

Telecommunication Services Engineering (TSE) Lab



Chapitre 3: La toile

Roch Glitho, PhD

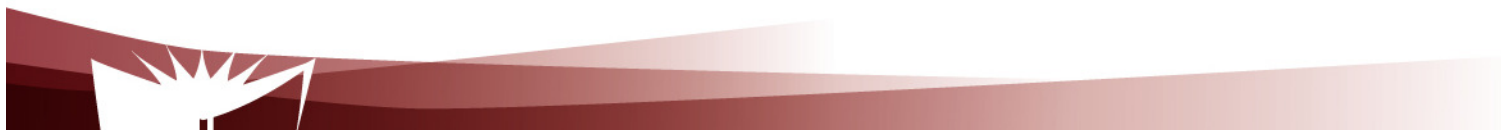
Associate Professor and Canada Research Chair

My URL - <http://users.encs.concordia.ca/~glitho/>

Telecommunication Services Engineering (TSE) Lab

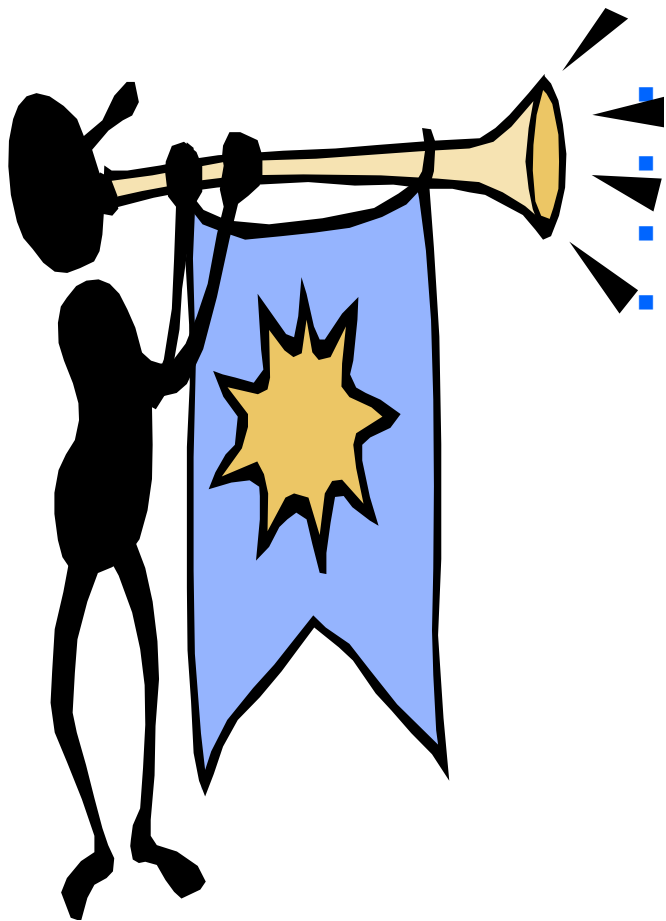
Couche Application

Application
Transport
Réseau
Liaison de données
Support physique

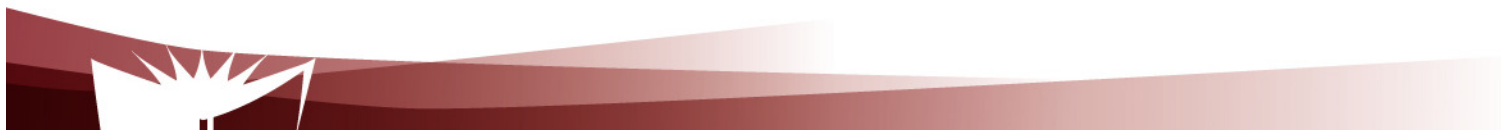


Telecommunication Services Engineering (TSE) Lab

Table des matières (La toile)

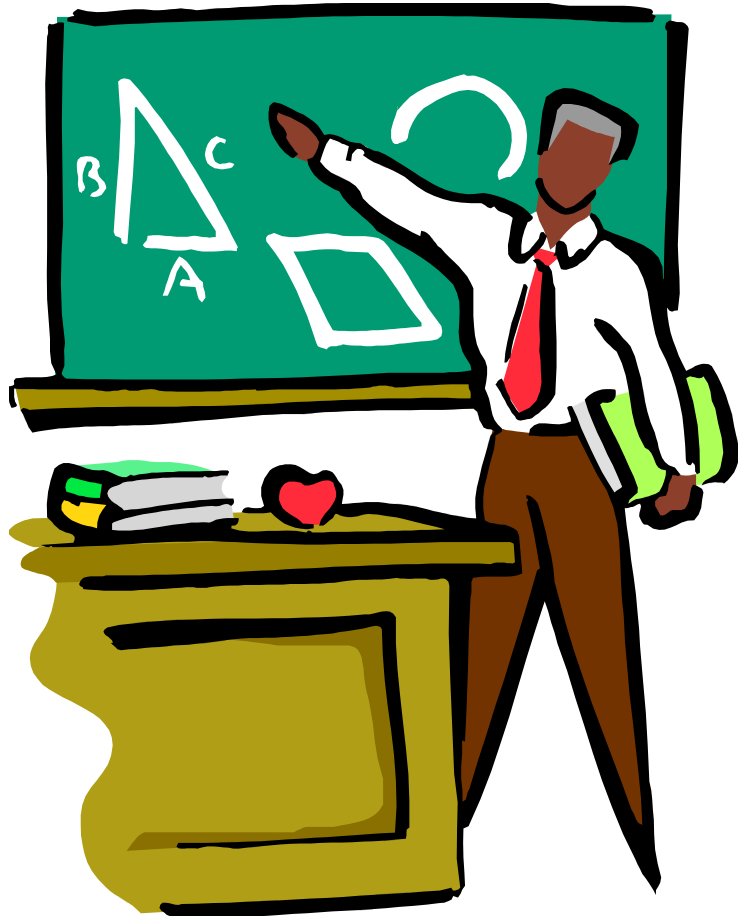


- Les jalons
- Architecture
- Documents
- Le protocole HTTP

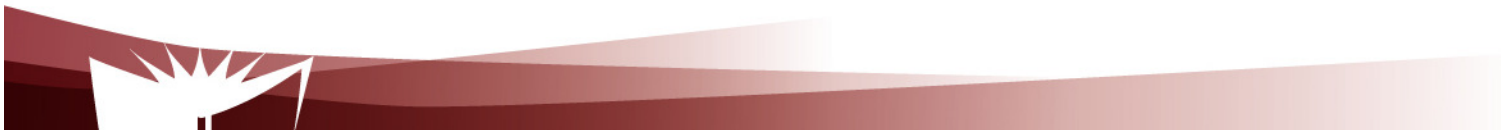


Telecommunication Services Engineering (TSE) Lab

La toile



Les jalons (Perspective historique)



Telecommunication Services Engineering (TSE) Lab

Les jalons

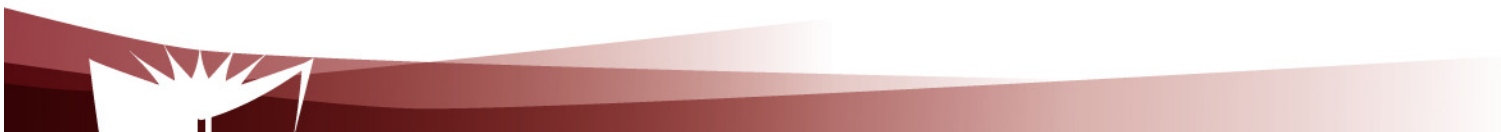
La toile (World Wide Web – WWW)

Cadre applicatif pour l'accès (via Internet) à des documents reliés entre eux

1989: Idée née au centre européen de recherche nucléaire (CERN) de Genève

Besoin pour des scientifiques géographiquement dispersés de collaborer (rapports, plans, dessins) de manière dynamique

1991: Première démonstration en aux Etats Unis (Conférence Hypertext)



Telecommunication Services Engineering (TSE) Lab

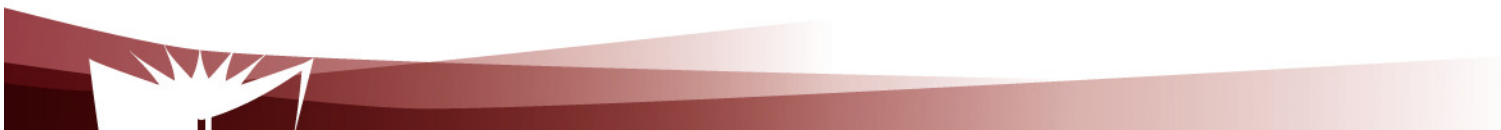
Les jalons

La toile (World Wide Web – WWW)

Cadre applicatif pour l'accès (via Internet) à des documents reliés entre eux

1993: Premier navigateur (i.e. Mosaic) disponible en 1993
(Université de Illinois)

1994: Naissance de world wide web consortium (W3C)
- organisme de normalisation du cadre application (ex: entités, protocoles)



Telecommunication Services Engineering (TSE) Lab

Les jalons

La toile (World Wide Web – WWW)

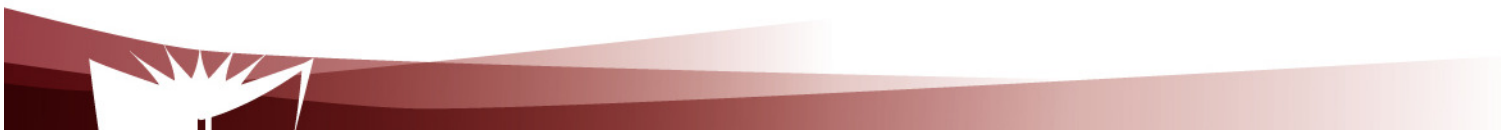
Cadre applicatif pour l'accès (via Internet) à des documents reliés entre eux

1995: Naissance des premiers navigateurs qui sont des produits

- Netscape – Utilise Mosaic comme base
- Microsoft Internet Explorer

1995 – 1998:

Rivalité entre Netscape et Microsoft explore qui entraîna la disparition de Netscape



Telecommunication Services Engineering (TSE) Lab

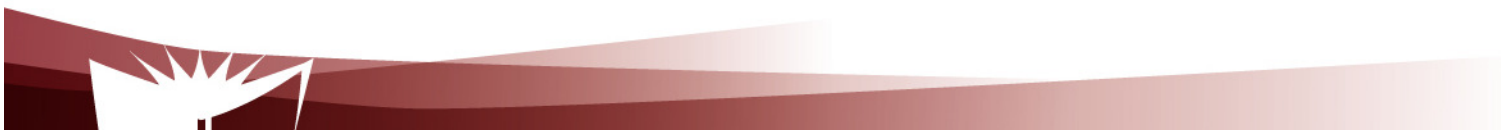
Les jalons

La toile (World Wide Web – WWW)

Cadre applicatif pour l'accès (via Internet) à des documents reliés entre eux

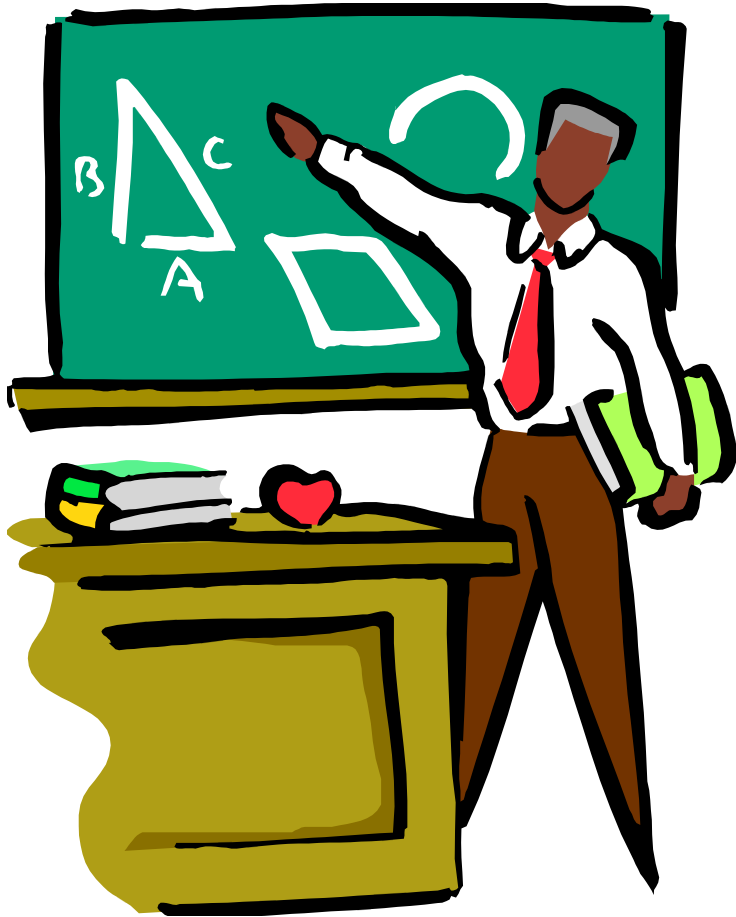
Depuis lors

- Microsoft detient la plus grosse part du marché
- Naissance par la suite d'autres navigateurs tels que Firefox

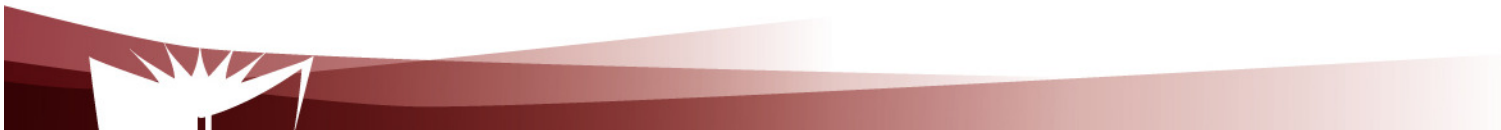


Telecommunication Services Engineering (TSE) Lab

Architecture



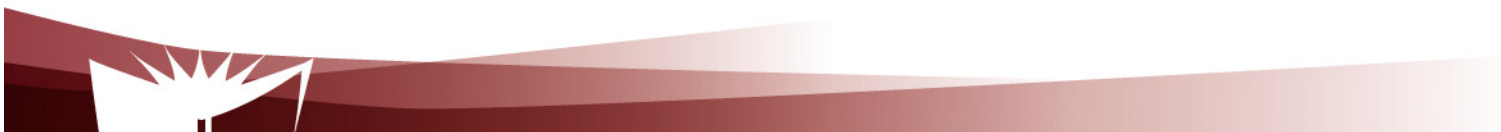
- 1 - Survol
- 2. Navigateur
- 3. Serveur Web



Survol

Client / serveur

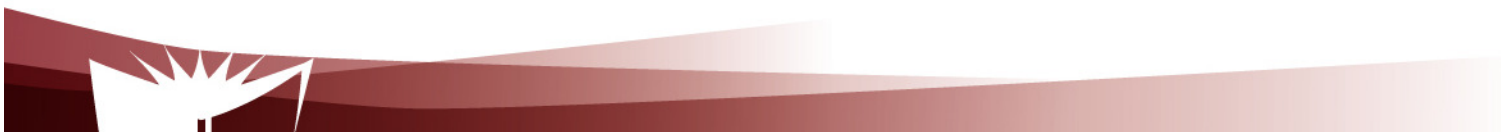
- Côté client: Navigateur
- Côté serveur: Pages a afficher
 - Universal Resource Locator (URL) utilisé comme identificateur
 - Ex: <http://www.abc.com/products.html>



Survol

Client / serveur

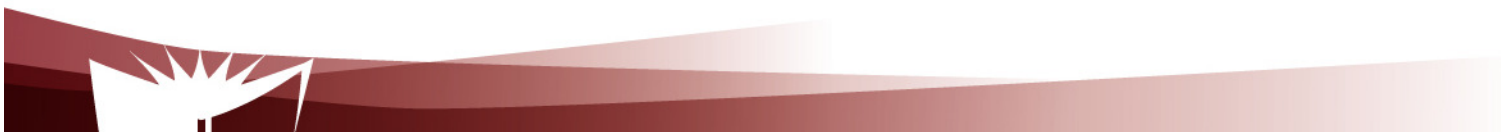
- Les étapes: côté client
 1. Détermination de l'URL
 2. Interrogation du serveur DNS pour avoir l'adresse IP
 3. Initiation d'une connexion TCP sur le port 80
 4. Envoi de la requête HTTP



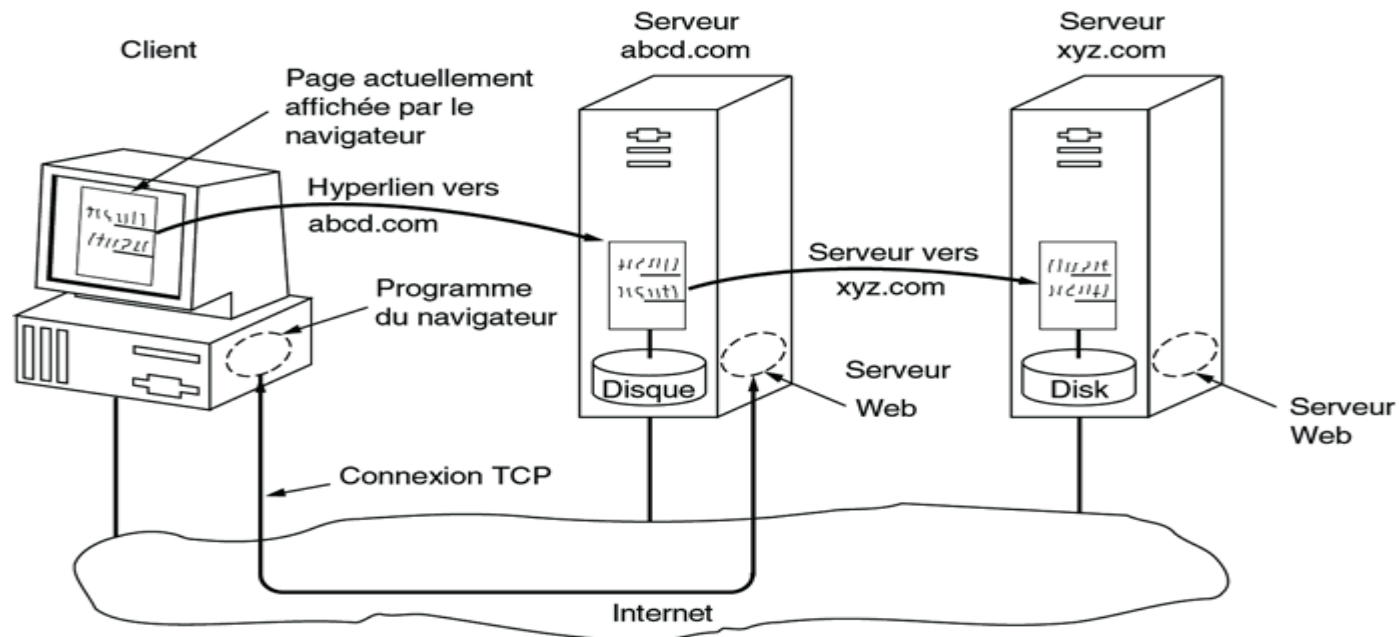
Survol

Client / serveur

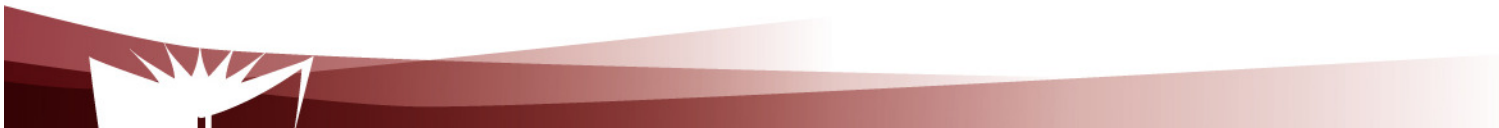
- Les étapes: côté serveur
 5. Réponse à la requête HTTP incluant document demandé
 6. Fin connexion
- Côté client
 7. Affichage du document



Survol



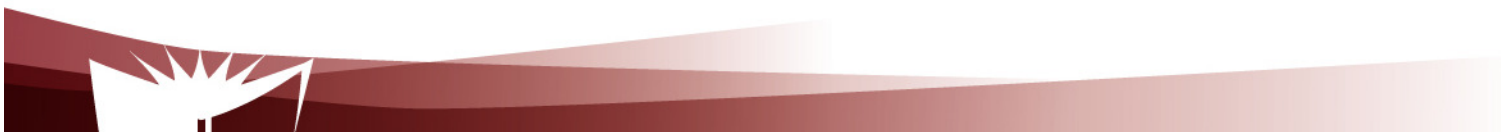
© Pearson Education France



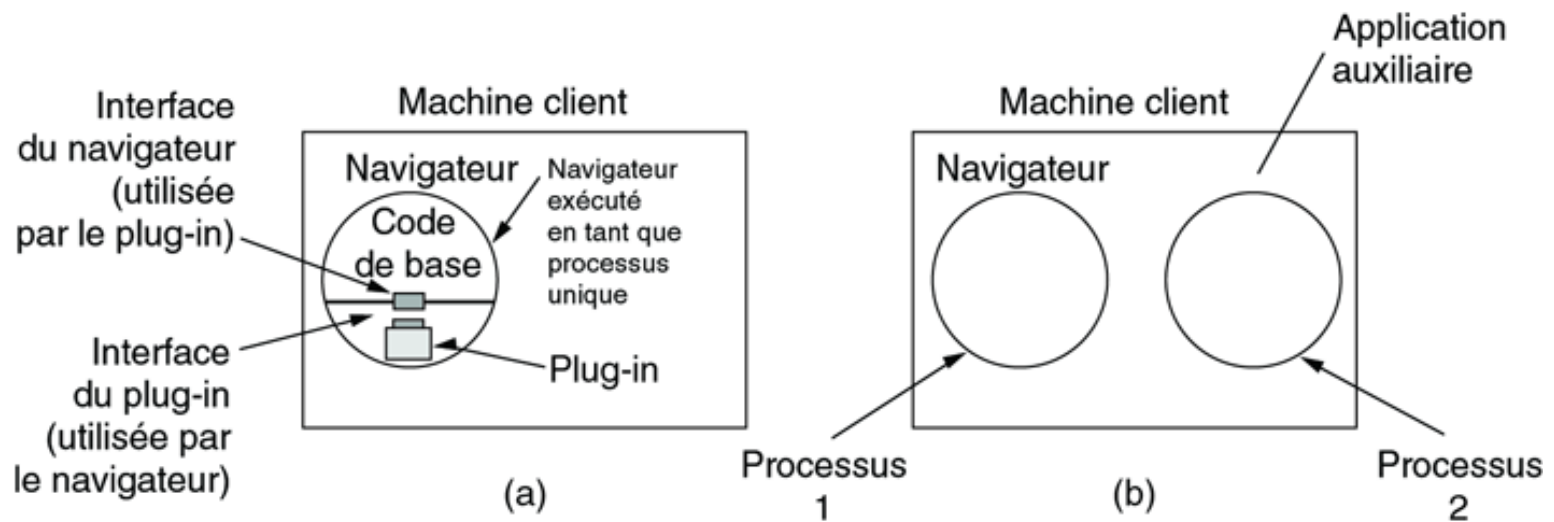
Le navigateur

Comment étendre la fonctionnalité de base

- Plug-in
 - Code que le navigateur peut ajouter a son code de base et l'exécuter dans le m^eme processus
- Application auxiliaire
 - Exécuté come processus distinct

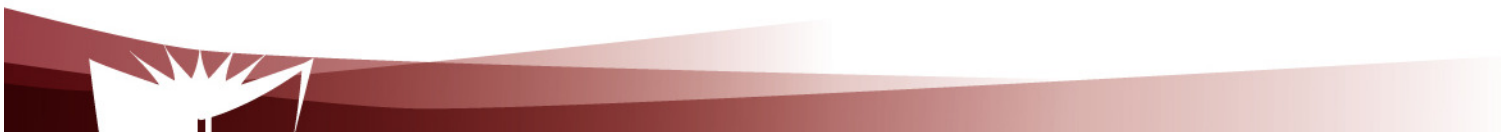


Le navigateur



© Pearson Education France

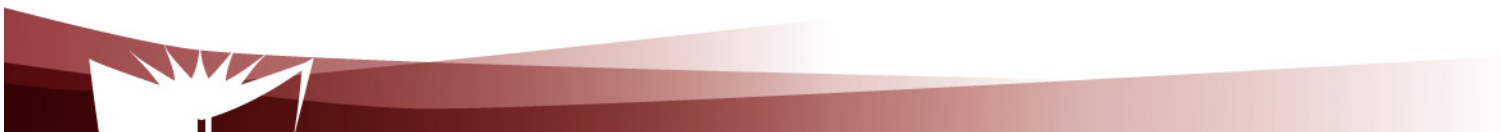
(a) A browser plug-in. (b) A helper application.



Le serveur Web

Tâches détaillées: Authentification, contrôle d'accès

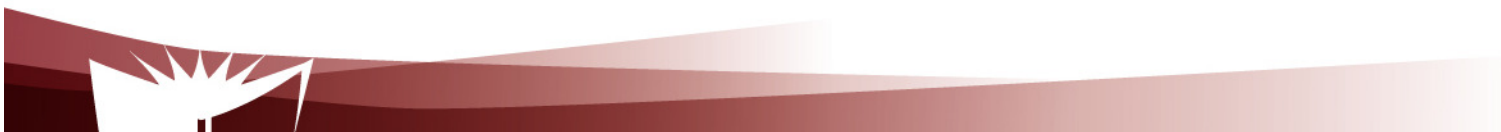
1. Résoudre nom page Web
2. Authentifier le client
3. Réaliser le contrôle d'accès du client
4. Réaliser le contrôle d'accès sur la page Web



Le serveur Web

Tâches détaillées: Recherche de la page demandée

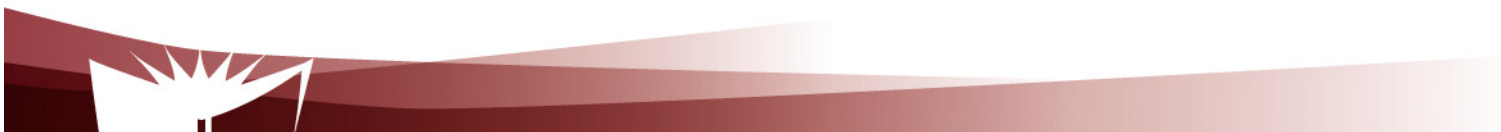
1. Vérifier l'anté mémoire (cache)
2. Rechercher la page demandée depuis un disque



Le serveur Web

Tâches détaillées: Préparation réponse et envoie au client

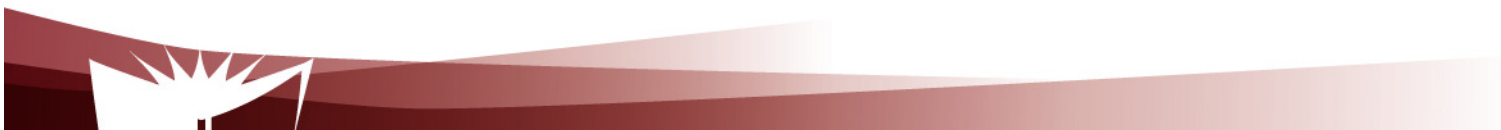
1. Déterminer le type de MIME pour l'inclure dans la réponse
2. Envoyer la réponse au client
3. Consigner une entrée dans le journal du serveur



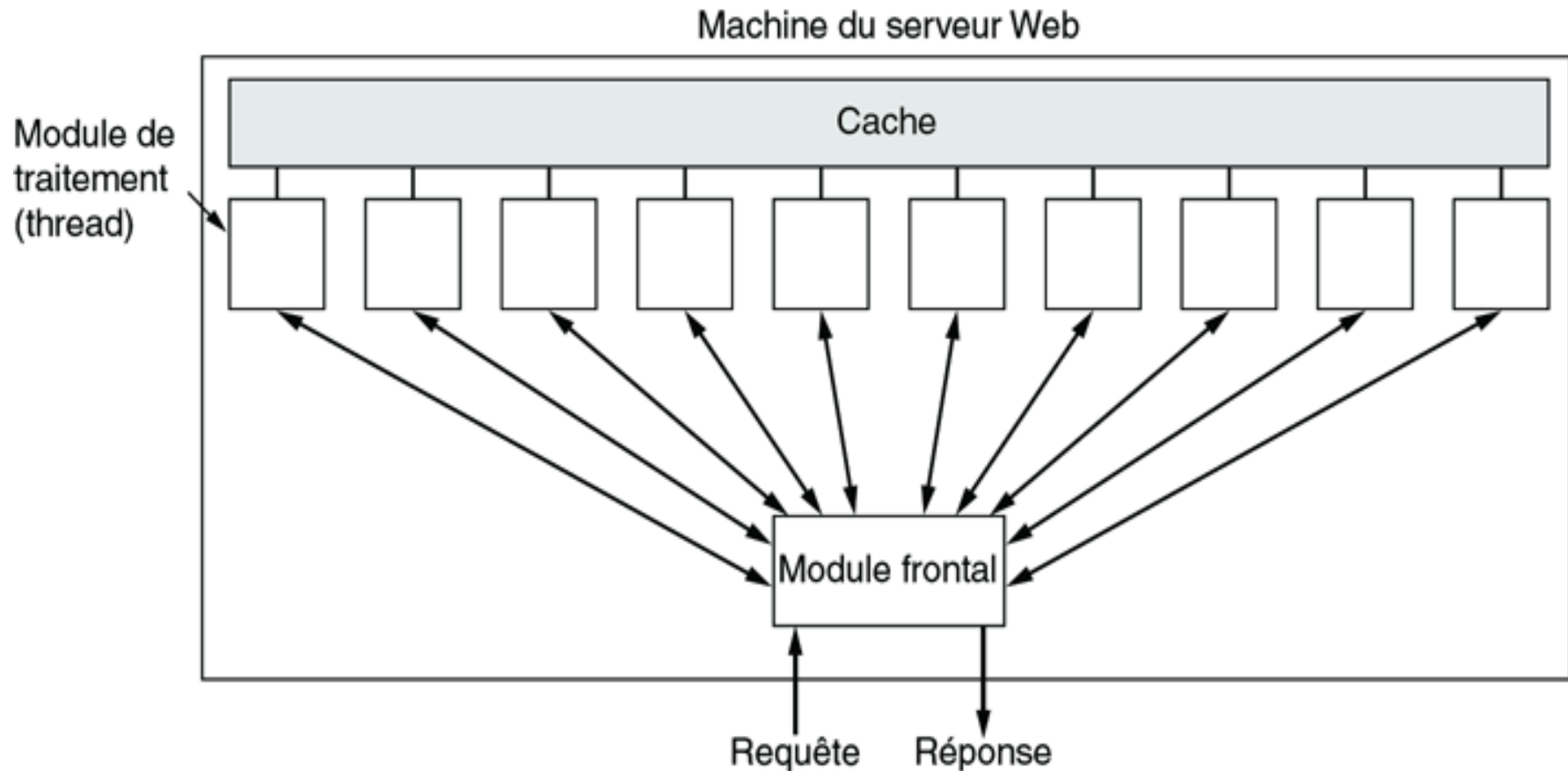
Le serveur Web

Pour améliorer les performances

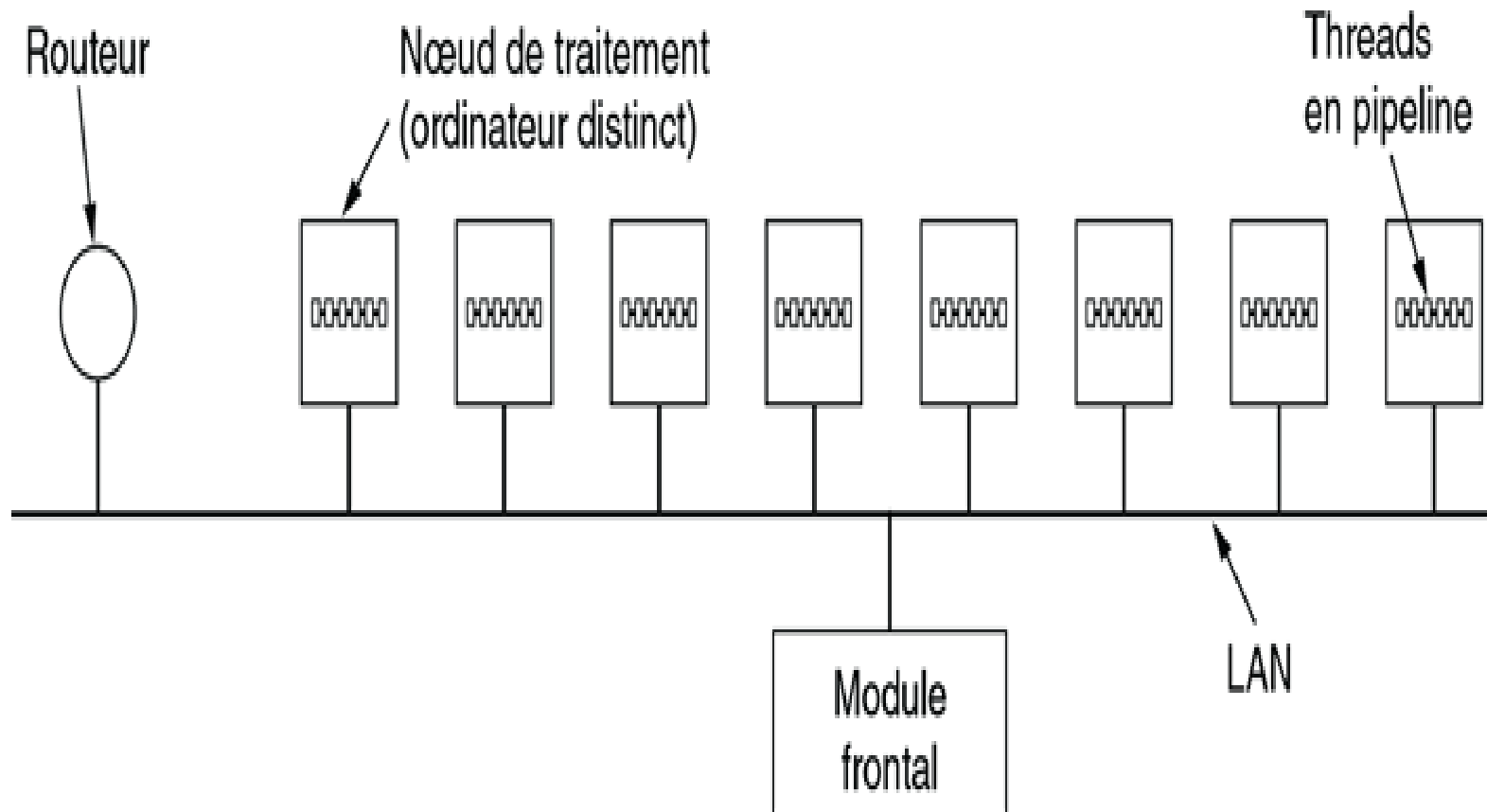
1. Anté mémoire
2. Multi threading
3. Ferme de serveurs



Le serveur Web



Le serveur Web

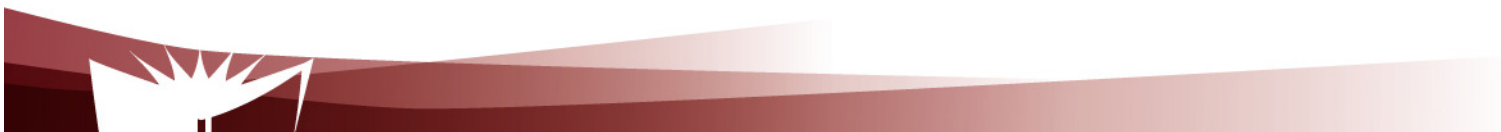


Le serveur Web

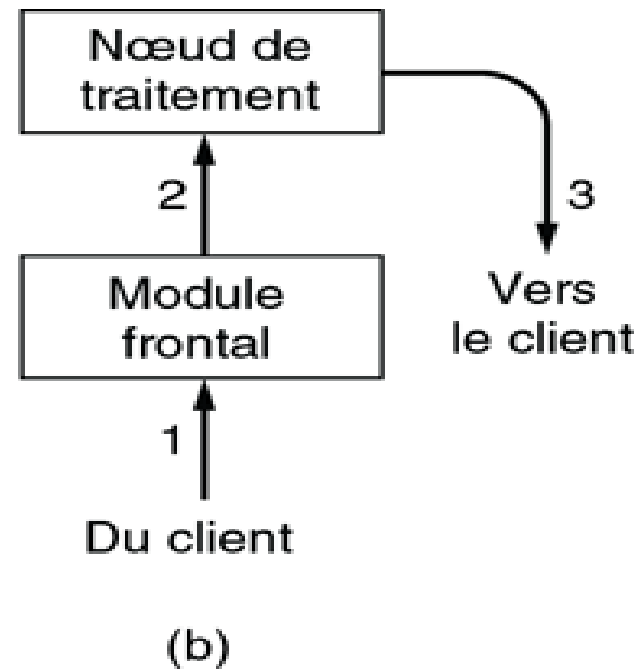
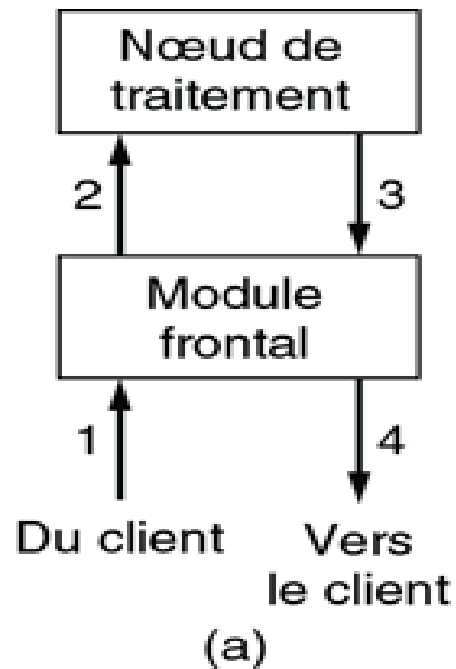
Connexion TCP

Dexu possibilités

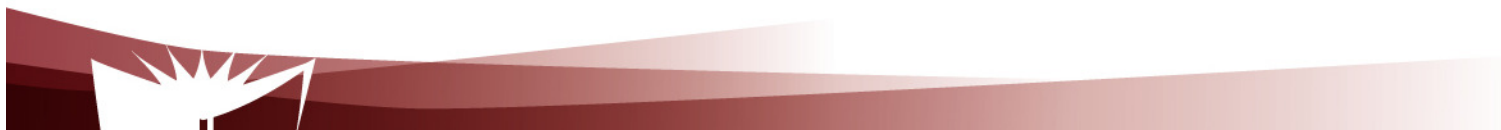
1. Traitement exclusif par module frontal
2. Triangulation avec implication du serveur lui-même



Le serveur Web

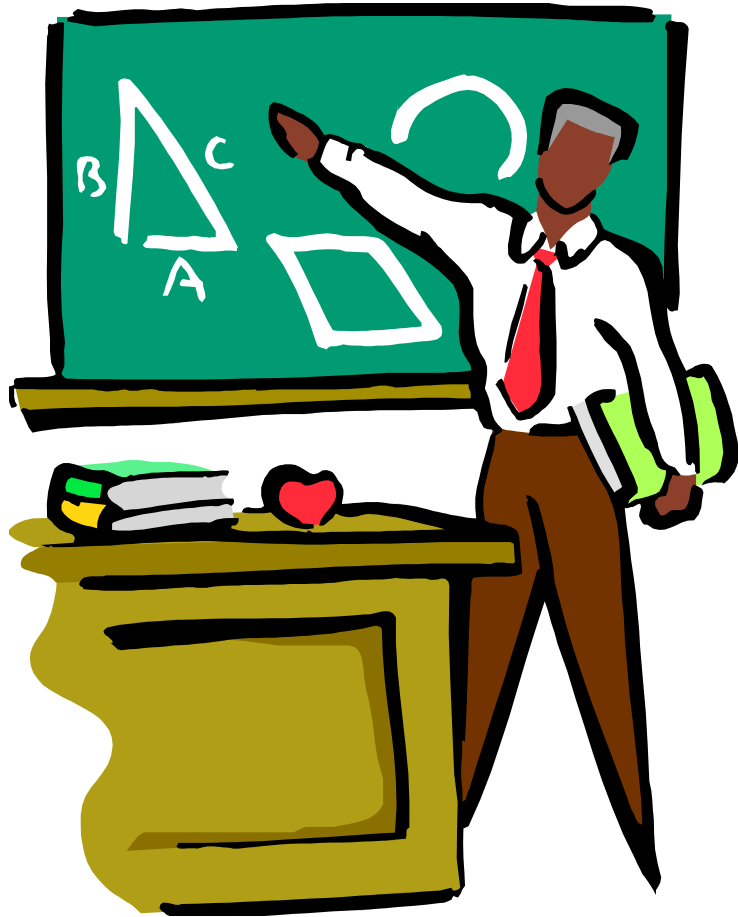


© Pearson Education France

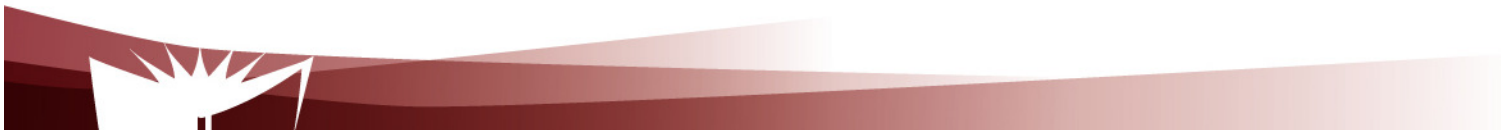


Telecommunication Services Engineering (TSE) Lab

Documents



- 1 - Identification
- 2. Document statique
- 3. Document dynamique

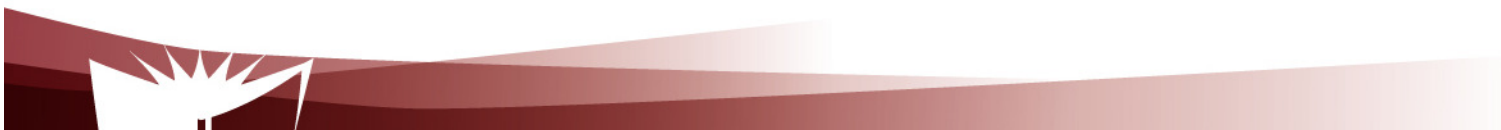


Identification

Identificateur = URL (Universal Resource Locators)

Questions essentielles

1. Quel est le nom de la page?
2. Ou est ce que la page se trouve?
3. Comment peut on accéder à la page?

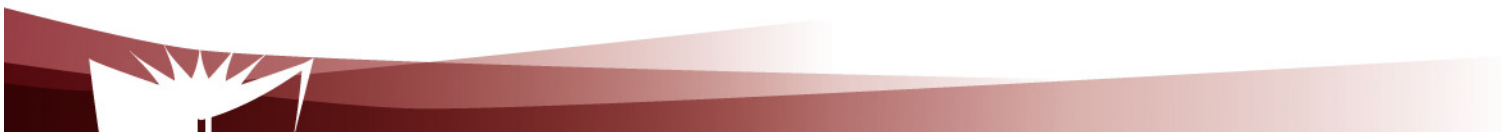


Identification

URL

Réponses

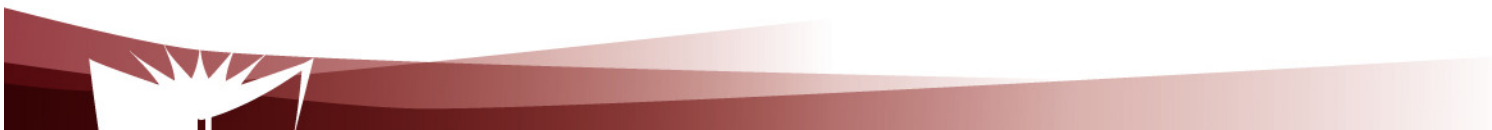
1. Quel est le nom de la page?
Nom local (nom de fichier)
2. Ou est ce que la page se trouve?
Nom DNS de la machine où se trouve le fichier
3. Comment peut on accéder à la page?
le protocole utilisé (ex: HTTP, FTP)



Telecommunication Services Engineering (TSE) Lab

Identification

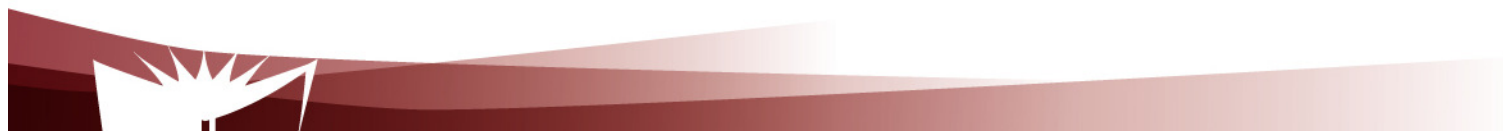
Name	Used for	Example
http	Hypertext (HTML)	http://www.cs.vu.nl/~ast/
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	Local file	file:///usr/suzanne/prog.c
news	Newsgroup	news:comp.os.minix
news	News article	news:AA0134223112@cs.utah.edu
gopher	Gopher	gopher://gopher.tc.umn.edu/11/Libraries
mailto	Sending e-mail	mailto:JohnUser@acm.org
telnet	Remote login	telnet://www.w3.org:80



Documents statiques

HTML (Hypertext Mark up Language)

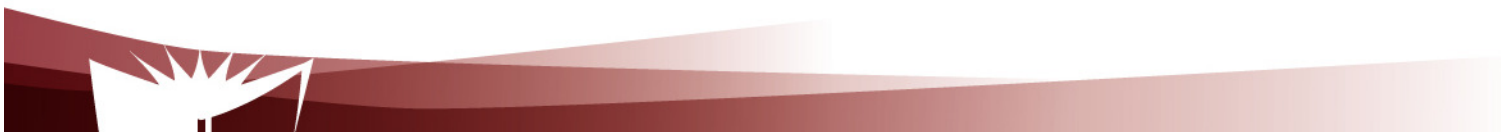
- Texte
- Graphique
- Pointeurs vers d autres pages



Documents statiques

HTML (Hypertext Mark up Language)

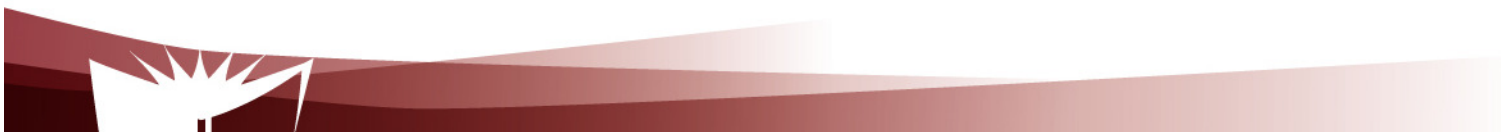
- Language de balisage (Balise = tag)
 - Inclut commande explicites concernant comment formater le texte
 - Exemple
 - `` : Debut de mis en gras
 - `` : Fin de mis en gras



Documents statiques

HTML (Hypertext Mark up Language)

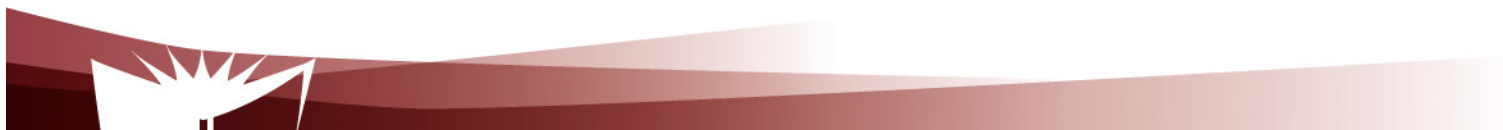
- Page HTML
 - Balises
 - `<html>` et `</html>`
 - En-tete
 - Balises
 - `<head>` et `</head>`
 - Corps
 - Balises
 - `<body>` et `</body>`



Documents statiques

HTML (Hypertext Mark up Language)

- Page HTML
 - Certaines balises peuvent avoir des attributs



Documents statiques: Balises HTML

Tag	Description
<html> ... </html>	Declares the Web page to be written in HTML
<head> ... </head>	Delimits the page's head
<title> ... </title>	Defines the title (not displayed on the page)
<body> ... </body>	Delimits the page's body
<h <i>n</i> > ... </h <i>n</i> >	Delimits a level <i>n</i> heading
 ... 	Set ... in boldface
<i> ... </i>	Set ... in italics
<center> ... </center>	Center ... on the page horizontally
 ... 	Brackets an unordered (bulleted) list
 ... 	Brackets a numbered list
	Starts a list item (there is no)
 	Forces a line break here
<p>	Starts a paragraph
<hr>	Inserts a Horizontal rule
	Displays an image here
 ... 	Defines a hyperlink



Telecommunication Services Engineering (TSE) Lab


Documents statiques

(a) HTML pour Web page. (b) page formatée

```
<html>
<head><title> AMALGAMATED WIDGET, INC. </title> </head>
<body> <h1> Welcome to AWI's Home Page</h1>
 <br>
We are so happy that you have chosen to visit <b> Amalgamated Widget's </b>
home page. We hope <i> you </i> will find all the information you need here.
<p>Below we have links to information about our many fine products.
You can order electronically (by WWW), by telephone, or by fax. </p>
<hr>
<h2> Product information </h2>
<ul>
  <li> <a href="http://widget.com/products/big"> Big widgets </a>
  <li> <a href="http://widget.com/products/little"> Little widgets </a>
</ul>
<h2> Telephone numbers</h2>
<ul>
  <li> By telephone: 1-800-WIDGETS
  <li> By fax: 1-415-765-4321
</ul>
</body>
</html>
```

(a)

Welcome to AWI's Home Page



We are so happy that you have chosen to visit **Amalgamated Widget's** home page. We hope you will find all the information you need here.

Below we have links to information about our many fine products. You can order electronically (by WWW), by telephone, or by FAX.

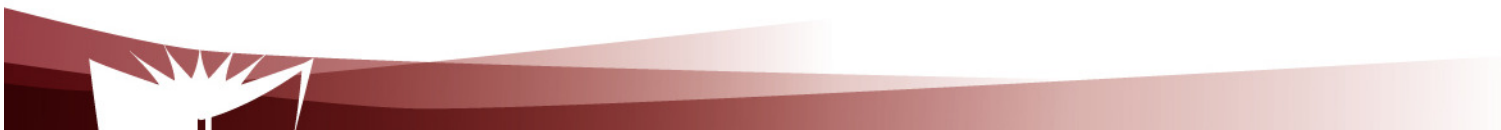
Product Information

- Big widgets
- [Little widgets](#)

Telephone numbers

- 1-800-WIDGETS
- 1-415-765-4321

(b)



Documents

Statiques

```
<html>
<head> <title> A sample page with a table </title> </head>
<body>
<table border=1 rules=all>
<caption> Some Differences between HTML Versions </caption>
<col align=left>
<col align=center>
<col align=center>
<col align=center>
<col align=center>
<tr> <th>Item <th>HTML 1.0 <th>HTML 2.0 <th>HTML 3.0 <th>HTML 4.0 </tr>
<tr> <th> Hyperlinks <td> x <td> x <td> x <td> x </tr>
<tr> <th> Images <td> x <td> x <td> x <td> x </tr>
<tr> <th> Lists <td> x <td> x <td> x <td> x </tr>
<tr> <th> Active Maps and Images <td> &nbsp; <td> x <td> x <td> x </tr>
<tr> <th> Forms <td> &nbsp; <td> x <td> x <td> x </tr>
<tr> <th> Equations <td> &nbsp; <td> &nbsp; <td> x <td> x </tr>
<tr> <th> Toolbars <td> &nbsp; <td> &nbsp; <td> x <td> x </tr>
<tr> <th> Tables <td> &nbsp; <td> &nbsp; <td> x <td> x </tr>
<tr> <th> Accessibility features <td> &nbsp; <td> &nbsp; <td> &nbsp; <td> x </tr>
<tr> <th> Object embedding <td> &nbsp; <td> &nbsp; <td> &nbsp; <td> x </tr>
<tr> <th> Scripting <td> &nbsp; <td> &nbsp; <td> &nbsp; <td> x </tr>
</table>
</body>
</html>
```

(a)

(a) Table.

(b) Page WEb

Some Differences between HTML Versions

Item	HTML 1.0	HTML 2.0	HTML 3.0	HTML 4.0
Hyperlinks	x	x	x	x
Images	x	x	x	x
Lists	x	x	x	x
Active Maps and Images		x	x	x
Forms		x	x	x
Equations			x	x
Toolbars			x	x
Tables			x	x
Accessibility features				x
Object embedding				x
Scripting				x

(b)

Telecommunication Services Engineering (TSE) Lab

Documents Statiques

(a) Formulaire HTML

(b) Version formattée.

```
<html>
<head> <title> AWI CUSTOMER ORDERING FORM </title> </head>
<body>
<h1> Widget Order Form </h1>
<form ACTION="http://widget.com/cgi-bin/widgetorder" method=POST>
<p> Name <input name="customer" size=46> </p>
<p> Street Address <input name="address" size=40> </p>
<p> City <input name="city" size=20> State <input name="state" size =4>
Country <input name="country" size=10> </p>
<p> Credit card # <input name="cardno" size=10>
Expires <input name="expires" size=4>
M/C <input name="cc" type=radio value="mastercard">
VISA <input name="cc" type=radio value="visacard"> </p>
<p> Widget size Big <input name="product" type=radio value="expensive">
Little <input name="product" type=radio value="cheap">
Ship by express courier <input name="express" type=checkbox> </p>
<p><input type=submit value="submit order"> </p>
Thank you for ordering an AWI widget, the best widget money can buy!
</form>
</body>
</html>
```

(a)

Widget Order Form

Name

Street address

City State Country

Credit card # Expires M/C ☐ Visa ☐

Widget size Big ☐ Little ☐ Ship by express courier ☐

Thank you for ordering an AWI widget, the best widget money can buy!

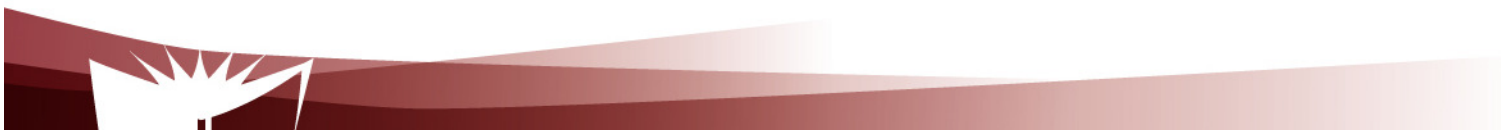
(b)



Documents statiques

Réponse possible du navigateur avec formulaire rempli par utilisateur.

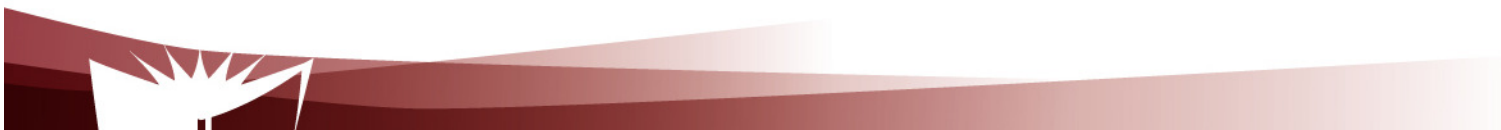
```
customer=John+Doe&address=100+Main+St.&city=White+Plains&  
state=NY&country=USA&cardno=1234567890&expires=6/98&cc=mastercard&  
product=cheap&express=on
```



Documents statiques

HTML (Hypertext Mark up Language)

- Les problèmes
 - Mélange contenu et instructions de formattage
 - Ne permet pas de structurer les documents
 - Alternative récente
 - XML

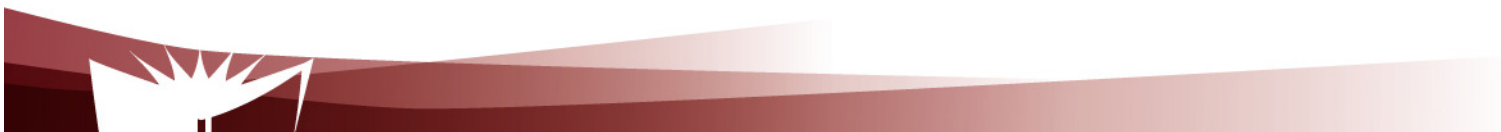


Documents statiques

XML est un langage à balise pour les documents qui contiennent de l'information structurée

XML utilise les balises comme HTML.

- **Mais les balises de XML sont extensible**

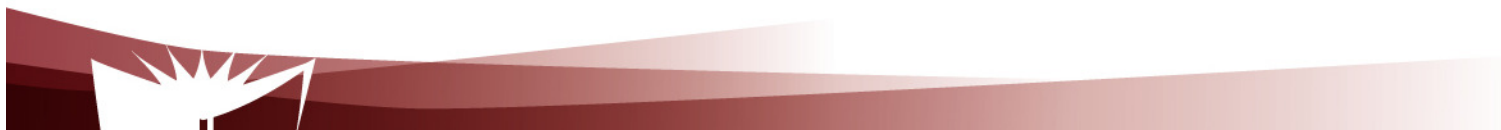


Documents statiques

XML documents

Schema and DTD (Document Type Definition)

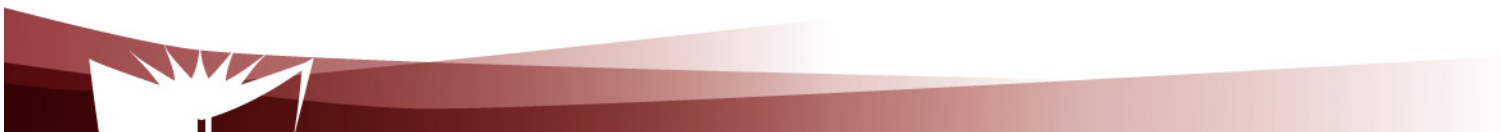
- Défini les balises qui sont autorisées



Documents statiques

XML documents Example

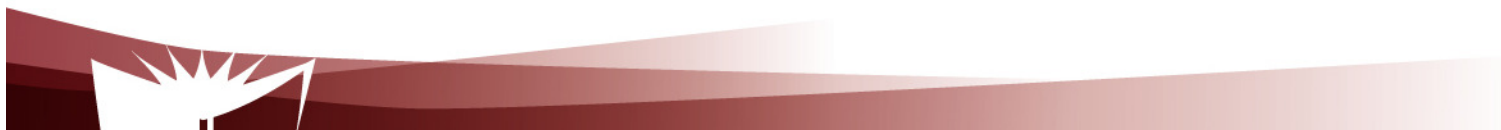
```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<book>  
  <title>Understanding Web Services</title>  
  <author>Eric Newcomer</author>  
  <price>39.99</price>  
</book>
```



Documents statiques

Processeur XML

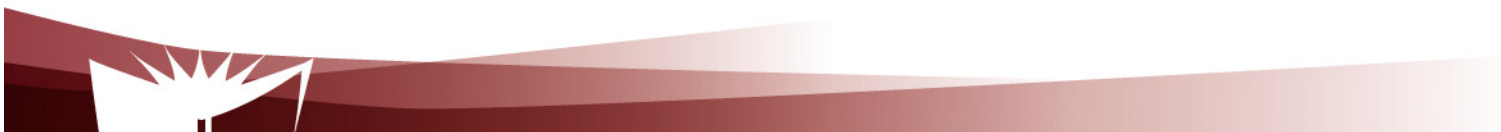
- Lire et analyser les documents XML
- Donner accès au contenu
- Permettre de les éditer et de les générer
-



Documents statiques

Processeur XML

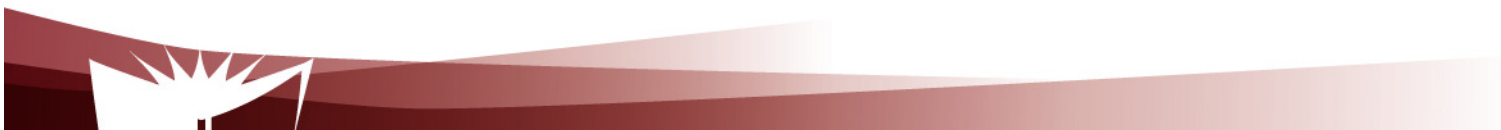
- APIs les plus populaires
 - Document Object Model (DOM) from W3C
 - Simple API for XML (SAX) – From XML-DEV mailing list



Documents dynamiques

Problématique: Contenu dynamique

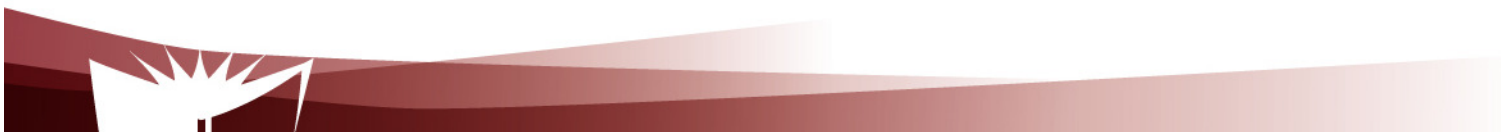
- Base de données
- Page qui a besoin d'être générée on the fly
 - Ex: Nombre de visiteurs sur site
- Utilisation de script (CGI) et aussi de servlets Java



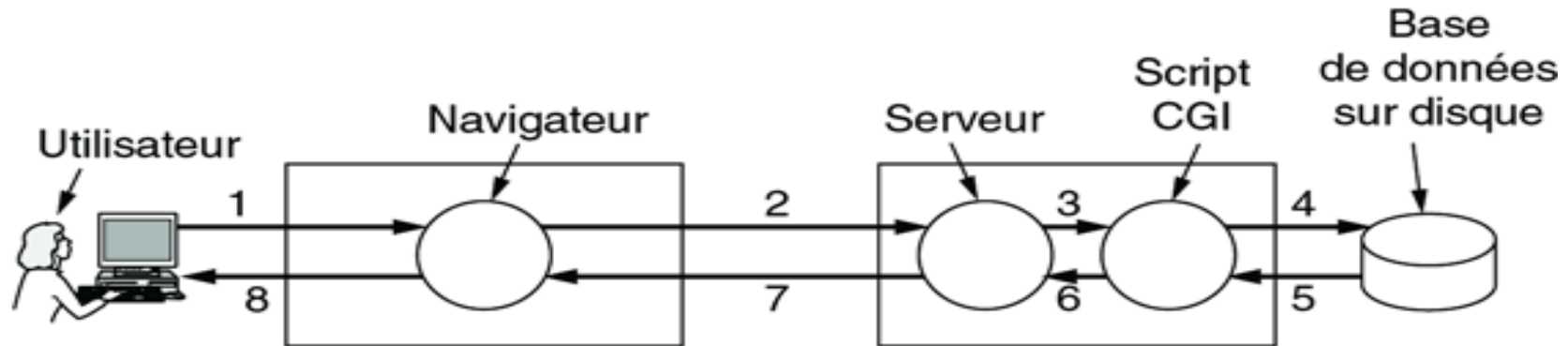
Documents dynamiques

Les étapes

1. Remplissage du formulaire
2. Formulaire envoyée au serveur Web
3. Transfert au script CGI ou au java servlet
4. Le script ou le servlet génère le contenu demandé
(ex: interroge la base de données)
5. Page retournée

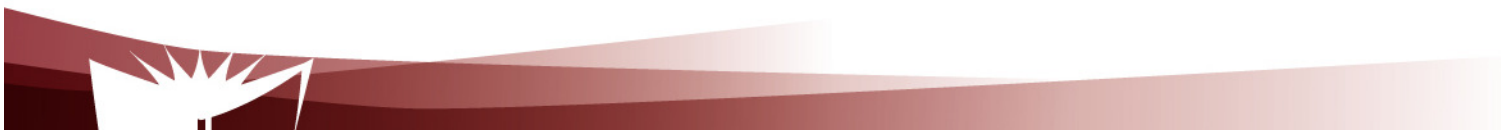


Documents dynamiques



1. Remplissage du formulaire
2. Formulaire retourné
3. Transfert au script CGI
4. Le script interroge la BDD
5. Enregistrement trouvé
6. Le script construit la page
7. Page retournée
8. Page affichée

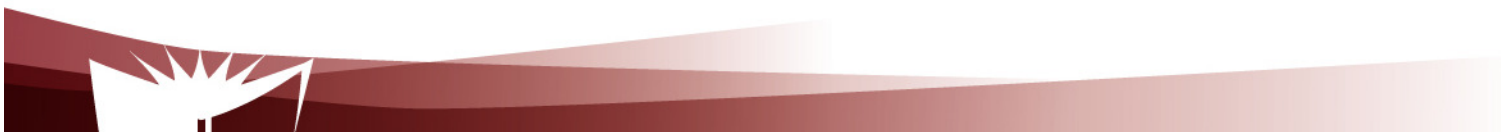
© Pearson Education France



Documents dynamiques

Les problèmes de l'approche script CGI

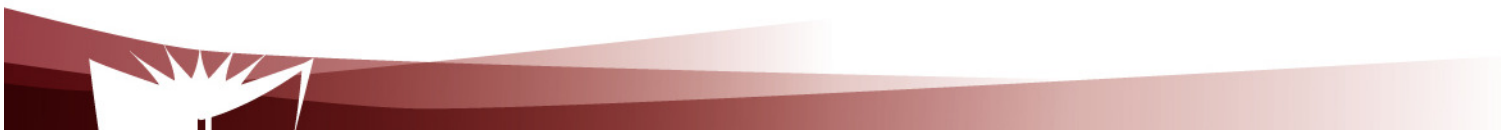
1. Le script doit s'exécuter sur le serveur Web (Peut poser des problèmes de montée en charge – scalability)
2. Les bases de données doivent être re-ouvertes et re-fermées à chaque requête (Problèmes de performance)



servlet API

Creation de contenu dynamique

- Servlet
 - Composante Java
 - Generation de contenu on the fly exactement comme CGI
 - interface entre requêtes HTTP et base de données
 - Formulaires
 - Date, nombre de visiteurs

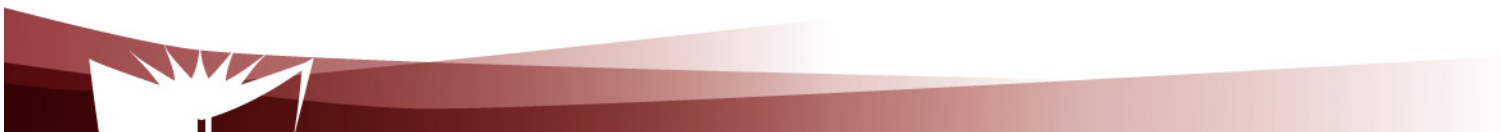


Telecommunication Services Engineering (TSE) Lab

servlet API

Conteneur servlets - Servlet container (aussi connu sous le nom de servlet engine)

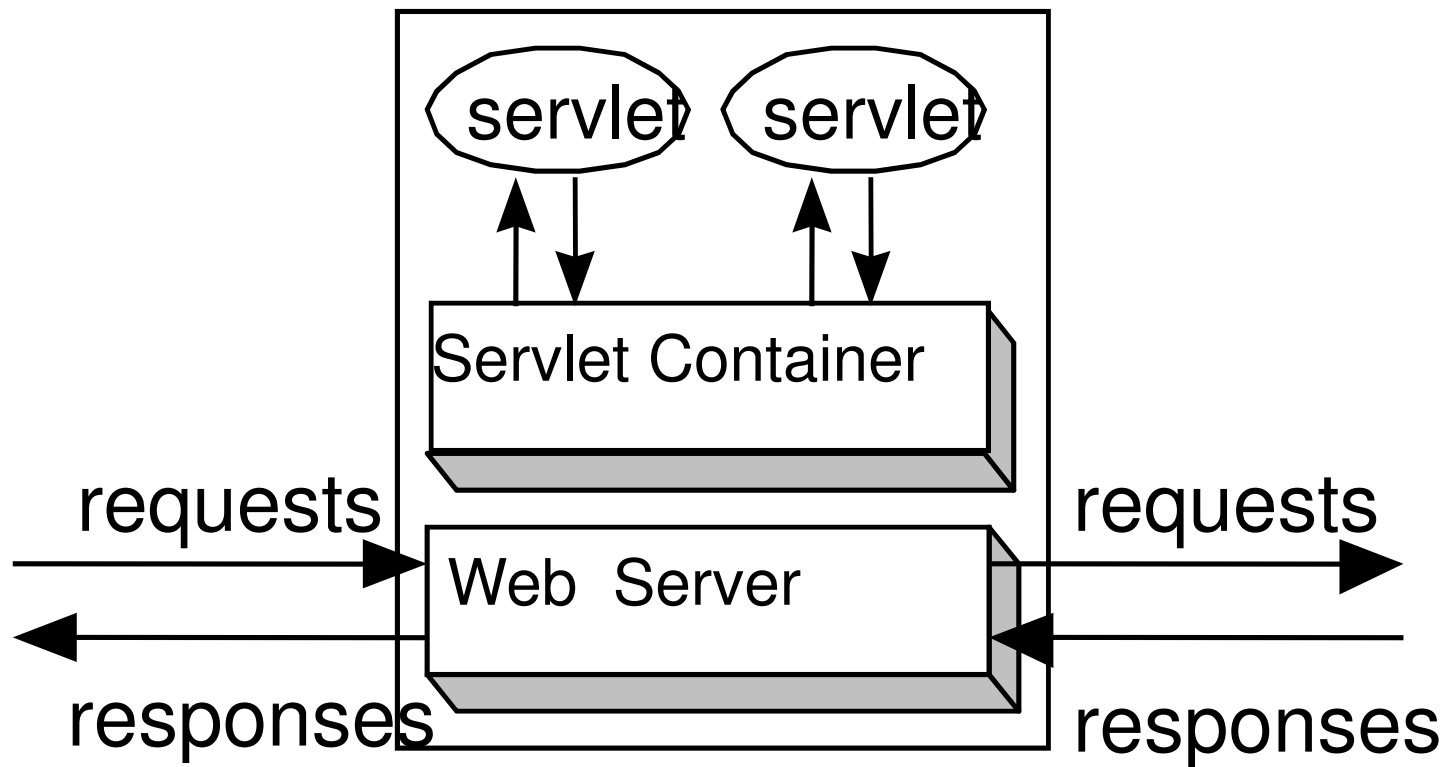
- Contains the servlets et les gère durant leur cycle de vie
 - Creation
 - Initialisation
 - Destruction
 - Recoit et decode des requêtes HTTP
 - Encode et renvoie les réponses



Telecommunication Services Engineering (TSE) Lab

Conteneur de servlets

Un exemple ou le conteneur et le serveur Web sont sur la même machine

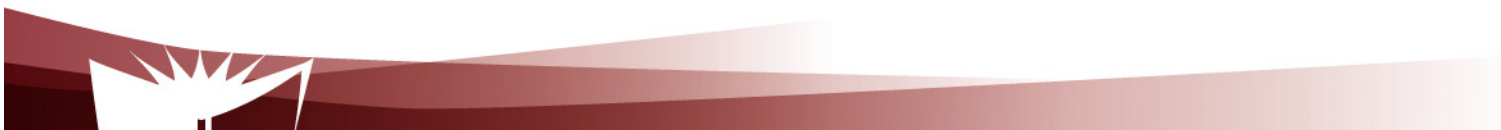


Documents dynamiques

Autres méthodes à part CGI et Servlets

1. PHP: Hypertext preprocessor

- Script Incorporé dans le fichier HTML lui-même
- Balise `<?php ?>`



Documents dynamiques

Un exemple

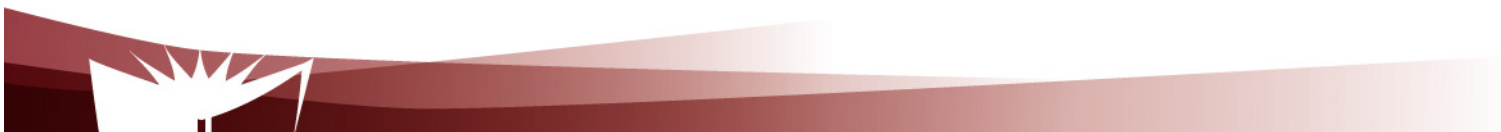
```
<body>
```

```
<h2> This is what I know about you </h2>
```

```
<?php echo $HTTP_USER_AGENT ?>
```

```
</body>
```

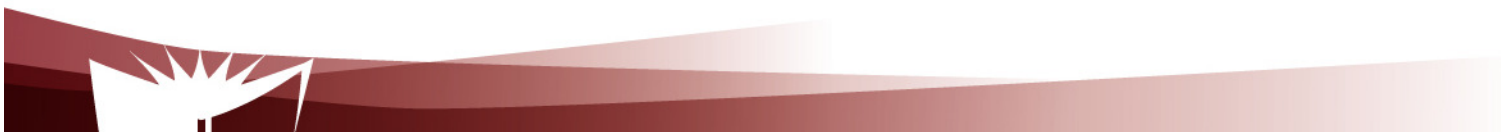
```
</html>
```



Documents dynamiques

Reponse:

- Navigateur
- Système d exploitation



Documents Dynamiques

```
<html>
<body>
<form action="action.php" method="post">
<p> Please enter your name: <input type="text" name="name"> </p>
<p> Please enter your age: <input type="text" name="age"> </p>
<input type="submit">
</form>
</body>
</html>
```

(a)

```
<html>
<body>
<h1> Reply: </h1>
Hello <?php echo $name; ?>.
Prediction: next year you will be <?php echo $age + 1; ?>
</body>
</html>
```

(b)

```
<html>
<body>
<h1> Reply: </h1>
Hello Barbara.
Prediction: next year you will be 25
</body>
</html>
```

(c)

(a) A Web page containing a form. (b) A PHP script for handling the output of the form. (c) Output from the PHP script when the inputs are "Barbara" and 24 respectively.



Documents dynamiques

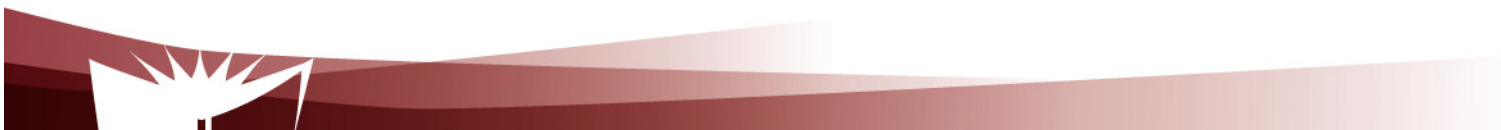
Autres méthodes à part CGI et Servlets

1. Java server page (JSP)

- Similaire a PHP mais écrit en Java

2. Active Server Page

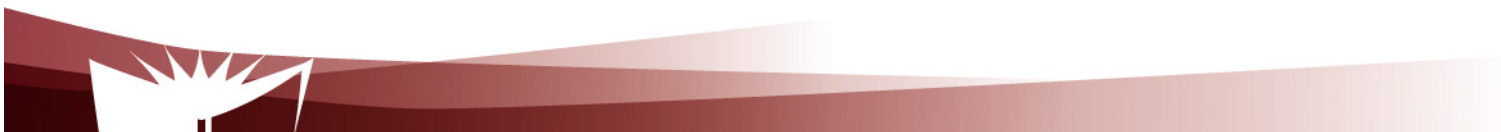
- Version de Microsoft, mais écrit en Visual basic



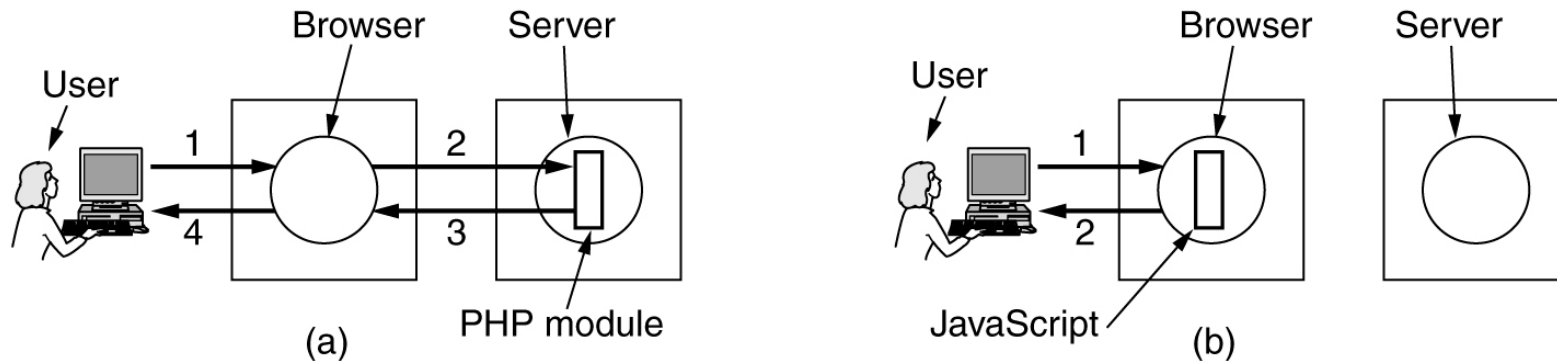
Documents dynamiques

Le contenu dynamique peut aussi être généré du côté client

1. Java script
2. Applets



Documents dynamiques



(a) Server-side scripting with PHP.

(b) Client-side scripting with JavaScript.

Telecommunication Services Engineering (TSE) Lab

Documente dynamique

```
<head>
<script language="javascript" type="text/javascript">
function response(test form) {
    var person = test form.name.value;
    var years = eval(test form.age.value) + 1;
    document.open();
    document.writeln("<html> <body>");
    document.writeln("Hello " + person + ".<br>");
    document.writeln("Prediction: next year you will be " + years + ".");
    document.writeln("</body> </html>");
    document.close();
}
</script>
</head>

<body>
<form>
Please enter your name: <input type="text" name="name">
<p>
Please enter your age: <input type="text" name="age">
<p>
<input type="button" value="submit" onclick="response(this.form)">
</form>
</body>
</html>
```

Use of JavaScript
for processing a
form.

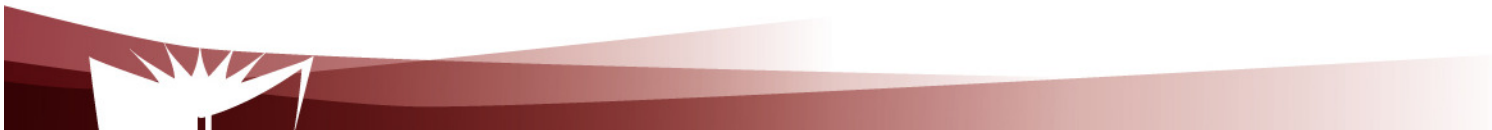


Documents dynamiques

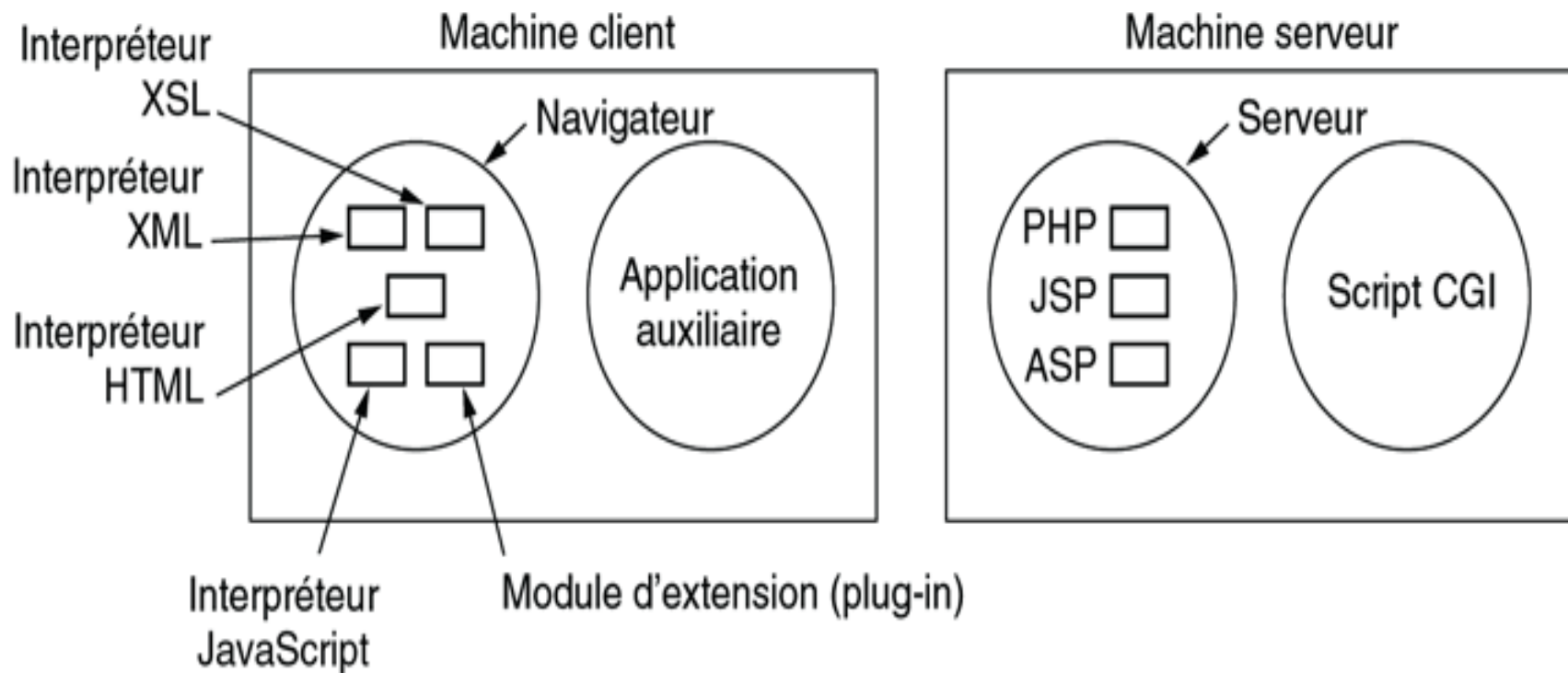
```
<html>
<head>
<script language="javascript" type="text/javascript">

function response(test_form) {
A J   function factorial(n) {if (n == 0) return 1; else return n * factorial(n - 1);}
f;   var r = eval(test_form.number.value);    // r = typed in argument
    document.myform.mytext.value = "Here are the results.\n";
    for (var i = 1; i <= r; i++)                // print one line from 1 to r
        document.myform.mytext.value += (i + "! = " + factorial(i) + "\n");
    }
</script>
</head>

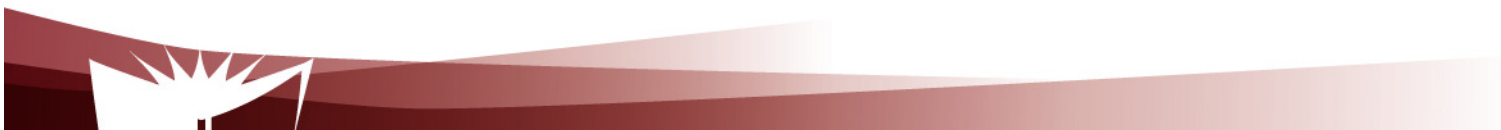
<body>
<form name="myform">
Please enter a number: <input type="text" name="number">
<input type="button" value="compute table of factorials" onclick="response(this.form)">
<p>
<textarea name="mytext" rows=25 cols=50> </textarea>
</form>
</body>
</html>
```



Documents dynamiques

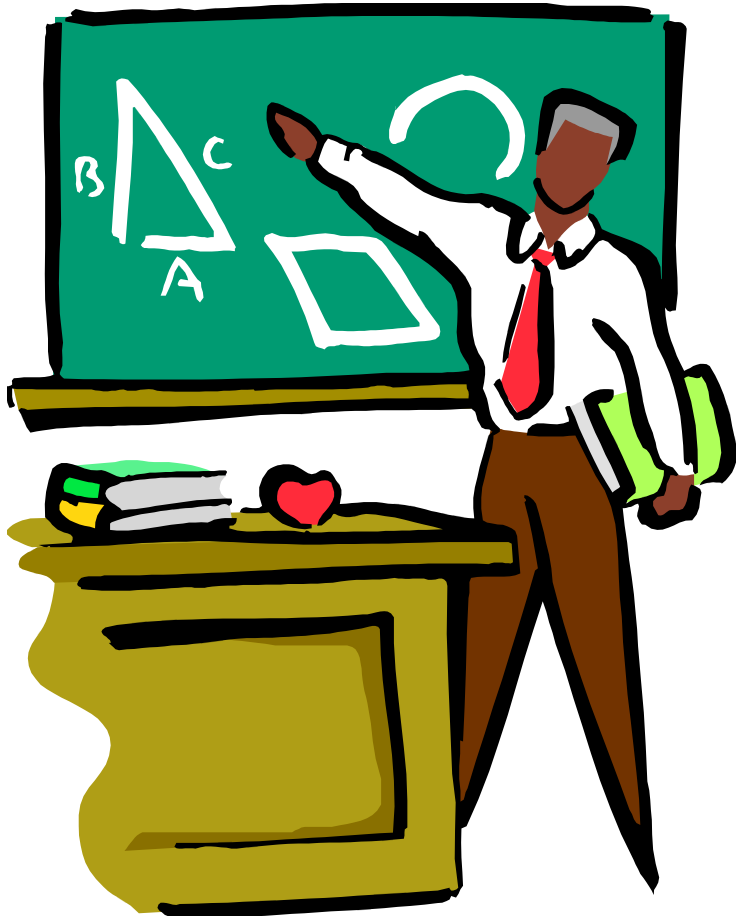


© Pearson Education France

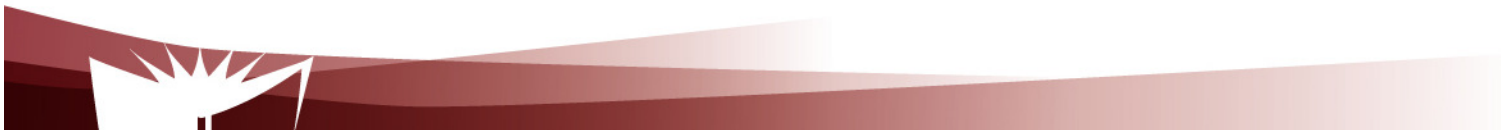


Telecommunication Services Engineering (TSE) Lab

HTTP



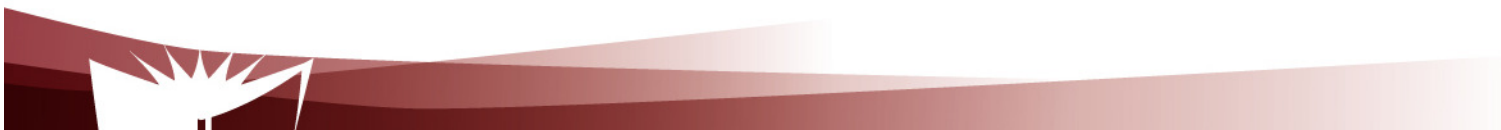
- 1 - Introduction
- 2. Entités fonctionnelles
- 3. Messages
- 4. Méthodes



Introduction

HTTP (HyperText Transfer Protocol)

- Protocole du niveau applicatif pour des systèmes d'information hypermédia distribués et collaboratifs
- Date de 1990
- Protocole requête / réponse avec les requêtes portant sur des ressources
 - Ressource: objet ou service de réseau qui peut être identifié par un URI (Universal Resource Identifier)
 - Multiplicité de formats, taille ...etc



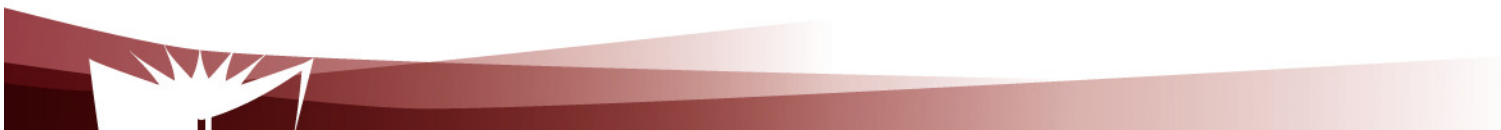
Entités fonctionnelles

Client

- Programme qui établit une connexion dans le but d'envoyer des requêtes

Agent utilisateur

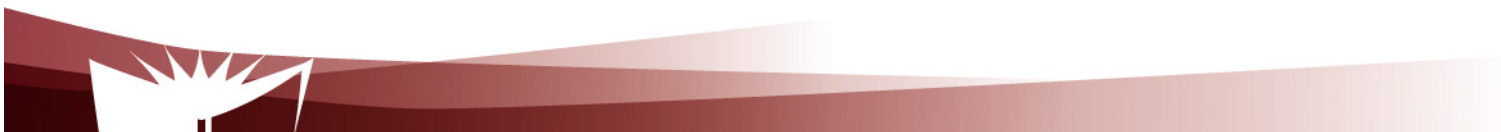
- Client qui est à l'origine de la requête (ex: navigateur)
- Note
 - Une requête peut transiter par plusieurs serveurs



Entités fonctionnelles

Serveur

- Un programme qui accepte des connexions dans le but de servir des requêtes en envoyant des réponses
- Un même programme peut agir comme client et serveur
- Le rôle dépend des connexions



Entités fonctionnelles

- **Serveur origine (Origin server)**
 - Serveur où réside la ressource
- **Serveur proxy**
 - Programme intermédiaire qui agit comme client et comme serveur
 - Points de transit
- **Serveur passerelle (Gateway server)**
 - Agit tout comme un serveur origine, mais transmet la requête à un autre serveur
 - Transparence vis à vis du client



Messages

HTTP-message = Request | Response

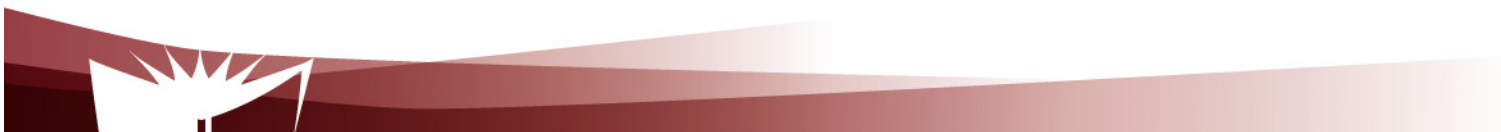
generic-message = start-line

*(message-header CRLF)

CRLF

[message-body]

start-line = Request-Line | Status-Line



Les méthodes (requêtes)

HEAD

- Demande des informations sur la page, mais sans la page (ex: dernière modification)

GET

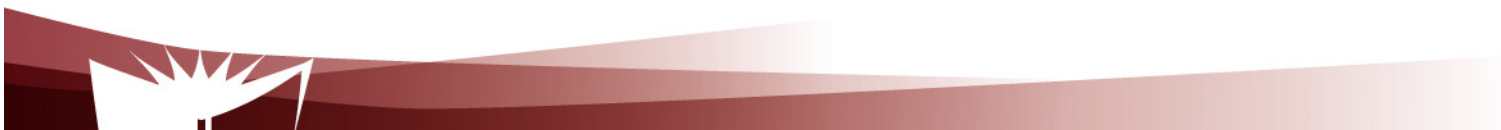
- Demande la page

PUT

- Inverse de GET

POST

- Ajoute de l'information à la page (groupe de discussions)



Les méthodes (requêtes)

DELETE

- Supprime la page

GET

- Demande la page

TRACE

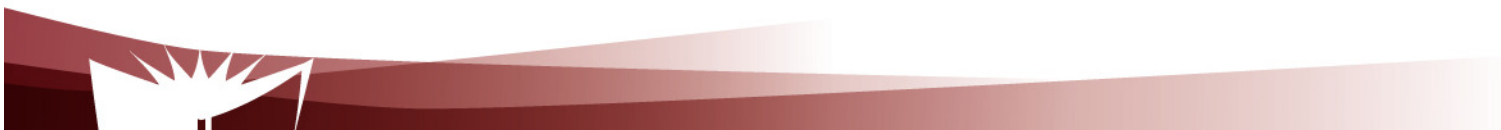
- Debogage

OPTIONS

- Permet de demander les options que le serveur supporte

CONNECT

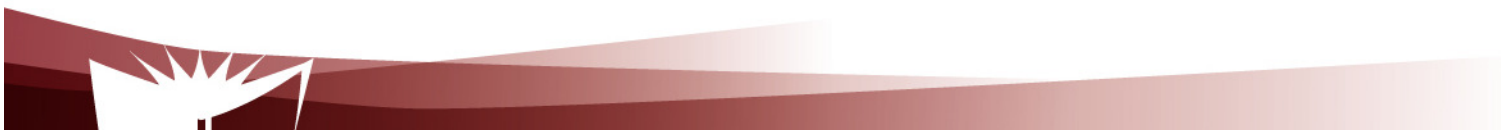
- Pas utilisé pour l'instant



Les méthodes (Requêtes)

The

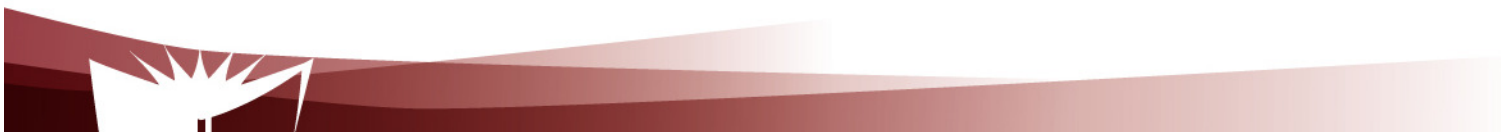
Method	Description
GET	Request to read a Web page
HEAD	Request to read a Web page's header
PUT	Request to store a Web page
POST	Append to a named resource (e.g., a Web page)
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Reserved for future use
OPTIONS	Query certain options



Les méthodes (réponses)

The status code response groups.

Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later



Telecommunication Services Engineering (TSE) Lab

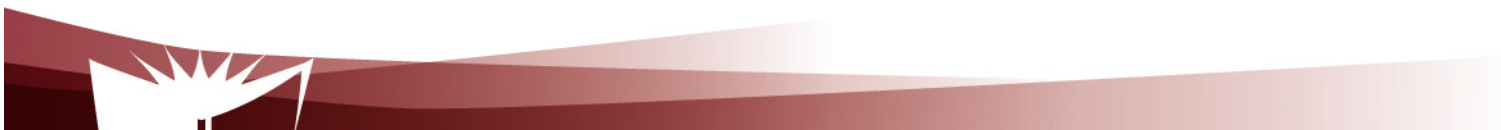
Les en-têtes

Header	Type	Contents
User-Agent	Request	Information about the browser and its platform
Accept	Request	The type of pages the client can handle
Accept-Charset	Request	The character sets that are acceptable to the client
Accept-Encoding	Request	The page encodings the client can handle
Accept-Language	Request	The natural languages the client can handle
Host	Request	The server's DNS name
Authorization	Request	A list of the client's credentials
Cookie	Request	Sends a previously set cookie back to the server
Date	Both	Date and time the message was sent
Upgrade	Both	The protocol the sender wants to switch to
Server	Response	Information about the server
Content-Encoding	Response	How the content is encoded (e.g., gzip)
Content-Language	Response	The natural language used in the page
Content-Length	Response	The page's length in bytes
Content-Type	Response	The page's MIME type
Last-Modified	Response	Time and date the page was last changed
Location	Response	A command to the client to send its request elsewhere
Accept-Ranges	Response	The server will accept byte range requests
Set-Cookie	Response	The server wants the client to save a cookie

Some HTTP message headers.

Références

1. Livre de référence, chapitre 7
2. HTTP RFC





UNIVERSITÉ
Concordia

UNIVERSITY

www.ciise.concordia.ca