modèle de document dynamique avec CGI



- Le serveur HTTP dispose une interface, dit CGI (Commun Gateway Interface), permettant d'invoquer l'exécution d'un programme quelconque se trouvant sur la même machine serveur selon un ordre venant du client web :
  - l'ordre d'exécution sera sous forme d'un hyperlien avec quelques conventions spécifique
  - 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 par le serveur HTTP au client web

Modèles de programmation via le service 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 dont un ou plusieurs URL
  - Format d'un URL : (note : le symbole [xxx] signifie que xxx est facultatif)

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

- Exemple 1 : une ressource statique est un fichier html

http://	deptinfo.unice.fr		/~renevier/	iut		

- Exemple 2 : une ressource dynamique générée par un programme cgi :

http://	iihm.imag.fr		/cgi-bin/Vitesse2/	vitesse2.bat		?Keywords=unsa&SearchEngine=Google&Kind=Search&InfoSpace=&MaxInfoNumber=100&VitesseMode=Win

Codage d'une URL

Tabulation	%09
Espace	%20
"	%22
#	%23
%	%25
&	%26
(	%28
)	%29
+	%2B
,	%2C
.	%2E
/	%2F
:	%3A
;	%3B

<	%3C
=	%3D
>	%3E
?	%3F
@	%40
[	%5B
\	%5C
]	%5D
^	%5E
'	%60
{	%7B
|	%7C
}	%7D
~	%7E

Modèles de programmation via le service 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 4.01 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 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

Modèles de programmation via le service Web

- Le passage de paramètres à un programme CGI :
  - Format de paramètre : en texte ASCII
    - chaque paramètre comprend 2 opérandes «Nom-de-variable » et «Valeur-de-variable » reliées par le symbole '='
    - les paramètres sont reliés par le symbole '&'
    - Convention : certaines règles de transformation automatique sont appliquées : le caractère d'espace (' ') est remplacé par '+', …

var1	=	val1	&	var2	=	val2	&	…	&	varn	=	valn

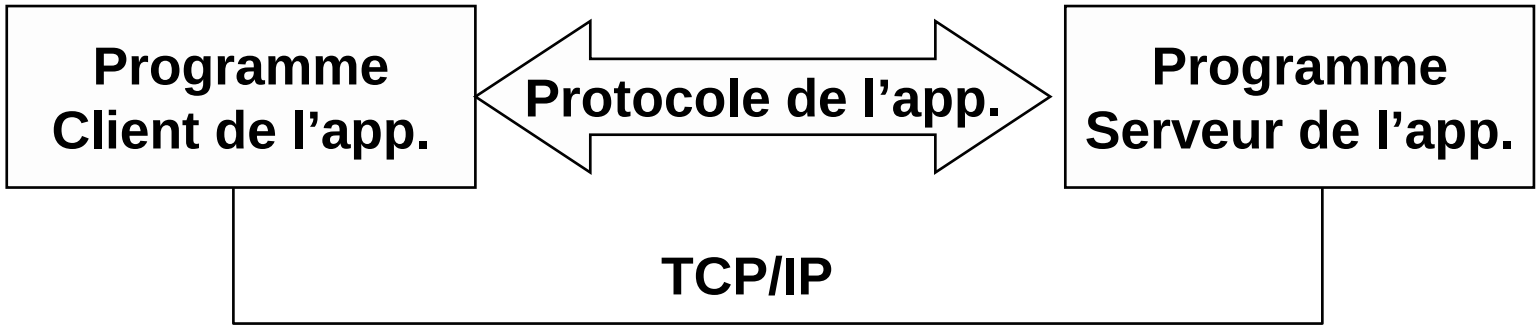
  - Modes de passage de paramètres :
    - GET : la chaîne de paramètres est envoyée avec l'URL après le caractère '?' et sera déposée dans une variable d'environnement, appelé QUERY_STRING du service HTTP (sur la machine serveur). Avantage : simple; Inconvénient : taille limitée à 200 caractères
    - POST : la chaîne de paramètre sera envoyée indépendamment de l'URL et dirigée vers le fichier STDIN (Standard INPUT) du programme CGI. Avantage : taille illimitée, traitement standard

Modèles de programmation via le service Web

- Le retour de données depuis d'un programme CGI au serveur HTTP :
  - Format de données de retour : Texte HTML
  - 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
  - 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'

Modèles de programmation via le service 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**

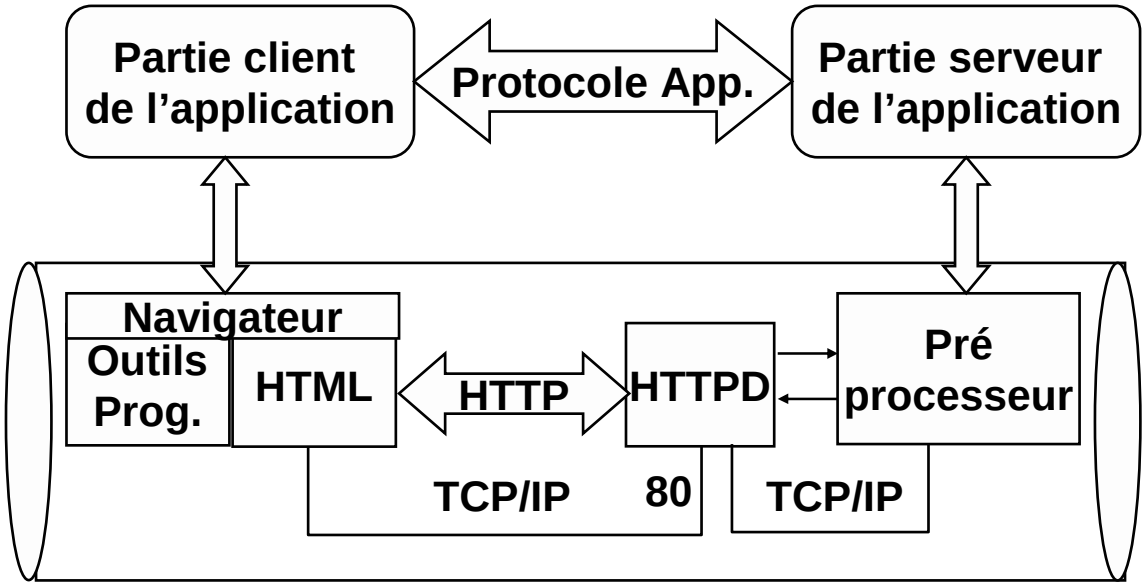


Programme Client
- **Interface d'utilisateur**
- **Non permanent**
- **Orienté graphique**
- **Langage intermédiaire**

Programme Serveur
- **Réalisation de services**
- **Processus permanent**
- **Orienté traitement**
- **Ports TCP exclusifs**

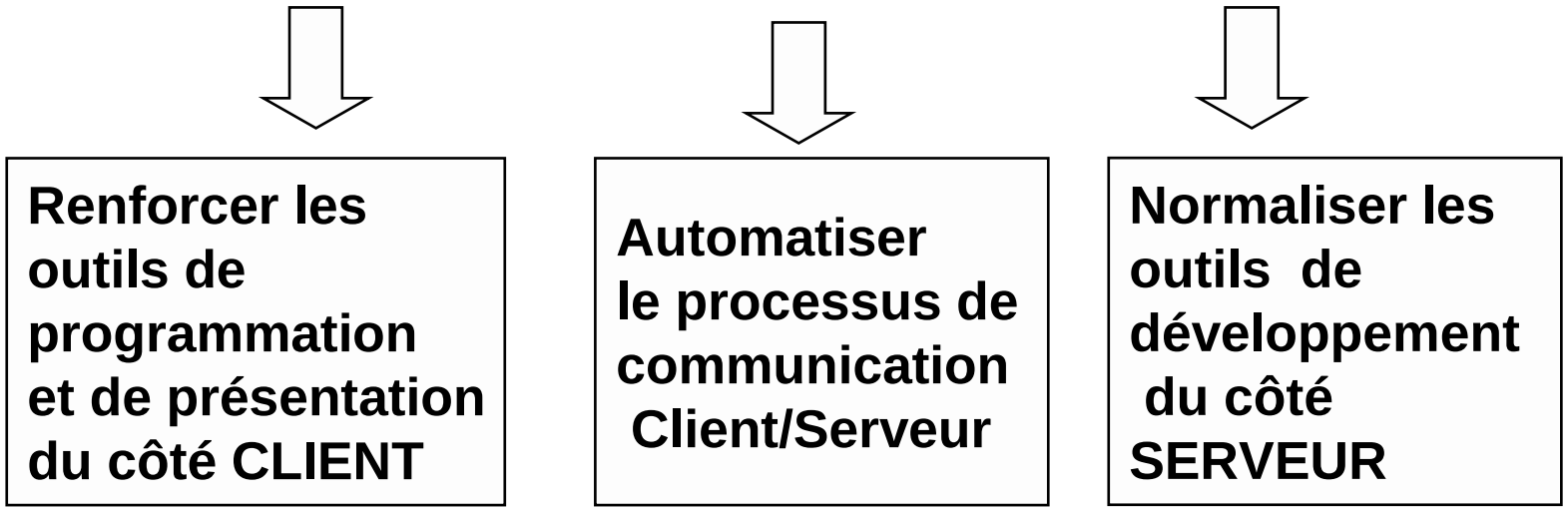
Modèles de programmation via le service Web

- Modèle de programmation web à 2-tiers :
  - Tiers Client : Programme interface comprenant des outils de présentation et de programmation : 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



Modèles de programmation via le service 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 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)

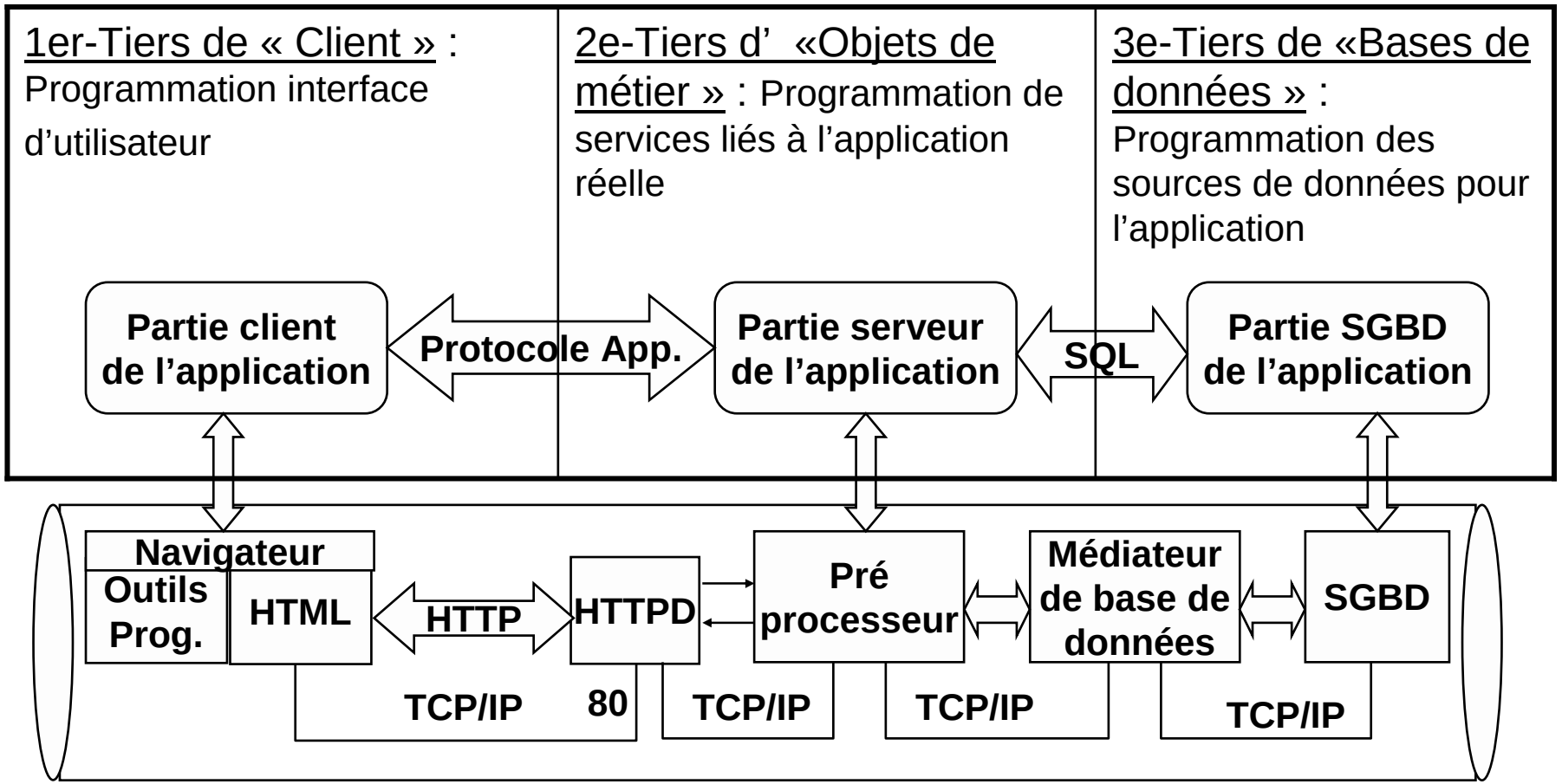


Modèles de programmation via le service Web

- 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 programmations et de présentation multimédia
    - CSS2 : Langage de définition de style de présentation pour les balises HTML. Il permet d'une présentation plus fine et paramétrable
    - Plug-in : Des programme d'interprète des types de données multimédia permettant au navigateur d'afficher ces données (word, excel, …)
  - Outils de programmation pour CLIENT :
    - JavaScript : Langage de programmation intégré dans le HTML, permettant de manipuler les objets documentaires de HTML, de créer et capturer et manipuler des évènements sur interface et d'effectuer des calculs
    - Applets Java, Flash, etc.

Modèles de programmation via le service Web

- Modèle de programmation web à 3-tiers :



Protocole HTTP

- **HTTP (HyperText Transfer Protocol) : protocole le plus utilisé sur Internet depuis 1990.**
  - **La version 0.9 était uniquement destinée à transférer des données sur Internet (en particulier des pages Web écrites en HTML).**
  - **La version 1.0 du protocole (la plus utilisée) permet désormais de transférer des messages avec des en-têtes décrivant le contenu du message en utilisant un codage de type MINE.**
- **Transfert de fichiers localisés grâce à une URL entre un navigateur (le client) et un serveur Web (*httpd*).**
  - **Le navigateur effectue une requête HTTP**
  - **Le serveur traite la requête puis envoie une réponse HTTP**
- ***RFC 1945 - Hypertext Transfer Protocol -- HTTP/1.0***
- ***RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1***

Requête HTTP

- **Envoyée au serveur par le navigateur, composée des éléments suivants**
- **Une ligne de requête (3 éléments séparés par un espace)**
  - **La méthode**
  - **L'URL**
  - **La version du protocole utilisé par le client (généralement *HTTP/1.0*)**
- **Les champs d'en-tête de la requête: ensemble de lignes facultatives donnant des informations supplémentaires sur la requête et/ou le client (navigateur, OS)**
  - **Ces lignes sont composées d'un nom (type d'en-tête) suivi de deux points (:) et de la valeur**
- **Le corps de la requête: ensemble de lignes optionnelles séparées des lignes précédentes par une ligne vide**
  - **Permet (par exemple) un envoi de données par une commande POST lors de l'envoi de données au serveur par un formulaire**

Requête HTTP

- **Syntaxe (<*crlf*> = saut de ligne):**

  METHODE URL VERSION<*crlf*>

  EN-TETE : Valeur<*crlf*>

  . . .

  EN-TETE : Valeur<*crlf*>

  Ligne vide<*crlf*>

  CORPS DE LA REQUETE

- **Exemple de requête HTTP:**

  GET http://www.commentcamarche.net HTTP/1.0

  Accept : text/html

  If-Modified-Since : Saturday, 15-January-2000 14:37:11 GMT

  User-Agent : Mozilla/4.0 (compatible; MSIE 5.0; Windows 95)

Commande d'une requête http

Commande	Description
GET	**Requête de la ressource située à l'URL spécifiée**
HEAD	**Requête de l'en-tête de la ressource située à l'URL spécifiée**
POST	**Envoi de données au programme situé à l'URL spécifiée**
PUT	**Envoi de données à l'URL spécifiée**
DELETE	**Suppression de la ressource située à l'URL spécifiée**

Entêtes d'une requête http

Nom de l'en-tête	Description
Accept	**Type de contenu accepté par le navigateur (par exemple *text/html*).**
Accept-Charset	**Jeu de caractères attendu par le navigateur**
Accept-Encoding	**Codage de données accepté par le navigateur**
Accept-Language	**Langage attendu par le navigateur (anglais par défaut)**
Authorization	**Identification du navigateur auprès du serveur**
Content-Encoding	**Type de codage du corps de la requête**
Content-Language	**Type de langage du corps de la requête**

Entêtes d'une requête http

Nom de l'en-tête	Description
Content-Length	**Longueur du corps de la requête**
Content-Type	**Type de contenu du corps de la requête (par exemple *text/html*).**
Date	**Date de début de transfert des données**
Forwarded	**Utilisé par les machines intermédiaires entre le browser et le serveur**
From	**Permet de spécifier l'adresse e-mail du client**
If-Modified-Since	**Permet de spécifier que le document doit être envoyé s'il a été modifié depuis une certaine date**
Link	**Relation entre deux URL**
Orig-URL	**URL d'origine de la requête**
Referer	**URL du lien à partir duquel la requête a été effectuée**
User-Agent	**Chaîne donnant des informations sur le client, comme le nom et la version du navigateur, du système d'exploitation**

Réponse HTTP

- Une ligne de statut:
  - La version du protocole utilisé
  - Le code de statut
  - La signification du code
- Les champs d'en-tête de la réponse: ensemble de lignes facultatives donnant des informations supplémentaires sur la réponse et/ou le serveur.
  - Ces lignes sont composées d'un type d'en-tête suivi de deux points (:) et de la valeur de l'en-tête
- Le corps de la réponse: contient le document demandé

Réponse HTTP

- **Syntaxe**

  VERSION-HTTP CODE EXPLICATION<*crlf*>
  EN-TETE : Valeur<*crlf*>
  . . .
  EN-TETE : Valeur<*crlf*>
  Ligne vide<*crlf*>
  CORPS DE LA REPONSE

- **Exemple de réponse HTTP:**

  HTTP/1.0 200 OK
  Date : Sat, 15 Jan 2000 14:37:12 GMT
  Server : Microsoft-IIS/2.0
  Content-Type : text/HTML
  Content-Length : 1245
  Last-Modified : Fri, 14 Jan 2000 08:25:13 GMT

  …

Entêtes des réponses HTTP

Nom de l'en-tête	Description
Content-Encoding	**Type de codage du corps de la réponse**
Content-Language	**Type de langage du corps de la réponse**
Content-Length	**Longueur du corps de la réponse**
Content-Type	**Type de contenu du corps de la réponse (par exemple *text/html*).**
Date	**Date de début de transfert des données**
Expires	**Date limite de consommation des données**
Forwarded	**Utilisé par les machines intermédiaires entre le browser et le serveur**
Location	**Redirection vers une nouvelle URL associée au document**
Server	**Caractéristiques du serveur ayant envoyé la réponse**

Code réponse HTTP

Code	Message	Description
10x	**Message d'information**	**Ces codes ne sont pas utilisés dans la version 1.0 du protocole**
20x	**Réussite**	**Ces codes indiquent le bon déroulement de la transaction**
200	**OK**	**La requête a été accomplie correctement**
201	**CREATED**	**Elle suit une commande POST, elle indique la réussite, le corps du reste du document est sensé indiquer l'URL à laquelle le document nouvellement créé devrait se trouver.**
202	**ACCEPTED**	**La requête a été acceptée, mais la procédure qui suit n'a pas été accomplie**
203	**PARTIAL INFORMATION**	**Lorsque ce code est reçu en réponse à une commande GET, cela indique que la réponse n'est pas complète.**
204	**NO RESPONSE**	**Le serveur a reçu la requête mais il n'y a pas d'information à renvoyer**
205	**RESET CONTENT**	**Le serveur indique au navigateur de supprimer le contenu des champs d'un formulaire**
206	**PARTIAL CONTENT**	**Il s'agit d'une réponse à une requête comportant l'en-tête *range*. Le serveur doit indiquer l'en-tête *content-Range***

Code réponse HTTP

Code	Message	Description
30x	**Redirection**	**Ces codes indiquent que la ressource n'est plus à l'emplacement indiqué**
301	**MOVED**	**Les données demandées ont été transférées à une nouvelle adresse**
302	**FOUND**	**Les données demandées sont à une nouvelle URL, mais ont cependant peut-être été déplacées depuis...**
303	**METHOD**	**Cela implique que le client doit essayer une nouvelle adresse, en essayant de préférence une autre méthode que GET**
304	**NOT MODIFIED**	**Si le client a effectué une commande GET conditionnelle (en demandant si le document a été modifié depuis la dernière fois) et que le document n'a pas été modifié il renvoie ce code.**

Code réponse HTTP

Code	Message	Description
40x	**Erreur due au client**	**Ces codes indiquent que la requête est incorrecte**
400	**BAD REQUEST**	**La syntaxe de la requête est mal formulée ou est impossible à satisfaire**
401	**UNAUTHORIZED**	**Le paramètre du message donne les spécifications des formes d'autorisation acceptables. Le client doit reformuler sa requête avec les bonnes données d'autorisation**
402	**PAYMENT REQUIRED**	**Le client doit reformuler sa demande avec les bonnes données de paiement**
403	**FORBIDDEN**	**L'accès à la ressource est tout simplement interdit**
404	**NOT FOUND**	**Classique! Le serveur n'a rien trouvé à l'adresse spécifiée.**

Code réponse HTTP

Code	Message	Description
50x	**Erreur due au serveur**	**Ces codes indiquent qu'il y a eu une erreur interne du serveur**
500	**INTERNAL ERROR**	**Le serveur a rencontré une condition inattendue qui l'a empêché de donner suite à la demande (comme quoi il leur en arrive des trucs aux serveurs...)**
501	**NOT IMPLEMENTED**	**Le serveur ne supporte pas le service demandé (on ne peut pas tout savoir faire...)**
502	**BAD GATEWAY**	**Le serveur a reçu une réponse invalide de la part du serveur auquel il essayait d'accéder en agissant comme une passerelle ou un proxy**
503	**SERVICE UNAVAILABLE**	**Le serveur ne peut pas vous répondre à l'instant présent, car le trafic est trop dense (toutes les lignes de votre correspondant sont occupées veuillez rappeler ultérieurement)**
504	**GATEWAY TIMEOUT**	**La réponse du serveur a été trop longue vis-à-vis du temps pendant lequel la passerelle était préparée à l'attendre (le temps qui vous était imparti est maintenant écoulé...)**

Langage HTML

- Langage à balises :
  - Une balise : une instruction de mise en forme
  - Une balise : contient des textes et/ou des autres balises autorisées
  - Une balise : structure le texte
  - Avant : une balise : mise en forme
- Navigateur : recherche des balises et interprétation
  - <NOM_DE_BALISE>….</NOM_DE_BALISE>
  - & (caractères spéciaux) : "&#code ascii;". Ex :
    - il y a : & /& ; < / < ; > / >
    - é / é (note les caractères s'obtiennent par la &+lettre+accent; les accents étant acute, grave, circ, cedil, tilde et aussi les ligatures avec lig, e.g. æ / æ).
- Liens hypertexte :
  - Chaque document HTML peut être représenté par un URL comme une ressource sur Internet.
  - Un document HTML peut contenir des liens vers les autres documents HTML ou les autres sources de données (multimédia, programme)
- Outil de validation : http://validator.w3.org

Fichier HTML minimum

- Version html 4.01

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

<HTML>
        <HEAD>
                <TITLE>
                </TITLE>
        </HEAD>

        <BODY>
        </BODY>

</HTML>
```

DTD

- Notion liée à XML (eXtensible Markup Language)
- DTD (*Document Type Definition*) : vérifier qu'un document XML est conforme à une syntaxe donnée (modèle).
  - une grammaire
  - document valide par rapport à une DTD
- Une DTD définie de 2 façons :
  - sous forme interne (inclure la grammaire dans le document)
  - sous forme externe (fichier local ou URL contenant la grammaire)

DTD

- **Définition d'un élément suivant la syntaxe : <! ELEMENT Nom Modèle >**
- **Modèle**
  - **ANY : L'élément peut contenir tout type de données**
  - **EMPTY : L'élément ne contient pas de données spécifiques**
  - **#PCDATA : L'élément doit contenir une chaîne de caractères**
    - **Le mot clé *#PCDATA* doit nécessairement être écrit entre parenthèses, sinon risque d'obtenir une erreur du parseur.**

Opérateur	Signification	Exemple
+	**L'élément doit être présent au minimum une fois**	**A+**
*	**L'élément peut être présent plusieurs fois (ou aucune)**	**A***
?	**L'élément peut être optionnellement présent**	**A?**
|	**L'élément A ou l'élément B peuvent être présents**	**A|B**
,	**L'élément A doit être présent et suivi de l'élément B**	**A,B**
()	**Les parenthèses permettent de regrouper des éléments afin de leur appliquer les autres opérateurs**	**(A,B)+**

Exemple de DTD

- <!ELEMENT personne (nom,prenom,telephone),email? >
- <!ELEMENT nom (#PCDATA) >
- <!ELEMENT prenom (#PCDATA) >
- <!ELEMENT telephone (#PCDATA) >
- <!ELEMENT email (#PCDATA) >

- <personne>
  - <nom>Renevier</nom>
  - <prenom>Philippe</prenom>
  - <telephone>04….</telephone>
  - <email>Philippe.Renevier@unice.fr</email>
- </personne>

DTD

- Attribut : <! ATTLIST Elément Attribut Type >
- Type représente le type de donnée de l'attribut, il en existe trois:
  - littéral: une chaîne de caractères, mot clé *CDATA*
  - l'énumération: une liste de valeurs possibles pour limiter le choix de l'utilisateur. Syntaxe
    - <! ATTLIST Elément Attribut (Valeur1 | Valeur2 | ...) >
    - <! ATTLIST Elément Attribut (Valeur1 | Valeur2) "valeur par défaut" > (valeur par défaut entre guillemets)
  - atomique: identifiant unique, mot clé *ID*.
- Caractère obligatoire d'un attribut (optionnel) : le faire suivre d'un mot clé particulier :
  - #IMPLIED : optionnel
  - #REQUIRED : obligatoire
  - #FIXED : valeur par défaut (à préciser entre guillemets) sinon défini.

- <! ATTLIST disque IDdisk ID #REQUIRED type (K7|MiniDisc|Vinyl|CD) "CD" >

DTD

- Entités : déclarer un groupe d'éléments sous un nom afin de ne pas avoir à réécrire ces derniers plusieurs fois dans la DTD
  - une meilleure lisibilité
  - un contrôle accru sur le contenu
  - une plus grande facilité de mise à jour
- On distingue plusieurs types d'entités dans XML :
  - les entités générales
    - <!ENTITY nom_de_l_entite "Contenu de l'entite">
    - <!ENTITY site "http://deptinfo.unice.fr/~renevier/L2">
    - usage : <site>&site;</site>
  - les entités paramètres
    - <!ENTITY % nom_de_l_entite definition>
  - les entités caractères
    - & : & < : < > : > ' : ‘ " : «
    - <!ENTITY nom_de_l_entite "&#xCODEHEXA;">
    - <!ENTITY ccedille "ç">

Déclaration de DTD HTML 4.01

- HTML 4.01 strict DTD : tous les éléments et attributs déclarés et non dépréciés (deprecated) et qui ne sont pas lié au frameset.
  - <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
- HTML 4.01 transitional DTD : strict DTD + deprecated (présentation visuelle).
  - <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
- HTML 4.01 Frameset DTD : transitional DTD + frame.
  - <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">

Langage HTML

- BALISES DE BASE
  - Format général de balise :
    - Les balises conteneurs : les balises contiennent de textes et des balises autorisées. Une balise conteneur est utilisée pour présenter ou structurer le texte et les balises qui lui appartiennent.

    Exemple : <H1> Mon document </H1> <!-- titre -->
    - Les balises vides : qui 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"} />

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

Propriétés des Balises html

- Propriétés générales (mais pas systématiques)

- id, class (document-wide identifiers)
- lang (language information), dir (text direction)
- title (element title)
- style (inline style information)

- onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup (intrinsic events)

xhtml

- Version « xml » de HTML
- Pour écrire un document xhtml (valide) à partir d'un document html 4 valide :
  - bien fermer les balises (ou tags)
  - document soit bien structuré (il faut bien fermer les balises dans le bon ordre)
  - balises et attributs en minuscule (sensible à la casse).
  - fermer les éléments vides (e.g.
)
  - valeurs des attributs entre guillemets et chaque attribut a une valeur
    - (la minimisation n'est pas autorisée : <input checked="checked" />)
  - généralement l'attribut name est à remplacer par l'attribut id

Fichier xhtml minimal

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "DTD/xhtml1-
    strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
 <head>
    <title>titre de la page</title>
</head>

<body>
</body>

</html>
```

Les autres déclarations de dtd xhtml

- <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd" >

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

- <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">

Balises HTML / xhtml

- Balises structurelles
- Balises contenant du texte
- Balise hypertext
- Balises de listes
- Balises de mise en forme (dépréciées !!)
- Balises de table (tableaux)
- Balises de formulaire
- Balises d'objets, images et applets
  - les images cliquables
- Balises de Frame
- Balises Meta (entêtes)

Balises Structurelles

- html : définition du document
  - un head, un body
  - *Start tag:* ***optional****, End tag:* ***optional***
- head : partie comprenant les entêtes et meta-données du document
  - Un TITLE obligatoire
  - META, SCRIPT, STYLE, etc. optionnelles
  - *Start tag:* ***optional****, End tag:* ***optional***
- title : titre du document
  - Titre de la page (en haut du navigateur)
  - Contient du texte
  - *Start tag:* ***required****, End tag:* ***required***
- body : corps du document
  - *Start tag:* ***optional****, End tag:* ***optional***
  - Partie centrale du navigateur
  - *« block »* ou script
  - ins et del
  - **« block » : p | h1 | h2 | h3 | h4 | h5 | h6 | ol | ul | dl | pre | div | noscript | blockquote | form | hr | table | fieldset | address**
  - Attribut spécifique : onload, onunload

Balises Structurelles (regroupement)

- div
  - « flow » : « block » ou « inline »
  - « **inline** » :
    - texte (PCDATA)
    - phrase : em | strong | dfn | code | samp | kbd | var | cite | abbr | acronym
    - special : a | img | object | br | script | map | q | sub | sup | span | bdo
    - formulaire : input | select | textarea | label | button
    - inline inclut aussi fontstyle : tt | i | b | big | small (**depreciated**)
  - *Start tag:* ***required****, End tag:* ***required***
- span
  - « inline »
  - *Start tag:* ***required****, End tag:* ***required***
- ne font que regroupé : partage de style, dhtml, etc.

Balises Structurelles (titres)

- h1 | h2 | h3 | h4 | h5 | h6
  - « inline »
  - Rendu par défaut dépendant du navigateur et de la plate-forme
  - *Start tag:* ***required****, End tag:* ***required***
- Plus l'indice est petit, plus le titre est important

- En prime : adress
  - « inline »
  - *Start tag:* ***required****, End tag:* ***required***

- exemple/baliseH.html

Balises Textuelles (phrases)

- *em - emphasis*, **strong - stronger emphasis**, *dfn - définition*, code - bout de code, samp - sample, kbd - keyboard : texte à entrer par l'utilisateur, *var - variable d'un programme*, *cite - citation ou une référence*, abbr - abbréviation, acronym
  - phrase ou fontstyle
  - *start tag:* ***required****, end tag:* ***required***

- exemple/balisePhrase.html

Balises Textuelles (quotations et indices)

- blockquote
  - « block » ou script
  - « block level » ; citation longue
  - attribut spécial cite (uri)
  - *start tag:* ***required****, end tag:* ***required***
- q
  - « inline » (citation courte)
  - attribut spécial cite (uri)
  - *start tag:* ***required****, end tag:* ***required***
- indice sub, exposant : sup
  - « inline »
  - *start tag:* ***required****, end tag:* ***required***

Balises Textuelles (lignes et paragraphes)

- p (paragraph)
  - « inline » : pas de <p> dans un <p> !!
  - « block level »
  - *Start tag:* ***required****, End tag:* ***optional***
- br (line break)
  - vide !
  - note : pour un espace non « cassable »   (ou   or) : non breakable space…
  - *Start tag:* ***required****, End tag:* ***forbidden***
- pre (texte pré-formaté)
  - espace et retours à la ligne conservé
  - « inline » sauf IMG | OBJECT |BIG | SMALL | SUB | SUP
  - *Start tag:* ***required****, End tag:* ***required***

- exemple/baliseP.html

Balises Textuelles (changement)

- ins et del
  - flow (block ou inline)
  - spéciaux : tantôt block-level tantôt inline (mais pas les deux)
  - *Start tag:* ***required****, End tag:* ***required***
  - Attribut : cite (uri) et dateTime
    - YYYY-MM-DD**T**hh:mm:ssTZD : 1994-11-05**T**08:15:30-05:00
      - YYYY = four-digit year : 1994
      - MM = two-digit month (01=January, etc.)
      - DD = two-digit day of month (01 through 31)
      - hh = two digits of hour (00 through 23) (am/pm NOT allowed)
      - mm = two digits of minute (00 through 59)
      - ss = two digits of second (00 through 59)
      - TZD = time zone designator : -05:00 écart avec GMT (côte est US)

Balise hypertext

- balise anchor (ancre) : a.
  - référencer une page par son URI
    - ….
    - *Start tag:* ***required****, End tag:* ***required***
  - définir une nouvelle attache pour être la cible d'autre lien
    - balise vide
    - (id ou name) ou <a…> …
    - référence : href= "*URI*#nom de l'ancre« (lien local si URI ="")
  - « inline » excepté a
  - Attributs
    - name (id)
    - href (uri)
    - hreflang = langcode : langue de la destination, seulement quand il y a href
    - type = content-type (MIMETYPE : ftp://ftp.isi.edu/in-notes/iana/assignments/media-types/.)
    - rel = relation entre le document courant et la cible (quand href)
      - **Alternate, Stylesheet, Start, Next, Prev, Contents, Index, Glossary, Copyright, Chapter, Section, Subsection, Appendix ,Help, Bookmark**
    - rev = idem rel (dans le sens inverse)
    - charset = encodage de la cible

Balise Listes

- ul (non ordonnée) ou ol (ordonnée)
  - (LI)+
  - *Start tag:* ***required****, End tag:* ***required***
  - Attributs dépréciés : type (1, A, a, i, I) ; start (ol : une valeur de début) ; compact
- li (list item)
  - « Flow »
  - *Start tag:* ***required****, End tag:* ***optional***
  - Attributs dépréciés : type, value (une valeur), compact
- dl (Définition list)
  - (dt | dd)+
  - *Start tag:* ***required****, End tag:* ***required***
- dt (term) ou dd (définition)
  - dt : « inline » et dd : « flow »
  - *Start tag:* ***required****, End tag:* ***optional***

- exemple/baliseList.html

Balise hr

- Séparation horizontale
  - "block level"
- Balise vide
  - *Start tag:* ***required****, End tag:* ***forbidden***
- Attributs (tous dépréciés)
  - noshade (pas de valeur) : pas d'ombre
  - size (nombre de pixels) : épaisseur du trait
  - width (nombre de pixels ou poucentage) : longueur du trait

Balises de mise en forme (dépréciées !!)

<B> …</B>	Gras
<I>…</I>	Italique
…	Centrer les éléments (textes, images)
<BIG>…</BIG>	En plus gros
<SMALL>…</SMALL>	En plus petit
<S>…</S> ou <STRIKE></STRIKE>	Texte rayé
<U>…</U>	Texte souligné
<TT>…</TT>	Style teletype (courrier)
<BLINK>…</BLINK>	Texte clignotant

- **Attributs dépréciés !**
  - **align (center, left, right, justify)**
  - **bgcolor (couleur de fond)**

Balises de mise en forme (dépréciées !!)

- 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. Sept tailles sont disponibles et la valeur par défaut est 3, la plus grosse étant de valeur 6
- Une partie de texte peut voir sa taille modifiée si elle est entourée de la balise : <FONT SIZE=valeur> ..</FONT>
  - 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
  - <FONT COLOR="#FF0000">permet de colorier les caractères (ici en rouge)</FONT> donne comme résultat :

permet de colorier les caractères (ici en rouge)

Balises Table

- table
  - (caption?, (col*|colgroup*), thead?, tfoot?, tbody+)
  - Attributs
    - summary (texte)
    - align (déprécié, left,center ou right)
    - width (largeur en pixel ou pourcentage)
    - cellspacing (pixel ou %) -- spacing between cells --
    - cellpadding (pixel ou %) -- spacing within cells --
    - frame = void|above|below|hsides|lhs|rhs|vsides|box|border (cadre autour du tableau):
      - void: No sides. This is the default value.
      - above: The top side only.
      - below: The bottom side only.
      - hsides: The top and bottom sides only.
      - vsides: The right and left sides only.
      - lhs: The left-hand side only.
      - rhs: The right-hand side only.
      - box: All four sides.
      - border: All four sides.

Balises Table

- table
  - Attributs (suite)
    - rules = none|groups|rows|cols|all (les traits internes)
      - none: No rules. This is the default value.
      - groups: Rules will appear between row groups and column groups only.
      - rows: Rules will appear between rows only.
      - cols: Rules will appear between columns only.
      - all: Rules will appear between all rows and columns.
    - border = pixel (épaisseur du trait externe)
  - Processus d'affichage
    - Résumé
    - Titre
    - Entêtes et pieds
    - Calcul du nombre de colonne (nombre de ligne donné par nombre de TR)
    - Rendu des cellules
  - *Start tag:* ***required****, End tag:* ***required***
- caption (titre)
  - inline
  - *Start tag:* ***required****, End tag:* ***required***

Balises Table

- thead (entêtes) et tfoot (pieds) et tbody (corps)
  - tr+
  - même nombre de colonne
  - ordre d'apparition
  - tbody toujours requis sauf si unique et pas de thead ni de tfoot
  - *start tag:* ***required****, end tag:* ***optional***
- colgroup
  - col*
  - regroupement de colonnes
  - attribut : span (nombre) width (% ou relative (poids))
  - *start tag:* ***required****, end tag:* ***optional***
- col
  - vide
  - regroupement d'attributs pour des colonnes
  - attribut : span (nombre) width (% ou relative (poids))
  - *start tag:* ***required****, end tag:* ***forbidden***

Balise Table

- tr (ligne)
  - (th | td)+
  - *start tag:* ***required****, end tag:* ***optional***
- th (titre) ou td (data) : les cellules
  - « flow »
  - attribut
    - rowspan = *nombre de ligne*
    - colspan = *nombre de colonne*
  - *start tag:* ***required****, end tag:* ***optional***

- exemple/baliseTable.html

Balises de formulaire : fieldset

- Mise en forme des formulaires
  - Cadre
  - Utilisable en dehors des formulaires
- fieldset
  - Des espaces (blancs), Une legend, (flow)*
  - *Start tag:* ***required****, End tag:* ***required***
- legend
  - « inline »

Balises de formulaire : form

- Contient des éléments de contrôle de formulaire (bouton, champs, etc.)
  - « block » (sauf form) ou script
  - Attributs
    - action (uri)
    - method ("get" ou "post")
      - get : envoi dans l'url des paires key/value : ?toto=val&titi=val2&…
      - post : envoi
    - enctype (pour une méthode "post")
      - Par défaut : **application/x-www-form-urlencoded -** encodage : espace devient + et les autres non alphanumériques %HH et les retours à la ligne : "CR LF" (i.e., `%0D%0A')
      - **multipart/form-data -** envoi en différentes parties (types à préciser à la source)
    - accept-charset (liste - , - d'encodage possible pour les caractères acceptés par le server)
    - Accept (liste - , - de types de contenu acceptés par le server)
    - events : onsubmit et onreset
  - *Start tag:* ***required****, End tag:* ***required***

Balises de formulaire : input

- Définit selon un type :
  - text : champs d'entrée de texte.
  - password : l'écho sont des '*'. sécutité pauvre.
  - checkbox
  - radio (radiobutton)
  - submit : un bouton pour envoyer
  - image : un bouton submit graphique. Attribut src donne l'URI de l'image. Utiliser l'attribut alt. Les coordonnées du clic sont passés au server sous la forme name.x et name.y ; mais à ne pas trop utiliser
    - problème d'accéssibilité : navigateur non graphique, clic difficile, etc.
    - à remplacer par plusieurs boutons submit ou par des scripts côtés client.
  - reset (bouton).
  - button : bouton sans comportement prédéfini (script)
  - hidden : champs caché (parfois utile pour passer une valeur masquée)
  - file : sélection d'un fichier

Balises de formulaire : input

- Balise vide
  - *Start tag:* ***required****, End tag:* ***forbidden***
- Attributs
  - type
  - name : nom de contrôle (très important)
  - value (valeur initiale ou libellé) : optionnel sauf pour radio et checkbox
  - size (en pixel sauf pour text et password où c'est un nombre de caractère)
  - maxlength : pour text ou password : nombre de caractères maximum
  - checked : pour radio et checkbox
  - src : pour image : la source (ne pas oublier alt)
- exemple/baliseInput.html

Balises de formulaire : select

- select : menu
  - (optgroup | option)+
  - attributs
    - name : nom de contrôle
    - size (nombre) : nombre d'éléments visibles pour une scroll list
    - multiple (pas de valeur) : permet la sélection multiple
  - *start tag:* ***required****, end tag:* ***required***
- option
  - #pcdata (texte)
  - attributs
    - selected : pour présélectionner l'élément
    - value (texte) : pour donner une valeur autre que le texte (#pcdata)
    - label (texte) : pour faire apparaître un autre nom (plus court) à la charge du navigateur !! (pas sûr que cela fonctionne !!)
  - *start tag:* ***required****, end tag:* ***optional***
- optgroup
  - regrouper les options : (option)+
  - attribut : label (texte) : libellé
  - *start tag:* ***required****, end tag:* ***required***
- exemple/baliseSelect.html

Balises de formulaire : textarea

- Champs d'entrée sur plusieurs lignes
- textarea
  - #PCDATA : texte initiale
  - Attributs
    - name : nom de contrôle
    - cols : nombre de colonne
    - rows : nombre de ligne
  - *Start tag:* ***required****, End tag:* ***required***

Balises de formulaire : label

- Permet d'associer un texte à un élément de formulaire sans texte
  - inline
  - *Start tag:* ***required****, End tag:* ***required***
- Attaché par l'attribut for
  - Valeur = id d'un champ de contrôle

Balises de formulaire : navigation

- Transfert de focus
  - Souris
  - Clavier : tabulation ou touche raccourcie
- Tabulation : attribut tabindex (numéro)
  - a, area, button, input, object, select et textarea.
  - 1) Ceux qui ont l'attribut : de la plus petite à la plus grande valeur. Pas forcément consécutifs. En cas d'égalité, ordre d'apparition (flots de caractère)
  - 2) Ceux qui ne n'ont pas (impossible ou non donné) : ordre d'apparition
  - 3) Les désactivés (c.f. transparent suivant) ne participent pas.
- Touche raccourcie : attribut (un caractère)
  - a, area, button, input, label, legend et textarea.
  - sous windows : besoin de la touche alt en plus…

- exemple/baliseNav.html

Balises de formulaires : attributs

- Attributs des contrôles
  - readonly (pas de valeur)
    - input et textarea
    - élément non modifiable mais inclus dans la navigation
  - disabled (pas de valeur)
    - button, input, optgroup, option, select, et textarea.
    - élément exclu de la navigation
    - élément grisé

Balises d'objets, images et applets

- Attributs communs à img, object et applet
  - width (nombre) : largeur en pixel
  - height (nombre) : hauteur en pixel
  - alt (texte) : texte alternatif
    - Ne pas donner de texte si l'image est décorative (formatage : une puce)
    - Ne pas remplir avec un texte « bidon »
- IMG
  - Balise vide
  - Dans inline
  - Attributs
    - src (uri) : source de l'image
    - alt : obligatoire
    - longdesc (uri) pour donner des informations plus conséquentes qu'avec alt.
  - Redimensionnement !!
  - *Start tag:* ***required****, End tag:* ***forbidden***

Balises d'objets, images… : object

- object : inclusion générique
- Besoin éventuel de plugin (ex: une applet)
- (param | flow)*
- Attributs
  - Codebase (uri): chemin de référence, qui complète les chemins relatifs des attributs classid, data et archive. Par défaut : la position du document
  - data (uri) : donne la localisation de l'objet
  - classid (uri) : précise la position d'une implémentation de l'objet. Alternative ou complément à data (selon objet)
  - archive (uri-list) : listes d'URIs séparés par des espaces. Précise des archives contenant des éléments utilisés par l'objet. Permet des les précharger.
  - type (content-type) : précise le type de contenu de l'objet spécifié par data. Optionnel mais permet au navigateur de ne pas charger pour rien un objet non supporté.
  - codetype (content-type) : précise le type de contenu de l'objet spécifié par classid. Optionnel mais permet au navigateur de ne pas charger pour rien un objet non supporté. (par défaut égal à type)
  - declare (pas de valeur) : Lorsqu'il est présent, le booléen inidque que l'objet doit être instancié.
  - standby (text) : message à afficher pendant le chargement

Balises d'objets, images… : param

- Spécifie un paramètre sous formes key / value
- Balise vide
  - *Start tag:* ***required****, End tag:* ***forbidden***
- Attributs
  - name (texte) : nom du paramètre (sensible à casse selon l'objet)
  - value (texte) : valeur associée à un nom. Interprétation par l'objet
  - valuetype (data | ref | object)
    - data (valeur par défaut) : valeur passée comme chaîne de caractères
    - ref : c'est une URI
    - object : référence à un autre objet du document (id)
  - Type (content-type) : précise le type des ressources en cas de valuetype="ref"

Balises d'objets… : Exemples du W3C

```
<P><OBJECT classid="http://www.miamachina.it/analogclock.py">
    <PARAM name="height" value="40" valuetype="data">
    <PARAM name="width" value="40" valuetype="data">
    This user agent cannot render Python applications.
</OBJECT> </P>

<!-- First, try the Python applet -->
<P><OBJECT title="The Earth as seen from space"
    classid="http://www.observer.mars/TheEarth.py">
    <!-- Else, try the MPEG video -->
    <OBJECT data="TheEarth.mpeg" type="application/mpeg">
            <!-- Else, try the GIF image -->
            <OBJECT data="TheEarth.gif" type="image/gif">
            <!-- Else render the text --> The <STRONG>Earth</STRONG> as seen from
            space.
            </OBJECT>
    </OBJECT>
</OBJECT> </P>
```

Balises d’objets, images… : APPLET

- Balise APPLET déprécié
- Utilisation de OBJECT
- public String getParameter(String name)
- Exemples (pas de codebase : les .class sont au même endroit) :

```
<P><OBJECT codetype="application/java" classid="java:Bubbles.class"
    width="500" height="500">
Java applet that draws animated bubbles.
</OBJECT></P>

<OBJECT codetype="application/java" classid="AudioItem" width="15"
    height="15">
<PARAM name="snd" value="Hello.au|Welcome.au"> Java applet that plays a
    welcoming sound.
</OBJECT>
```

Images Cliquables

- des images contenant des zones servant de liens
- xhtml 1.0
- 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 l'attribut USEMAP
- Deux types d'images cliquables
  - Côté client (le plus fréquent et préférable) : traitement par le navigateur
  - Côté serveur (c.f. plus loin) : traitement sur le serveur

Images Cliquables : map

- map
  - (block | area)+
  - Désigné dans une balise IMG l'attribut usemap…
  - qui fait référence à l'attribut name (texte) de cette MAP
  - Attribut id obligatoire (et identique à name)
  - *Start tag:* ***required****, End tag:* ***required***
- area
  - Balise vide
  - Attributs
    - shape = default|rect|circle|poly
    - coords = coordinates
      - rect: left-x, top-y, right-x, bottom-y.
      - circle: center-x, center-y, radius (pixel ou %)
      - poly: x1, y1, x2, y2, ..., xN, yN. (x1=xN et y1=yN)
    - nohref (pas de valeur)
- exemple/baliseMap.html

Images Cliquables : côté serveur

- Possib le uniquement pour img et input.
  - IMG avec l'attribut ismap dans un A
  - INPUT de type "image".
- Un clic sur la zone provoque l'envoie des coordonnées
- Exemple :
  - <P><A href="http://www.acme.com/cgi-bin/competition">
    <IMG src="game.gif" ismap alt="target"></A></P>
- Envoie des coordonnées du clic : URI du A+`?' + x et y, séparés par une virgule
  - http://www.acme.com/cgi-bin/competition?10,27

Balises de Frame

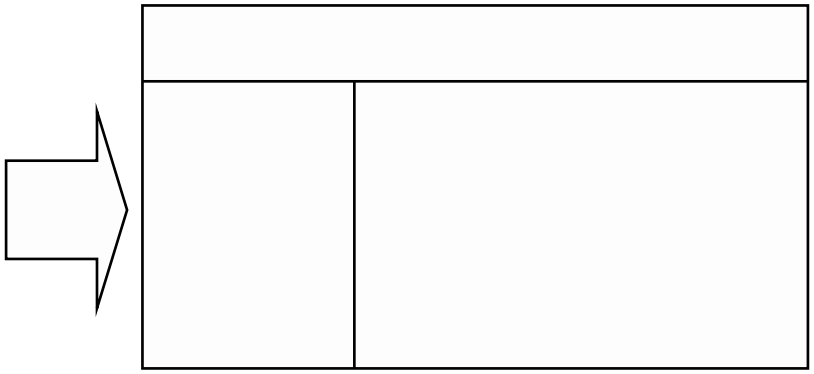
- Balises FRAMESET et FRAME permettent de définir des fenêtres dans une page HTML, chacune de ces fenêtres se comportant comme une page HTML indépendante. Ainsi on peut disposer d'un ascenseur pour chacune de ces zones, faire défiler des textes ou des images dans chacune d'entre elles.
- Les frames mettent 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. 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
  - la balise FRAME définissant le contenu des fenêtres au niveau feuille de la hiérarchie de fenêtre.

Balises de Frame

- 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 *).

```
<FRAMESET ROWS=20%,*>
<FRAME …>
<FRAMESET COLS=30%,*>
<FRAME …>
<FRAME … >
</FRAMESET>
</FRAMESET>
```



Balises de Frame

- 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=*URI* : donne l'URI (valeur initiale) 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.
    - Frameborder (1 | 0)
    - longdesc = URI : pour indiquer une description longue

Balises de Frame

- Préciser la cible pour les éléments qui créent des liens (A, LINK), des images cliquable (AREA) et des formulaires (FORM).
- Attributs targets
- Frame target names
  - Sauf pour les noms suivants, les noms donnés doivent être alphabétique
  - Noms réservés
    - _blank : nouvelle fenêtre
    - _self : dans le même cadre
    - _parent : dans le frameset parent
    - _top : dans la même fenêtre (en remplacement des frames)

Balises de Frame

- Frame interne : IFRAME
  - flow
  - Attribut : longdesc, name, width, height, src, frameborder, marginwidth, marginheight, scrolling
  - Équivalent à OBJECT avec un fichier html

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

Balises Meta (entêtes)

- <meta> donne à votre document des informations qui seront lues par le serveur httpd. 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.
  - <meta http-equiv="Expires" content="Tue, 20 Aug 2006 14:25:27 GMT">
- <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
- <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)