

Web dynamique

Techniques, outils, applications

(Partie A)

SOMMAIRE

1. Introduction (25mn)
2. Les bases de données (10mn)
3. Serveurs HTTP (10mn)
4. Technologies de connexion (15mn)
5. Mise en œuvre de CGI (28 mn)
6. Présentation avec HTML (23 mn)
7. Les formulaires HTML (15mn)

1/7 - introduction

Objectifs de ce cours

De quoi s'agit il ?

- Le Web ne s'est vraiment développé qu'à partir du moment où il a proposé des contenus dynamiques
 - Exemples: publication d'un catalogue, commandes en ligne, services d'annuaires..
 - Le besoin étant de générer des pages Web en fonction d'entrées utilisateur
- Un contenu dynamique est nécessairement connecté à une base de données (dans son sens large)
- L'objet de ce cours est de couvrir les divers moyens de connecter les bases de données au Web

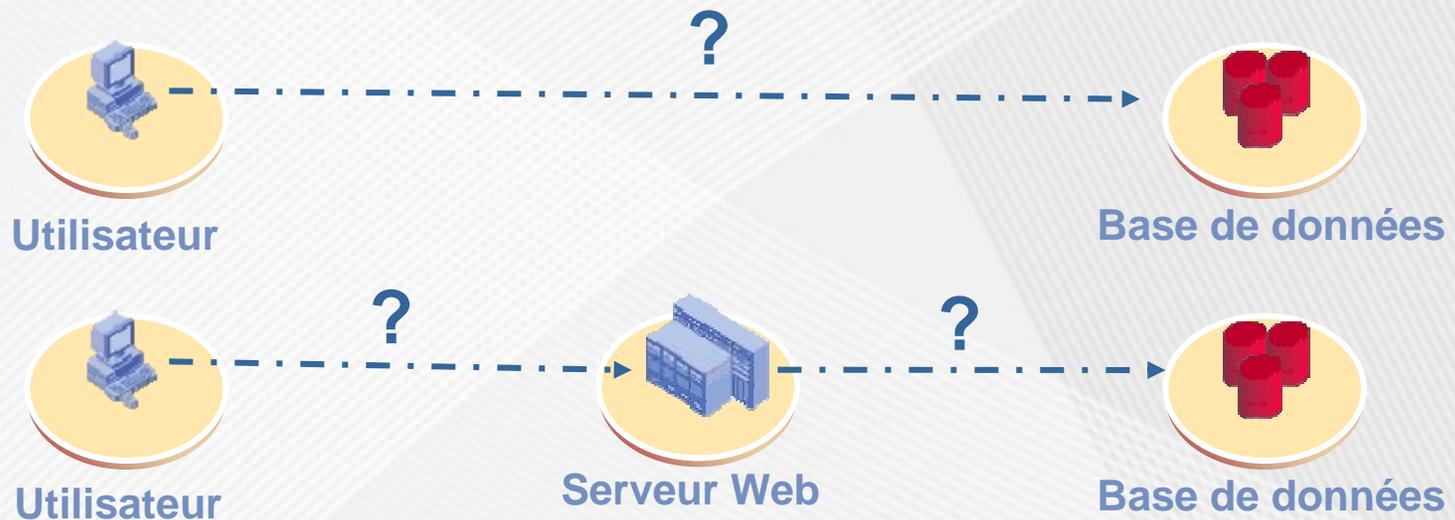
Objectifs de ce cours

1. Donner un éclairage sur les orientations multi-tiers de l'industrie
2. Etudier les divers moyens actuels de liaison du Web aux bases de données
 - La problématique
 - les Générations
 - L'Offre Actuelle
3. Etudier la chaîne de développement Web
 - Les particularités
 - Les difficultés

Connecter le Web et les SGBD

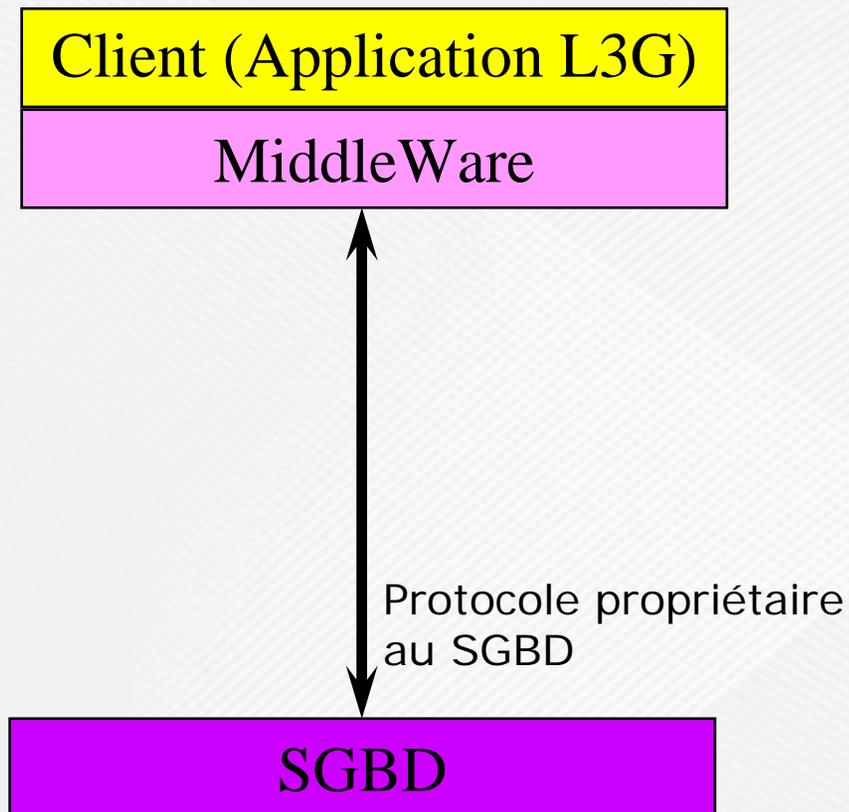
□ Les attraits du multi tiers

- Pourquoi l'industrie s'est orienté vers ces technos?
- Au point de les utiliser en interne (intranets)?
- Au point même de redévelopper des applications existantes?

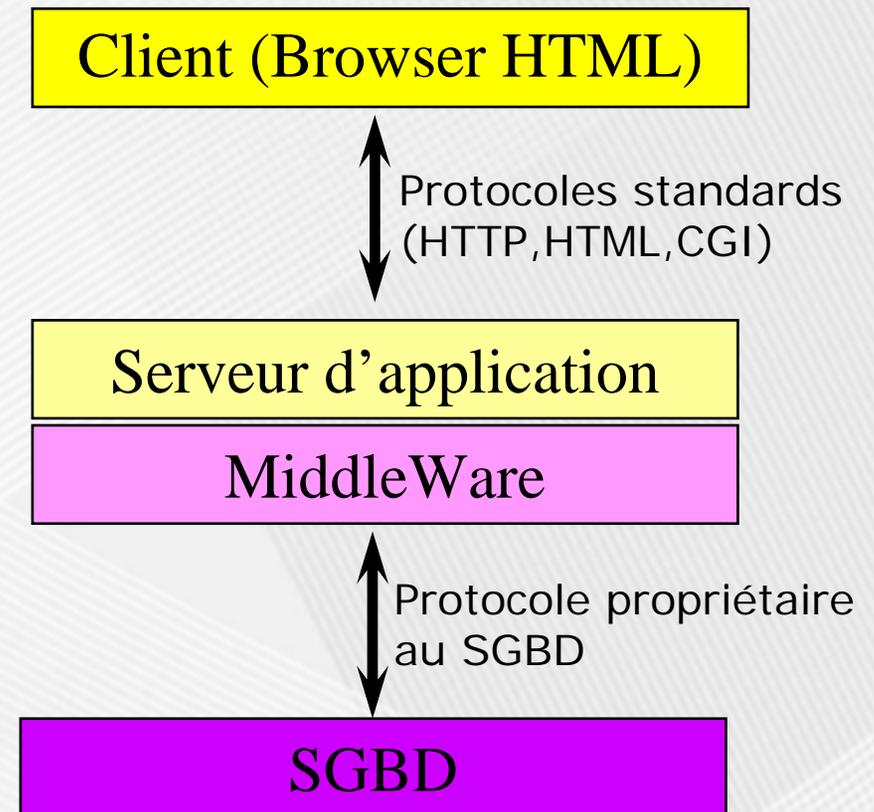


Modèles d'architectures

2 tiers

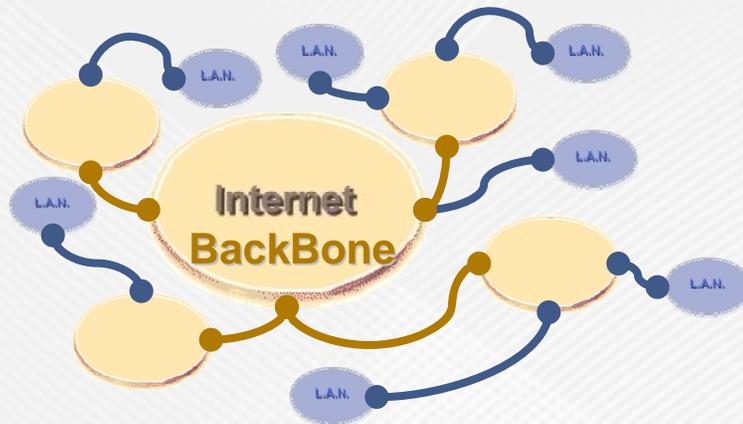


3 tiers



Internet

Internet: un Réseau



→ Comparable au réseau routier

- Protocoles associés: IP, TCP

Les services Internet

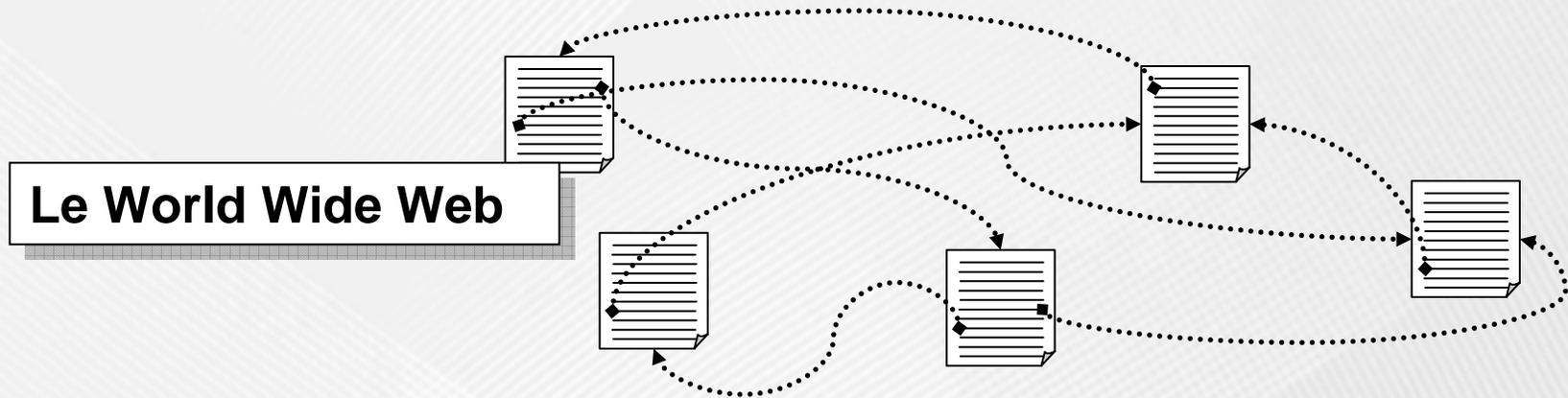
- Web
- News
- Mail
- RealAudio
- FTP
- P2P
- Telnet
- ...

→ Comparable aux services usuels tels que les livraison de marchandises, le Facteur, la Pizza à domicile...

- Protocoles associés:
 - spécifiques au service
 - Pas nécessairement standards

Le Web n'est qu'un service s'appuyant sur un réseau IP

Les protocoles associés au Web



- ❑ Le Web est Idéalement vu comme un réseau de documents (dont on ne connaît pas forcément l'emplacement)
- ❑ En pratique, vu comme un réseau de serveurs de pages au format HTML
- ❑ S'appuie sur: HTTP, URL, HTML, CGI

Le socle technique du Web

- Le Web s'appuie sur:
 - HTTP: le transport
 - URL: l'adressage des documents
 - HTML: la description du contenu et de sa mise en forme (on le voit plus loin)
 - CGI: la passerelle autorisant l'interactivité

- Technos utilisées de manière
 - Globale: World Wide Web
 - Locale: Intranet

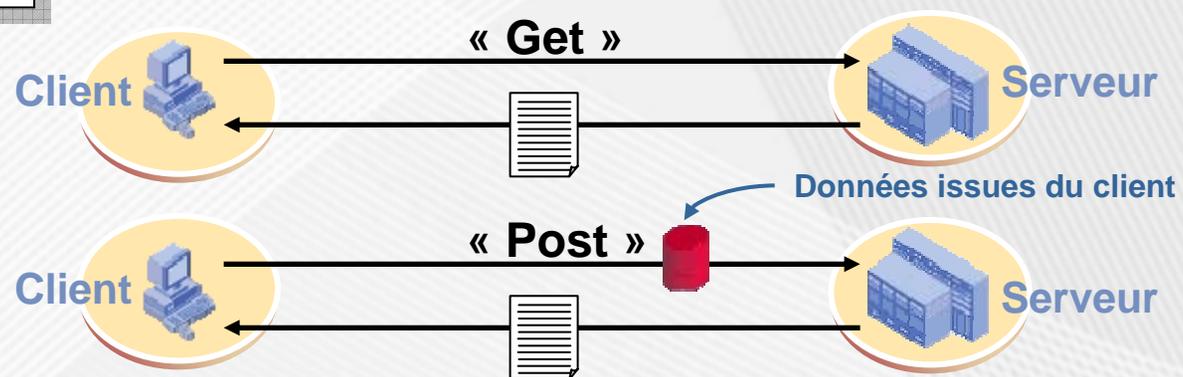
Le socle technique du Web

- URL (Uniform Resource Location) est un moyen de désignation universel de l'emplacement de certains fichiers.
 - Indépendant de l'OS
 - Indépendant de la machine
- Format général:
 - protocole://machine[:port]/[chemin_du_fichier]
- En particulier
 - protocole :ftp, http, autres (nntp, gopher..)
 - Machine: IP ou nom
 - Port est optionnel, 80 par défaut
 - Le chemin est optionnel, il doit normalement désigner un fichier (sinon on dit que l'URL est incomplète)

Le socle technique du Web

- HTTP: les principales méthodes
 - GET URL : demander le contenu de la ressource
 - POST URL: envoi de données vers une application
- HTTP: le transport
 - Architecture C/S, mode « Pull »
 - Connections courtes, « Sans état » (stateless)

Echanges HTTP



Exemple d'échange HTTP

Requête:

```
GET / HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; fr-FR; ...
Accept:
    text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/p
    lain;q=0.8,image/png,*/*;q=0.5
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
```

Demo avec
apache et un
proxy http

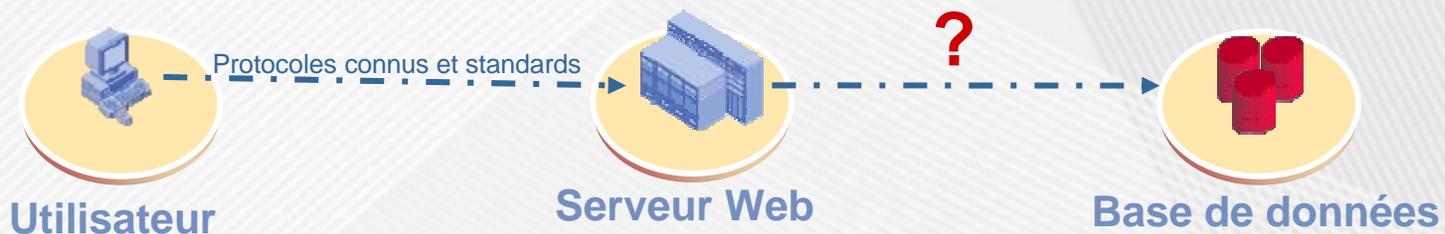
Réponse:

```
HTTP/1.1 200 OK
Date: Sun, 14 Aug 2005 15:10:14 GMT
Server: Apache/1.3.23 (Win32)
Content-Type: text/html
```

```
<HTML>
  <BODY><H1>Page</H1><BODY>
<HTML>
```

Connecter le Web et les SGBD

- Rappel (sur le premier slide!)
 1. Le Web ne s'est vraiment développé qu'à partir du moment où il a proposé des contenus dynamiques
 2. Un contenu dynamique est nécessairement connecté à une base de données (dans son sens large)
- La problématique
 - Le maintien de sessions transactionnelles
 - la performance
 - la sécurité
 - la réutilisabilité du code



2/7 - Les bases de données

Socle du multi-tiers

Idée de base (petit rappel)

- ❑ La majorité des applications nécessitent le stockage et la restitution d'information
- ❑ Un simple tableau constitue déjà une base de données!
- ❑ Si on veut une persistance pour les données, un seul moyen: les fichiers
- ❑ Ces applications s'appuient donc
 - Soit sur des fichiers au format propriétaire
 - Soit sur des fichiers respectant un standard
 - ...Soit délèguent à un composant externe la gestion de ces fichiers (un SGBD)

Bases de données

- ❑ Base de données (BD) : Collection de données cohérentes et structurées
- ❑ Système de Gestion de Bases de Données (SGBD) : Logiciel assurant la structuration, stockage, maintenance, mise à jour et consultation des données d'une BD
- ❑ Simplification :
 - base de données = grande quantité d'information
 - SGBD = logiciel pour les gérer



Les plus d'un SGBD

- Gestion des données
 - Problématique de gestions efficace des fichiers résolue (cf fichier séquentiels indexés, journaux...)
 - Garantie l'indépendance physique et logique de l'application
 - Avancées: Bases de données réparties et répliquées
- Manipulation des données hors application
 - Garantie l'efficacité des accès aux fichiers de données
 - Optimisation des requêtes pendant leur évaluation
- Maintien de l'intégrité des données
 - Cohérence des données (contraintes d'intégrités, domaines..)
 - Sécurité des données vis-à-vis des pannes
 - La gestion des transactions
- Partage des données
 - utilisation simultanée des données par différentes applications
 - contrôle des accès concurrents aux données, verrouillage
- Confidentialité des données
 - les données sont protégées contre les accès non-autorisés
 - hiérarchie des privilèges d'accès aux données

Principaux SGBD du marché

□ Commerciaux:

- Oracle (Windows et Unix)
- Microsoft SQL Server (Windows uniquement)
- IBM DB2 universal database (Windows et Unix)
- Autres: Informix, Sybase

□ Open source:

- Mysql
- PostgreSQL
- mSQL

3/7 - Serveurs HTTP

Serveurs Web

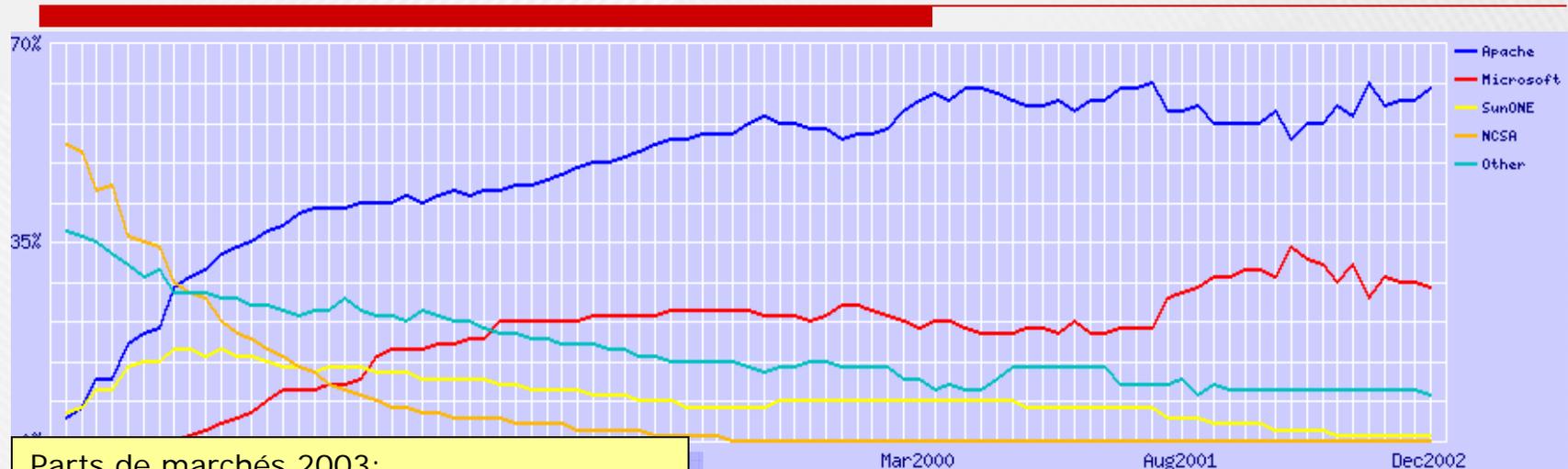
□ Principaux serveurs :

- Apache (GNU, Unix/NT notamment)
- Microsoft Internet Information Server (NT)

□ A propos de Apache

- Existe depuis 1995, à l'origine basé sur le serveur NCSA httpd 1.3 (Université de l'Illinois Etats-Unis)
- 1996: Apache devient le serveur Web le plus utilisé dans le monde
- Voir <http://www.apache.org>
- distribué sous une licence "Open source" (le code source est accessible à tout le monde)

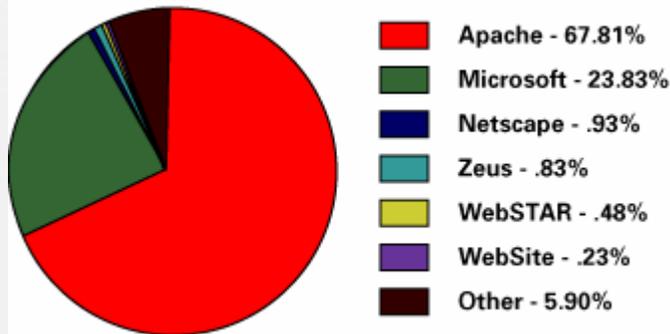
Serveurs Web en ligne



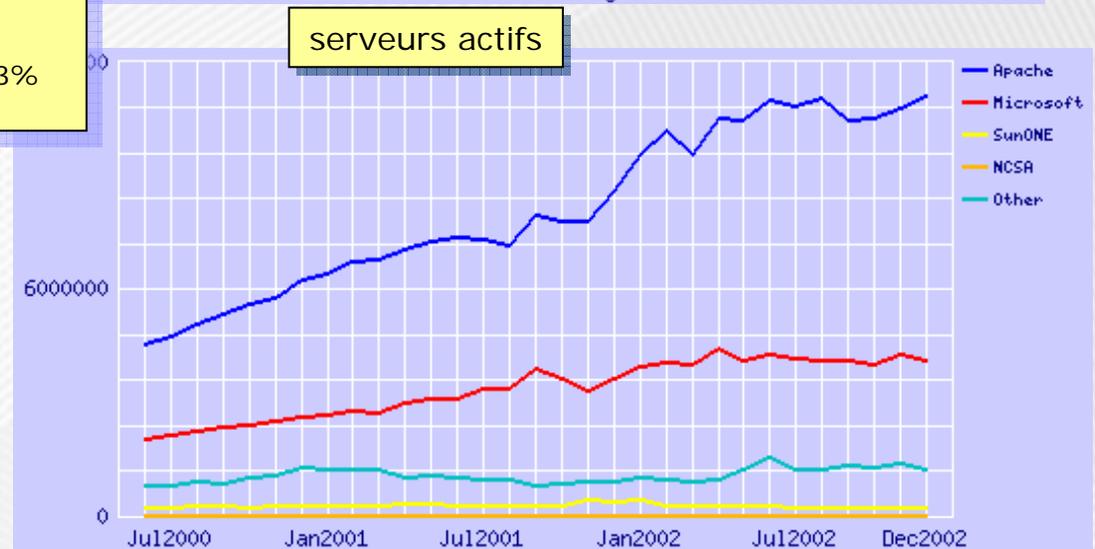
Parts de marchés 2003:

- Apache : 63%
- Microsoft Internet Information Server : 23%

Market Share for October 2003 — Across All Domains



Copyright 1998-2003 E-Soft Inc.



serveurs actifs

(source <http://www.netcraft.com/survey/>) et <http://www.securityspace.com>

Serveurs Web : Mise en place

Dénominateur commun :

Demo avec
apache servcd

- Répertoire root et Mapping
- Répertoires virtuels (« alias »)
- Page par défaut

Et:

- Listing automatique
 - Gestion des droits d'accès
-

Mise en place Sites Web

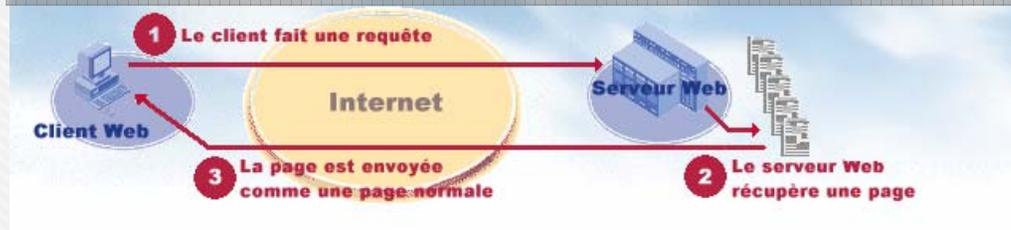
- Dans un sous-répertoire de l'arborescence du serveur
- Association d'un nom DNS au site
 - Sous-domaine
 - Site par défaut
 - « Virtual hosting »
- Mise à jour
 - Accès FTP (ouverture d'un compte FTP)
 - Accès filesystem (copie directe)
 - Extension serveur (Frontpage, webDAV)

4/7 - Technologies de connexion

Pour obtenir des contenus dynamiques

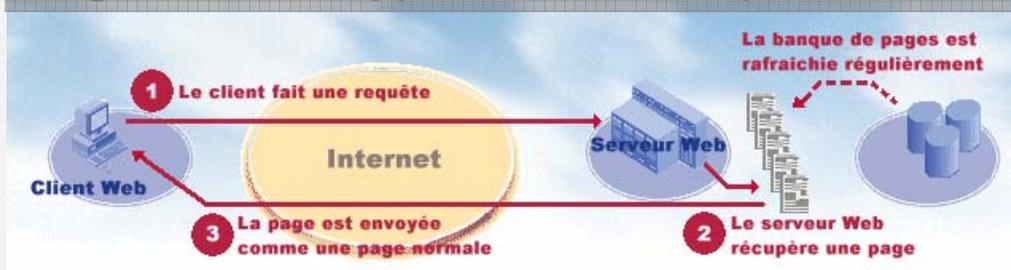
Evolution du Web

1e génération : contenus statiques



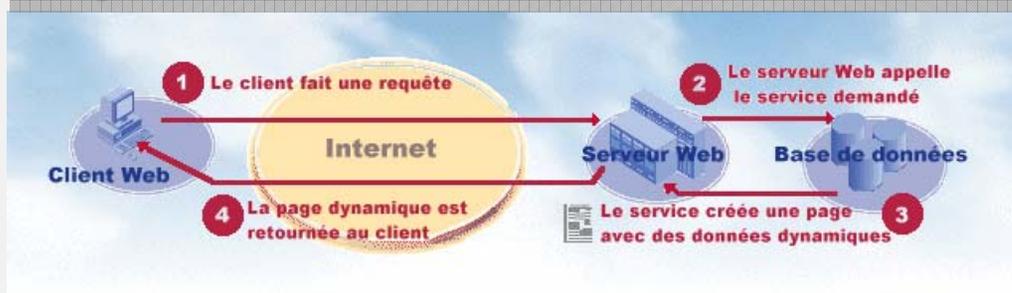
- Pages HTML statiques
- Publications de notes, documents, etc
- Simple, rapide mais figé
- Contenus ne dépendent pas du client mais de l'auteur

2e génération : Approche bibliothèque d'écrans



- Pages HTML statiques
- SGBD non directement connecté
- Duplication des données
- Inconsistance

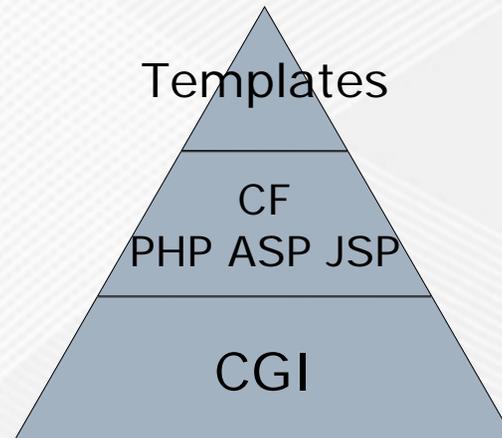
3e génération : Liaisons dynamiques



- Utilisé avec des formulaires
- Interactivité
- Pages dynamiques (Contenus dépendent du client)
- Contenu à jour par rapport aux données

Technologies de connexion

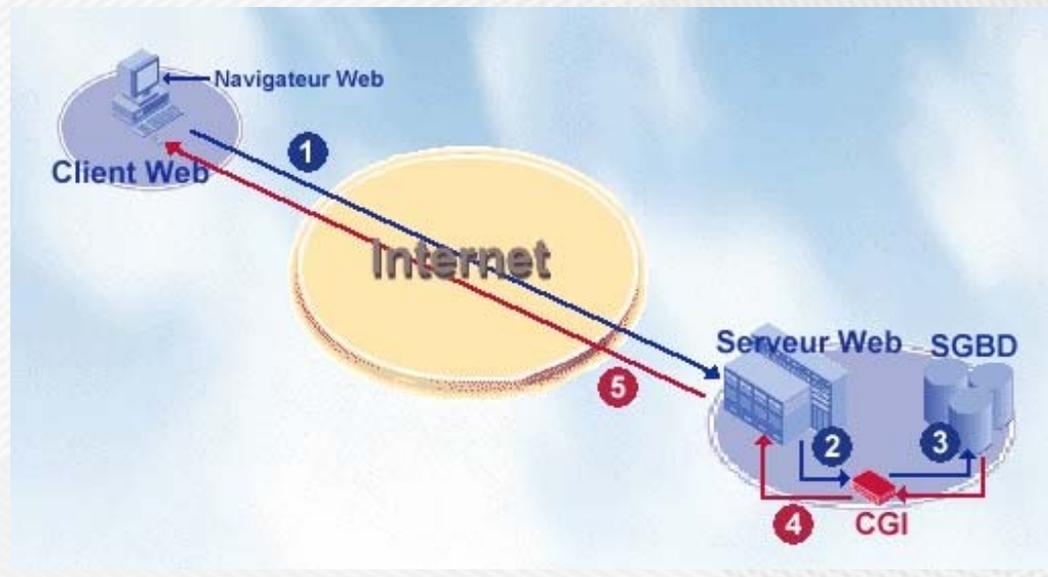
- CGI : le standard
 - API Serveurs: plus rapide mais propriétaire
- Langages de scripts serveur
 - JSP, ASP, PHP, Cold Fusion
 - Langages s'appuyant sur CGI et proposant des facilités (sessions, mixage codes tiers serveur/client)
- Templates
 - Tentatives de séparation du contenu et de la présentation
 - Existe de multiples déclinaisons dans tous les langages ci-dessus
 - Par exemple: struts, JSF (Java Server Faces), PHPtemplates, smarty...
- JAVA : ouvert et puissant
 - Servlets, Applets
 - Extension du champs d'application de ce langage au domaine du Web



Aujourd'hui
on ne voit
que CGI !

Passerelle CGI: principe

- Protocole d'échange entre serveur Web et application externe
 - présent sur tous les serveurs Web
- 5 étapes :



Passerelle CGI: développement

- Programmes L3G + CGI + API d'accès SGBD :
 - Perl (+ DBI + DBD ...)
 - Python
 - C, C++ (+ ODBC + Pro*C ...)
 - sh, csh, tcl, ksh..
 - Java (+ JDBC)
 - Tous les langages capables de produire des exécutables (C++, Delphi, Visual Basic...)
- Perl est le plus utilisé

Passerelle CGI : + et -

- Les +:
 - +Standardisé: reconnu sur tous les serveur Web et navigateurs
 - +Flexible: n'importe quel langage (C++, Perl, Java, ...) utilisable

- Les -:
 - Sans états : Absence de support des sessions
 - ne conserve rien d'une requête HTTP sur l'autre
 - Gaspillage de ressources
 - Chaque appel provoque la création d'un processus
 - La sécurité
 - Permettre l'exécution d'un programme CGI équivaut à laisser le monde entier exécuter un programme sur la machine serveur.
 - C'est le(s) développeur(s) de CGI qui est/sont responsables de la sécurité
 - Réutilisabilité
 - L'interface CGI provoque une prolifération de scripts ou binaires spécifiques à certaines pages Web
 - Pour mettre à jour la mise en forme HTML, il faut être développeur !

Extensions propriétaires de CGI

- Les « API » serveur :
 - Microsoft (ISAPI)
 - Netscape (NSAPI) – historique (n'existe plus)
 - « Modules » apache - Apache API
- Le pourquoi :
 - régler le problème du gaspillage de ressources des CGI
- Avantages:
 - requièrent moins de mémoire
 - L'initialisation ne se fait qu'une fois
- Inconvénients:
 - leur code est non portable (spécifique au serveur)

5/7 – Mise en œuvre de CGI

CGI : mise en oeuvre

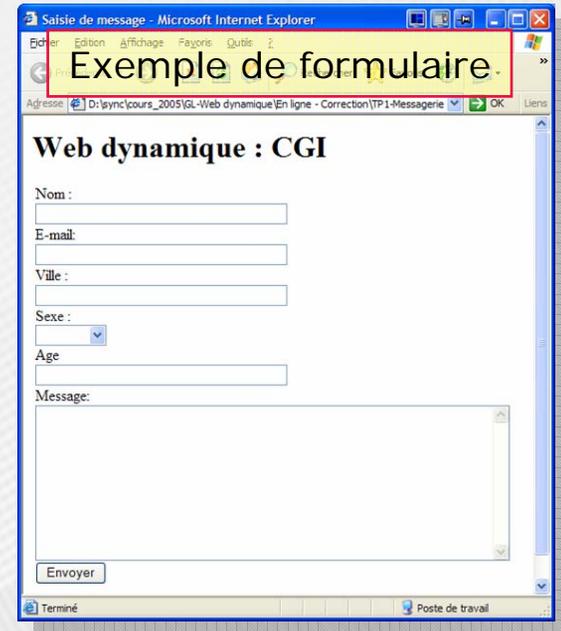
- Lié Unix et langage C
- Voies de communication:
 - entrée standard
 - sortie standard
 - Variables d'environnement
- Les serveurs Web offrent différentes options dans leur configuration:
 - CGI non autorisés
 - CGI forcés dans un répertoire sous le contrôle du webmaster (/cgi-bin)
 - CGI forcés d'avoir un suffixe spécial ('.cgi', '.pl')

CGI : mise en oeuvre

- Appel d'un programme CGI:
 - Par une requête HTTP !
 - Soit la requête GET
 - Soit la requête POST

- Déclencher un GET :
 - En saisissant simplement une URL
 - Via un formulaire HTML (method=GET)

- Déclencher un POST:
 - formulaire HTML uniquement(method=POST)



CGI : mise en œuvre

- Lorsque le bouton SUBMIT est cliqué, l'URL précisé dans l'attribut ACTION de l'élément FORM est appelé
- Méthode **GET** :
 - L'URL est appelé et les données ajoutées à la fin de l'URL dans une seule chaîne sans espace
<http://srv.fr/test.cgi?nom=val&nom=val>
 - Méthode non recommandée, certains serveurs ou OS limitent le nombre de caractères de l'URL
- Méthode **POST** :
 - L'URL est appelé simplement :
<http://srv.fr/user/test.cgi>
 - La partie paramètres est envoyée séparément dans l'entête HTTP
 - Même format que pour GET ([nom=val&nom=val](#))
 - Lisible par le CGI sur stdin

CGI : réception des paramètres

- ❑ Le programme CGI est renseigné via des variables d'environnement (par le serveur web)
- ❑ Il peut ainsi déterminer:
 - Par où lui vient la chaîne de paramètres : REQUEST_METHOD
 - De quelle taille est cette chaîne: CONTENT_LENGTH
 - Selon le cas la chaîne elle-même: QUERY_STRING
- ❑ Rappel: la chaîne de paramètres est le contenu de la saisie provenant du client (navigateur)

CGI : algorithme

L'algorithme du programme CGI sera le plus souvent:

1. Tester la chaîne «REQUEST_METHOD » pour récupérer les paramètres. Si :
 - "POST" : sur l'entrée standard (stdin)
 - "GET" : via la variable QUERY_STRING
2. Décoder les paramètres
 - CONTENT_LENGTH : taille des données transmises
 - Format de la chaîne de paramètres reçue :
 - « var1=valeur1&var2=valeur2... »
 - '&' joue le rôle de séparateur
 - Les espaces sont transformés en '+'
 - les caractères accentués sont transformés en '%xx'
3. Écrire sur la sortie standard le code décrivant la page HTML à renvoyer au client Web

CGI : envoie de la réponse

- ❑ Un programme CGI doit toujours fournir une réponse (un contenu)
- ❑ La réponse se fait via stdout (sortie standard)
- ❑ Sa réponse a le format suivant:
 - Entête destiné au serveur (renseignant sur le type de contenu): en-tête MIME (texte ascii, texte html, gif, mpeg, son, etc...).
 - Une ligne vide (séparateur) OBLIGATOIRE
 - Le contenu lui-même (HTML ou autre)
- ❑ La Réponse au client est les plus souvent:
 - un document (son contenu)
 - une référence à un document (son URL)

CGI : réponse avec un contenu

- ❑ Le CGI renvoi un document
- ❑ Le plus souvent il s'agit d'une page HTML
 - text/html pour du html
 - text/plain pour du texte simple
- ❑ le format de l'en-tête (content-type) est normalisé donc la syntaxe doit être rigoureusement respectée

- ❑ Exemple :

```
Content-type: text/html
```

```
Ici, saut une ligne vide
```

```
<HTML>  
  <HEAD> <TITLE>Document réponse d'un CGI</TITLE>  
  </HEAD>  
  <BODY> <H1>Hello from CGI !<H1> </BODY>  
</HTML>
```

CGI : réponse avec une URL

- ❑ Le CGI renvoi non pas un document mais la référence à un document (son URL)
- ❑ Il est intéressant de noter que le serveur http peut à ce moment là devenir à son tour client HTTP
- ❑ Exemple de réponse:

```
Content-type: text/html
Location: http://www.monserveur.fr/mapage.html
  Ici, saut une ligne vide
<HTML>
<HEAD>
<TITLE>Resultat au cas où...</TITLE>
</HEAD>
<BODY>La redirection ne fonctionne pas
</BODY>
</HTML>
```

- ❑ Le contenu HTML qui suit est facultatif
- ❑ sert au cas où le browser ne gère pas la redirection

CGI: Exemple de requête POST

Requête:

```
POST /cgi-bin/message.pl HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows; ...) Gecko/20050511 Firefox/1.0.4
Accept: text/xml, ...,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Referer: http://localhost:8080/TP1-Messagerie%20en%20CGI/PERL/form.html
Content-Type: application/x-www-form-urlencoded
Content-Length: 117
```

```
nom=nadir&email=boussou%40gmail.com&ville=Marseille&sexe=M&age=32
[...]&message=un+petit+message.%0D%0Adeuxieme+ligne%0D%0A
```

Réponse:

```
HTTP/1.1 200 OK
Date: Sun, 14 Aug 2005 15:03:36 GMT
Server: Apache/1.3.23 (Win32)
Transfer-Encoding: chunked
Content-Type: text/html
```

```
51
<HTML>
<BODY BGCOLOR=WHITE>
<CENTER>
<H1>Message reçu</H1>
</BODY>
</HTML>
```

CGI: Exemple de requête GET

Requête:

```
GET /cgi-bin/message.pl?nom=nadir&email=boussou%40gmail.com&ville=Marseille&sexe=M
[...]&age=32&message=ligne+1%0D%0Aaligne+2 HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows; ...) Gecko/20050511 Firefox/1.0.4
Accept: text/xml, ...,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Referer: http://localhost:8080/TP1-Messagerie%20en%20CGI/PERL/form.html
```

Réponse:

```
HTTP/1.1 200 OK
Date: Sun, 14 Aug 2005 15:08:47 GMT
Server: Apache/1.3.23 (Win32)
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html
```

```
51
<HTML>
<BODY BGCOLOR=WHITE>
<CENTER>
<H1>Message recu</H1>
</BODY>
</HTML>
```

CGI : variables transmises par le serveur

Autres variables d'environnement :

- ❑ Passées par le serveur
- ❑ pas forcément initialisées (REMOTE_USER par exemple)
- ❑ Celles qui commencent par HTTP sont issues de la requête

Extrait :

AUTH_TYPE
HTTP_USER_AGENT
HTTP_HOST
HTTP_ACCEPT
HTTP_ACCEPT_LANGUAGE
SERVER_SOFTWARE
SERVER_NAME
SERVER_ADDR
SERVER_PORT
SERVER_PROTOCOL
SERVER_SIGNATURE
DOCUMENT_ROOT
SERVER_ADMIN
SCRIPT_FILENAME
CONTENT_TYPE
CONTENT_LENGTH
GATEWAY_INTERFACE
REQUEST_METHOD
REQUEST_URI
QUERY_STRING

Exemples:

```
AUTH_TYPE : '(null)'  
HTTP_USER_AGENT : 'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)'  
HTTP_HOST : 'localhost'  
HTTP_ACCEPT : 'image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,  
application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword,  
application/x-shockwave-flash, */*'  
HTTP_ACCEPT_LANGUAGE : 'fr'  
SERVER_SOFTWARE : 'Apache/1.3.20 (Win32) PHP/4.0.6'  
SERVER_NAME : 'localhost'  
SERVER_ADDR='127.0.0.1'  
SERVER_PORT : '80'  
SERVER_PROTOCOL : 'HTTP/1.1'  
SERVER_SIGNATURE='<ADDRESS>Apache/1.3.23 Server ...</ADDRESS>\n'  
DOCUMENT_ROOT : '/servcd/www'  
SERVER_ADMIN : 'admin@domain.com'  
SCRIPT_FILENAME : '/servcd/www/cgi-bin/cgi-env.exe'  
GATEWAY_INTERFACE : 'CGI/1.1'  
REQUEST_METHOD : 'GET'  
REQUEST_URI="/cgi-bin/cgi-env.exe"
```

6/7 Présentation avec HTML

HTML

- HTML est un langage standardisé pour décrire la *structure* et le *contenu* des documents qui comportent de l'hypertexte
- HTML est un langage de description
- HTML définit un ensemble de **balises** (« Tags ») pour
 - formater le texte
 - incorporer d'autres éléments
 - Faire référence à des documents

HTML : + et -

- + Universel :
 - s'adapte à toutes les plate-formes (Windows, Macintosh, Unix, OS/2...)

- + Léger :
 - Réduire le trafic au minimum

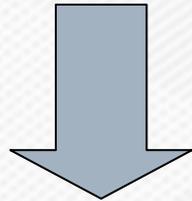
- - précision :
 - Interprétation par les navigateurs
 - « versions » de navigateurs

HTML : Tags

- **Debut** « < » et « > »
- **Fin** : « </ » et « > ».

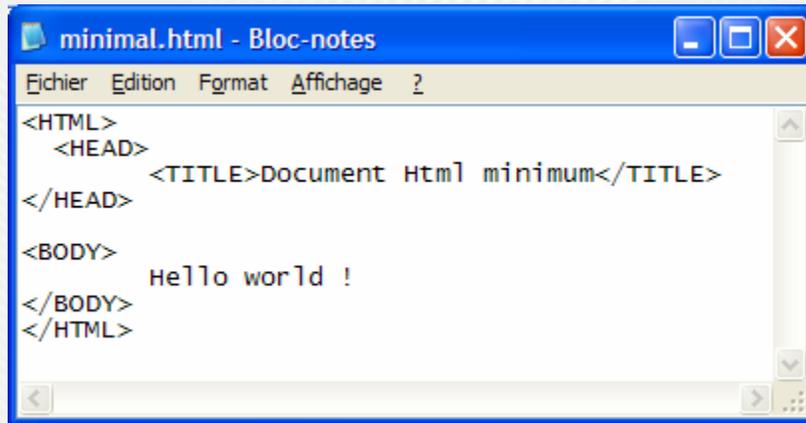
exemple :

Voici du texte important<U> et souligné</U>

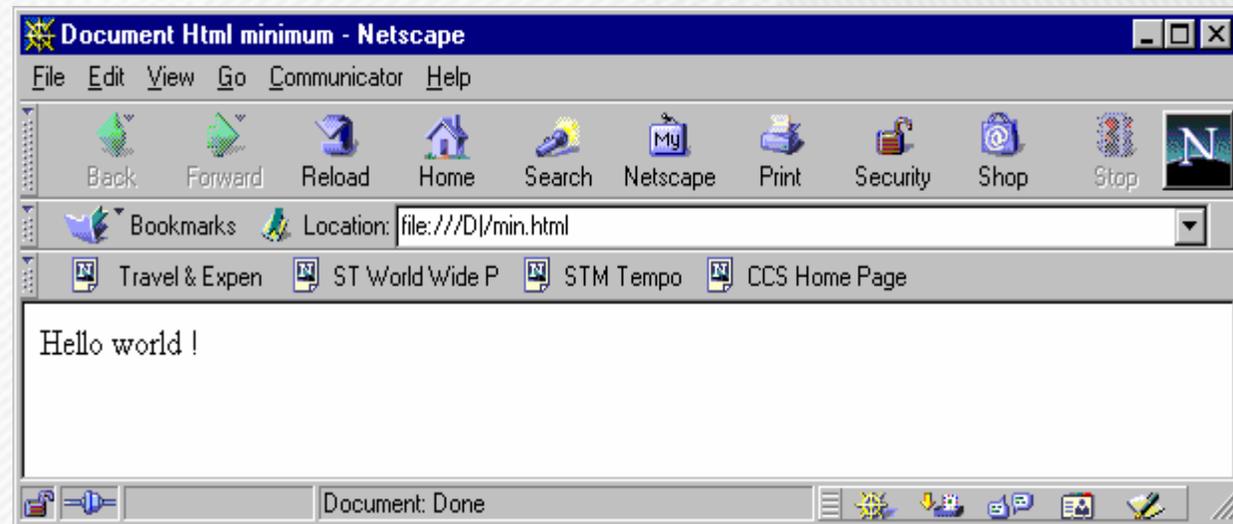


Voici du texte **important et souligné**

HTML: Structure d'un document



```
minimal.html - Bloc-notes
Fichier Edition Format Affichage ?
<HTML>
  <HEAD>
    <TITLE>Document Html minimum</TITLE>
  </HEAD>
  <BODY>
    Hello world !
  </BODY>
</HTML>
```



HTML : mise en forme de texte

Gras	<code>...</code> <code>...</code>
Italique	<code><I>...</I></code> <code>...</code>
Taille (1-7 et 3 par défaut)	<code>...</code>
Couleur (details prochain slide)	<code> </code>
Retour A la ligne	<code>
</code>
Nouveau paragraphe	<code><P></code>
Alignements	<code><P ALIGN=LEFT> ... </P></code> <code><P ALIGN=CENTER> ... </P></code> <code><P ALIGN=RIGHT> ... </P></code>
Titres niveaux 1 à 6	<code><H1>...grand titre...</H1></code> jusque <code><H6>...petit titre...</H6></code>

NOTE 1: Certaines balises (comme FONT ou P) peuvent avoir des attributs

NOTE 2: On peut cumuler les attributs (ex: FONT)

HTML : couleurs

aqua	#00FFFF	navy	#000080
black	#000000	olive	#808000
blue	#0000FF	purple	#800080
fuchsia	#FF00FF	red	#FF0000
gray	#808080	silver	#C0C0C0
green	#008000	teal	#008080
lime	#00FF00	white	#FFFFFF
maroon	#800000	yellow	#FFFF00

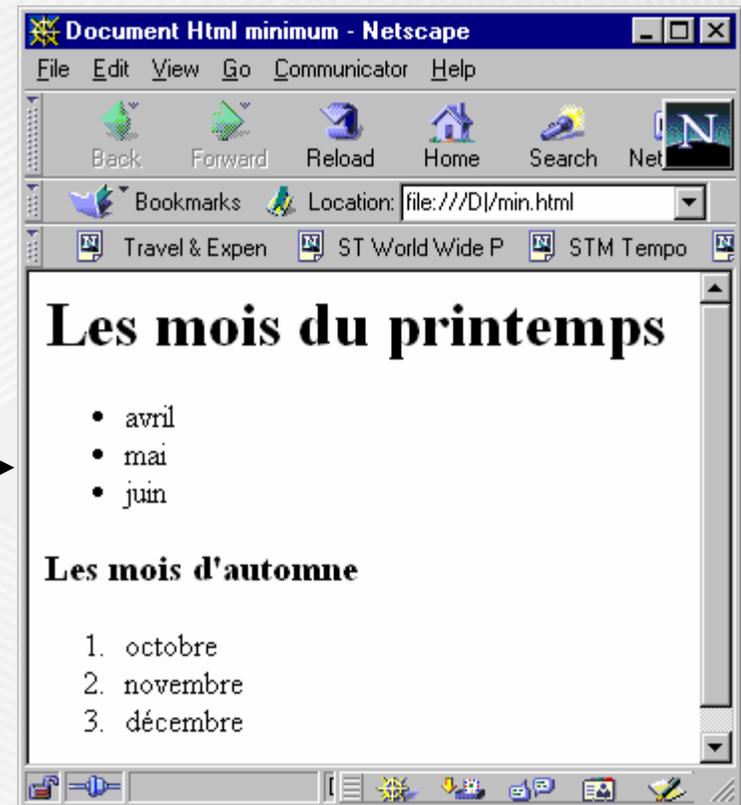
- Utilisable dans des balises pour les textes, tableaux etc
- On peut soit indiquer les composante RVB soit indiquer des noms en anglais
 - red, blue, green, cyan, magenta...
 - Composantes RVB sous forme #RRVVBB avec chacune allant de 00 à FF (256 tons)
- Ci-contre les 16 noms de couleurs de Microsoft, les plus répandues
- IE, Firefox reconnaissent en plus les 140 noms de couleurs de Netscape
 - cyan, darkgray, turquoise, violet ...
- Exemples:
 - <Font color="silver"...
 - <table bgcolor=#FF0000...
 - <table bgcolor="red"...

HTML : listes

Liste non-ordonnée	
Liste ordonnée	
Élément de liste	
Paragraphe	<P></P>



```
<H1>Les mois du printemps</H1>
<UL>
<LI>avril
<LI>mai
<LI>juin</UL>
<P>
<H3>Les mois d'automne</H3>
<OL><LI>octobre
<LI>novembre
<LI>décembre</OL>
```



HTML: les liens

- Fondement de l'hypertexte !
- Permet de cliquer pour sauter à un autre document
- Syntaxe
 - `...`
- Possibilités
 - vers un autre endroit du document (une marque)
 - vers un autre fichier situé sur le même serveur (url relative)
 - vers un autre fichier situé sur un autre serveur sur le Web (url complète de la forme `http://...`)
 - Il est même possible de sauter à une marque sur un autre fichier sur un autre serveur sur le Web

HTML: les liens

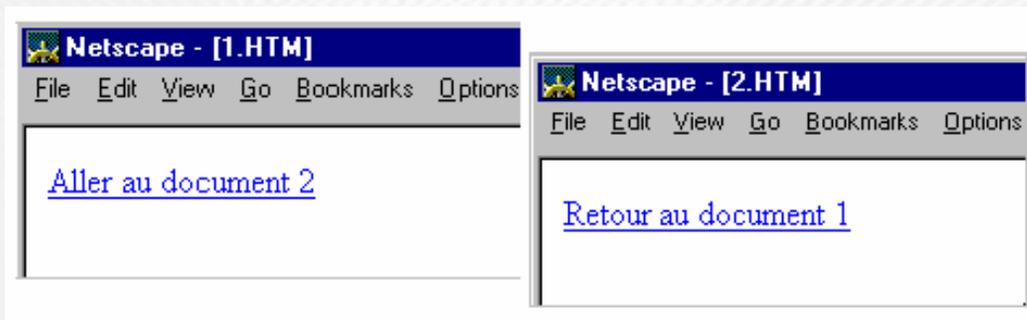
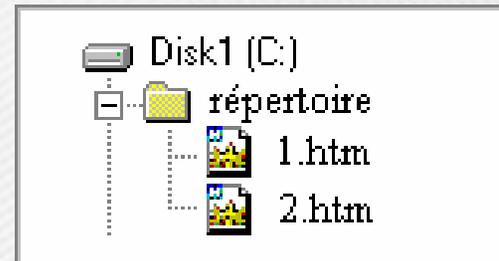
□ Soit la configuration suivante:

fichier 1.htm

```
<BODY>
<A HREF="2.HTM">Aller vers
le document 2</A>
</BODY>
```

fichier 2.htm

```
<BODY>
<A HREF="# 1.HTM">Aller
vers le document 1</A>
</BODY>
```



Les Marques:

Et
...

HTML : les tables

```
<TABLE>
<TR>
  <TD>1</TD><TD>2</TD>
</TR>
<TR>
  <TD>3</TD><TD>4</TD>
</TR>
</TABLE>
```

```
<TABLE border=1>
<TR>
  <TD>1</TD><TD>2</TD>
</TR>
<TR>
  <TD>3</TD><TD>4</TD>
</TR>
</TABLE>
```

```
<TABLE border=1 CELSPACING=0
  CELLPADDING=5>
<TR>
  <TD>1</TD><TD>2</TD>
</TR>
<TR>
  <TD>3</TD><TD>4</TD>
</TR>
</TABLE>
```

Memo:
Table Row,
Table data

1	2
3	4

1	2
3	4

1	2
3	4

```
<TABLE border=1 CELSPACING=0 CELLPADDING=5>
<TR >
  <TD rowspan=4>1</TD>
</TR>
<TR>
  <TD>2</TD>
</TR>
<TR>
  <TD>3</TD>
</TR>
<TR>
  <TD>4</TD>
</TR>
</TABLE>
```

1	2
	3
	4

```
<TABLE border=1 CELSPACING=0 CELLPADDING=5>
<TR>
  <TD colspan=3>1</TD>
</TR>
<TR>
  <TD>2</TD><TD>3</TD><TD>4</T
D>
</TR>
</TABLE>
```

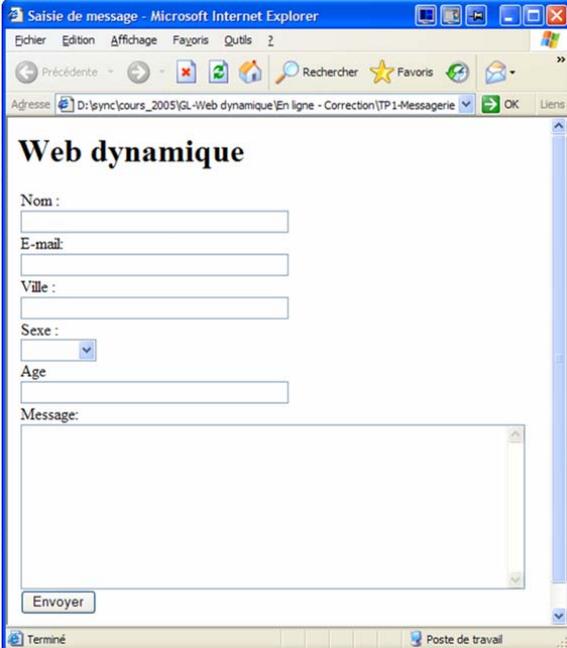
1		
2	3	4

HTML : non couvert

- Les balises cosmétiques (HR,IMG...)
- Tous les attributs de balises
- Les frames
- Les images et les images cliquables
- Les imagemaps
- Les détails de la balise <HEAD>
- Les CSS (et les balises associés)

7/7 Les formulaires HTML

Interactivité du Web



The screenshot shows a Microsoft Internet Explorer browser window with the title 'Saisie de message - Microsoft Internet Explorer'. The address bar contains the path 'D:\sync\cours_2005\GL-Web dynamique\En ligne - Correction\TP1-Messagerie'. The main content area displays a web form titled 'Web dynamique' with the following fields:

- Nom :
- E-mail :
- Ville :
- Sexe :
- Age :
- Message:

At the bottom of the form is an 'Envoyer' button. The browser's status bar at the bottom shows 'Terminé' and 'Poste de travail'.

HTML : les formulaires

- Interactivité: Permet la saisie d'information dans le navigateur Web
 - Utilisation de CGI (Envoie des données vers le serveur)

- Définition :

```
<FORM method="post" action="URL d'expédition" >
```

... Ici définitions HTML normales et spécifiques à la saisie

```
</FORM >
```

- Exemple :

```
<FORM method="post" action="http://www.serveur.fr/cgi-bin/mon_cgi.pl" >
```

- La méthode correspond au mode d'envoi des données saisies vers la destination conformément au protocole CGI

HTML : les formulaires

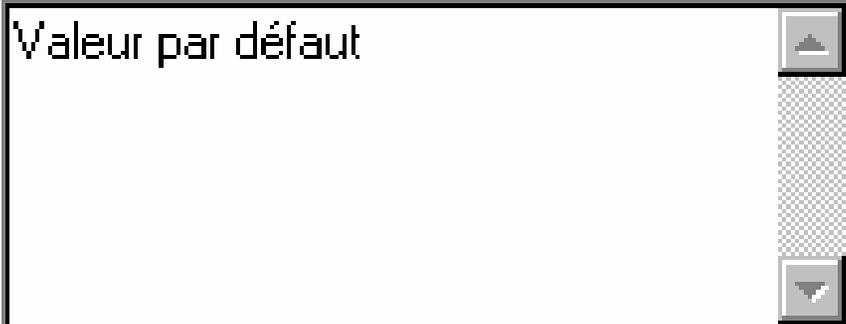
□ Texte

- `<INPUT type="text" name="nom" size="50">`

A single-line text input field with a thin black border and a light gray background, representing the visual output of the HTML code above.

□ Zone de texte

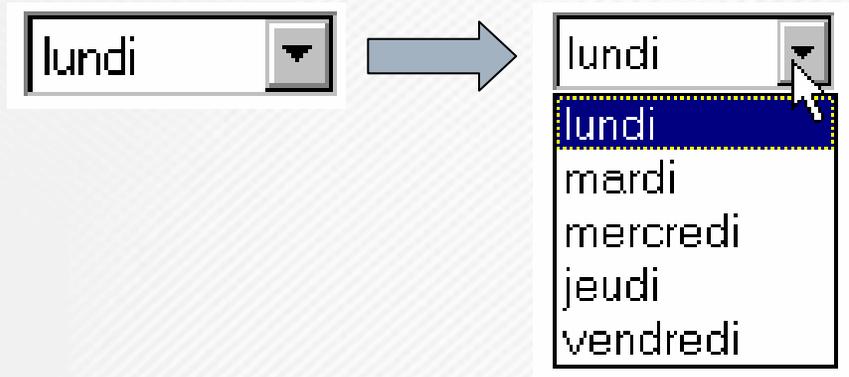
- `<TEXTAREA name="nom" rows=4 cols=40>Valeur par défaut</TEXTAREA>`

A multi-line text area with a thin black border and a light gray background. It contains the text "Valeur par défaut" in the top line. On the right side, there is a vertical scrollbar with a gray track and a white slider.

HTML : les formulaires

□ Liste déroulante

```
<SELECT name="nom" size="1">  
<OPTION>lundi  
<OPTION>mardi  
<OPTION>mercredi  
<OPTION>jeudi  
<OPTION>vendredi  
</SELECT>
```



```
<SELECT name="nom" size="5">  
<OPTION value=LU>lundi  
<OPTION value=MA>mardi  
<OPTION value=ME>mercredi  
<OPTION value=J>jeudi  
<OPTION value=V>vendredi  
</SELECT>
```



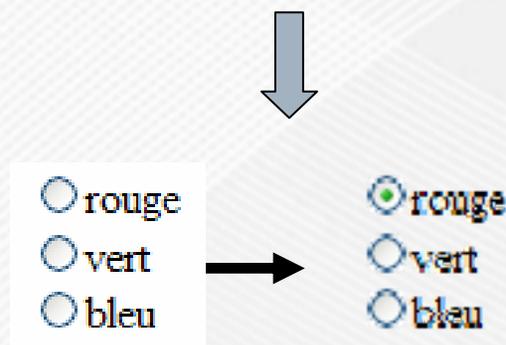
HTML : les formulaires

□ boutons « Radio »

- `<INPUT type="radio" name="nom du groupe" value="valeur">`
- On peut ajouter « checked » pour indiquer celui qui doit être sélectionné par défaut (`<INPUT type="radio" name="X" value="V" checked>`)

□ Exemple :

- `<INPUT type="radio" name="tarif" value="rouge">rouge
`
- `<INPUT type="radio" name="tarif" value="vert">vert
`
- `<INPUT type="radio" name="tarif" value="bleu">bleu
`



HTML : les formulaires

☐ Cases à cocher

- `<INPUT type="checkbox" name="nom" value="valeur si cochée" >`
- On peut aussi ajouter « checked » pour indiquer si elle doit être cochée par défaut

☐ Exemple :

- `<INPUT type="checkbox" name="choix1" value="1" >glace vanille`
- `<INPUT type="checkbox" name="choix2" value="2" >chantilly`
- `<INPUT type="checkbox" name="choix3" value="3" >chocolat chaud`
- `<INPUT type="checkbox" name="choix4" value="4" >biscuit`



<input type="checkbox"/> glace vanille	→	<input type="checkbox"/> glace vanille
<input type="checkbox"/> chantilly		<input checked="" type="checkbox"/> chantilly
<input type="checkbox"/> chocolat chaud		<input type="checkbox"/> chocolat chaud
<input type="checkbox"/> biscuit		<input checked="" type="checkbox"/> biscuit

HTML : les formulaires

- Boutons spéciaux : Submit et Reset
 - `<INPUT TYPE="submit" NAME="nom" VALUE=" Envoyer" >`
 - Submit déclenche l'envoi vers le programme CGI (action)
 - Reset est moins utilisé
 - Il en faut obligatoirement un dans un formulaire!



- Bouton cachés
 - `<INPUT TYPE="hidden" NAME="mask" VALUE="fantomas" >`
 - Intérêt: un script ou un CGI peut y placer une valeur qu'il retrouvera lorsque le formulaire lui sera renvoyé