

Programmation Web

ENSIBS Cyberlog/Cyberdata 3^{ème} année 2023-2024

Mazen EL-SAYED

mazen.el-sayed@univ-ubs.fr

Organisation du cours

- 6h de CM + 15h de TP
- Evaluation
 - Compte rendu des TP
 - Projet rendu à la fin du module

Références

3

- Christophe AUBRY - HTML5 et CSS3- eni 2020
- Olivier Glück - support de cours Université LYON 1/Département Informatique - <http://perso.univ-lyon1.fr/olivier.gluck>
- Philippe Genoud - support de cours - Université Grenoble Alpes - <https://lig-membres.imag.fr/genoud/teaching/PL2AI/cours/cours.html>
- <https://www.w3schools.com/>

Plan de la matière

4

1. Introduction
2. Le langage HTML
3. Le langage CSS
4. Le langage javascript
5. Ajax et jQuery
6. Node js

Chapitre 1 : Introduction

Le Web

6

- Le World Wide Web (WWW ou W3 est connu également sous le nom Web) **est un système de documents hypertextes interconnectés et accessibles via Internet.**
- Qu'est ce que l'hypertexte ?
 - Définition : **document non linéaire** constitué de pages permettant de passer de l'une à l'autre par des liens (liens hypertextes)
- Web = hypertexte + internet
 - Web : Ensemble de pages hypertextes accessibles à travers l'internet
- Web ≠ Internet qui supporte de nombreux autres services
 - messagerie électronique (ou email) : POP ou IMAP
 - transfert de fichiers : FTP
 - ...

Qu'est ce qu'une page web ?

7

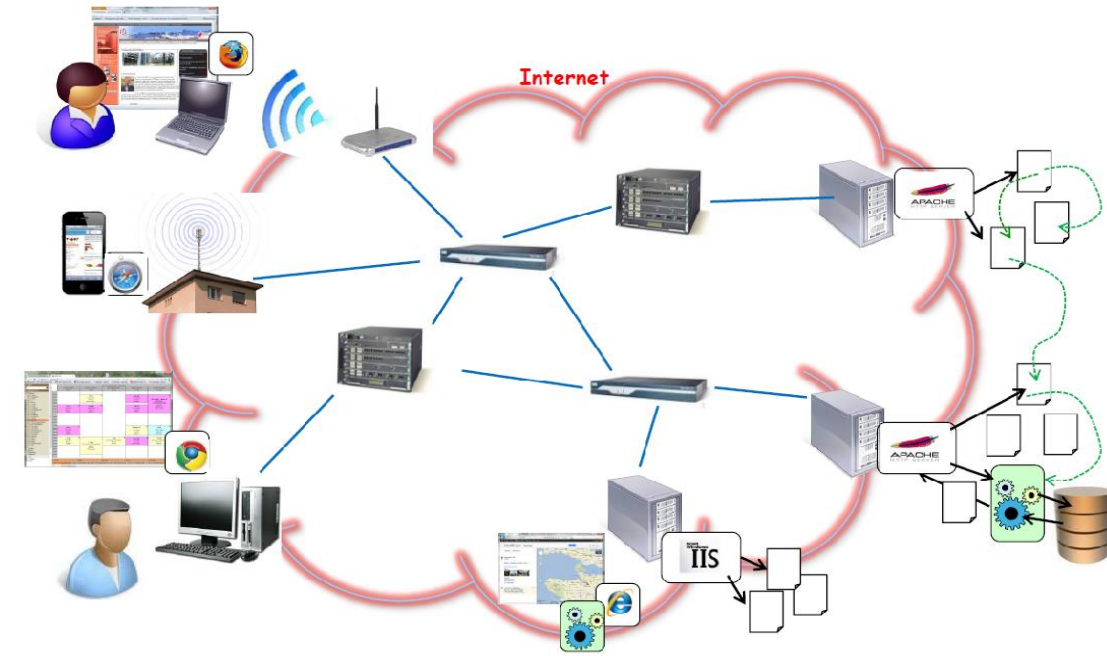
Une page web est un document numérique qui est

- composé de
 - textes, sons, vidéos, images, ...
 - liens hypertextes vers d'autres documents permettant de passer d'un document à un autre
- consultable à l'aide d'un navigateur (browser)
 - firefox, chrome, safari, edge, ...

Internet

8

- Internet :
 - réseau mondial connectant des ordinateurs
 - En fait un réseau de réseaux d'ordinateurs
- Adresse ou Numéro IP (Internet Protocol) : identifie chaque ordinateur par une adresse unique
 - adresse IPv4 sur 32 bits (4 octets) : 172.100.100.100



Nom du domaine et serveur DNS

9

- Les équipements (ou hôtes) connectés à un réseau IP, comme Internet, possèdent une adresse IP qui les identifie sur le réseau.
- Ces adresses sont numériques afin de faciliter leur traitement par les machines.
 - Difficile à retenir par les usagers
- Pour faciliter l'accès aux hôtes sur un réseau IP, un mécanisme a été mis en place pour associer un nom à une adresse IP. Ce nom, plus simple à retenir qu'une suite de chiffres, est appelé « nom de domaine ».
- C'est plus facile d'utiliser le nom du domaine www.monSite.com à la place de l'adresse IP associée
- Résoudre un nom de domaine consiste à trouver l'adresse IP qui lui est associée.
- C'est le rôle du serveur DNS (Domain Name System)
 - Il permet d'établir la correspondance entre les adresses IP des machines et leurs noms

Consulter une page web

10

Principe de base : Client/Serveur

Machine Client
(ex : votre PC)

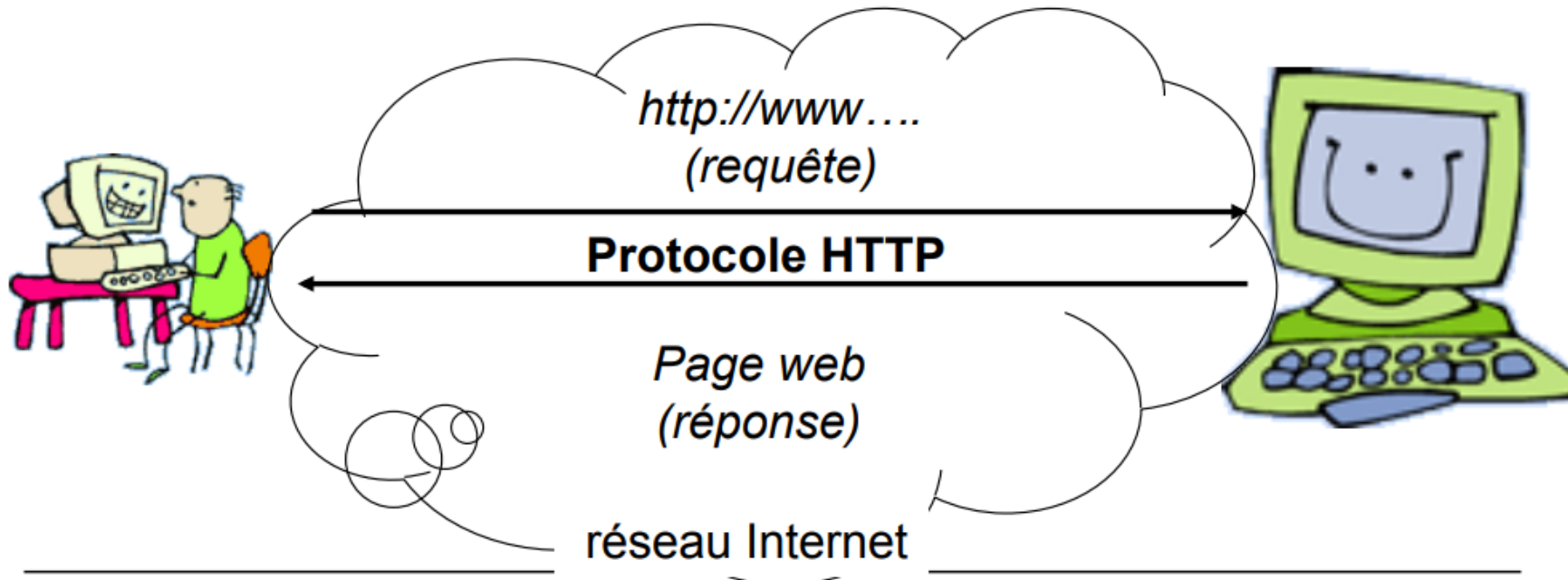
+

un logiciel « client web »
(ex : Firefox, Chrome, Safari, IE,...)

Machine Serveur
(univ-ubs.fr)

+

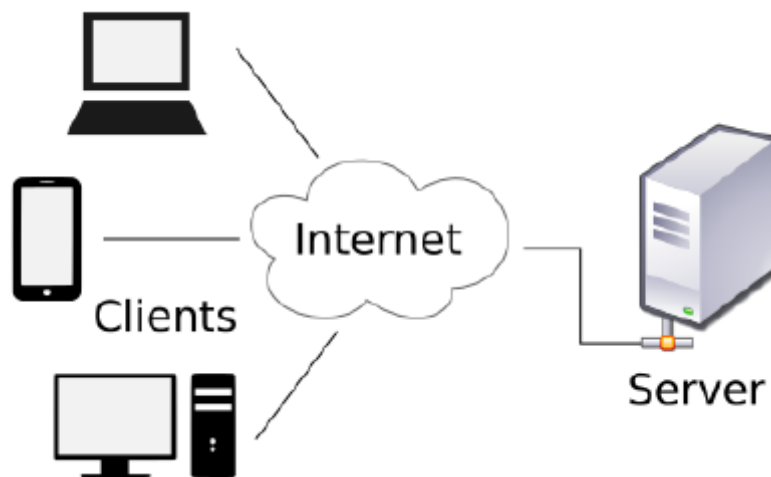
un logiciel « serveur web »
(ex : Apache, IIS)



Le protocole HTTP

11

- HTTP (HyperText Transfer Protocol)
- Protocole de communication, de la couche application dans le modèle OSI. Il permet le transfert client/serveur
 - un client HTTP
 - celui qui demande à accéder (demande de récupérer) **une ressource**, le navigateur
 - toujours à l'origine d'une communication
 - un serveur HTTP
 - celui qui va répondre à la demande (qui va « servir » la ressource)



URL

12

- Une URL (Uniform Resource Locator) est une forme spéciale d'adresse d'une ressource donnée sur Internet.
- Elle peut pointer vers un site web, un document particulier ou une image.
- Autrement dit, grâce à cette adresse, l'utilisateur dispose des coordonnées de l'information dont il a besoin.
- Exemple : <http://www.univ-ubs.fr/dossier1/dossier2/index.html>
 - [http](#) : la méthode d'accès à la ressource, également appelée protocole de communication;
 - [univ-ubs.fr](#) : nom du domaine,
 - [dossier1/dossier2/](#) : le chemin d'accès sur le serveur,
 - [index.html](#) : nom de la page à télécharger (la ressource)

Le navigateur web

13

- L'utilisateur saisit une URL dans le navigateur
- Le navigateur analyse l'URL demandée et récupère le nom du serveur
- Il demande au DNS l'adresse IP de la machine serveur à partir de l'URL
- Se connecte à cette adresse IP sur le port 80
- Fabrique la **requête HTTP** et l'envoie au serveur
- Réceptionne la **réponse HTTP** du serveur
- Interprète le code HTML reçu : commandes de formatage et de mise en forme (police, gras, couleurs...)
- Demande les objets incorporés au serveur (images, css, js) et affiche la page correctement formatée
- Exécute les programmes Javascript s'il y en a

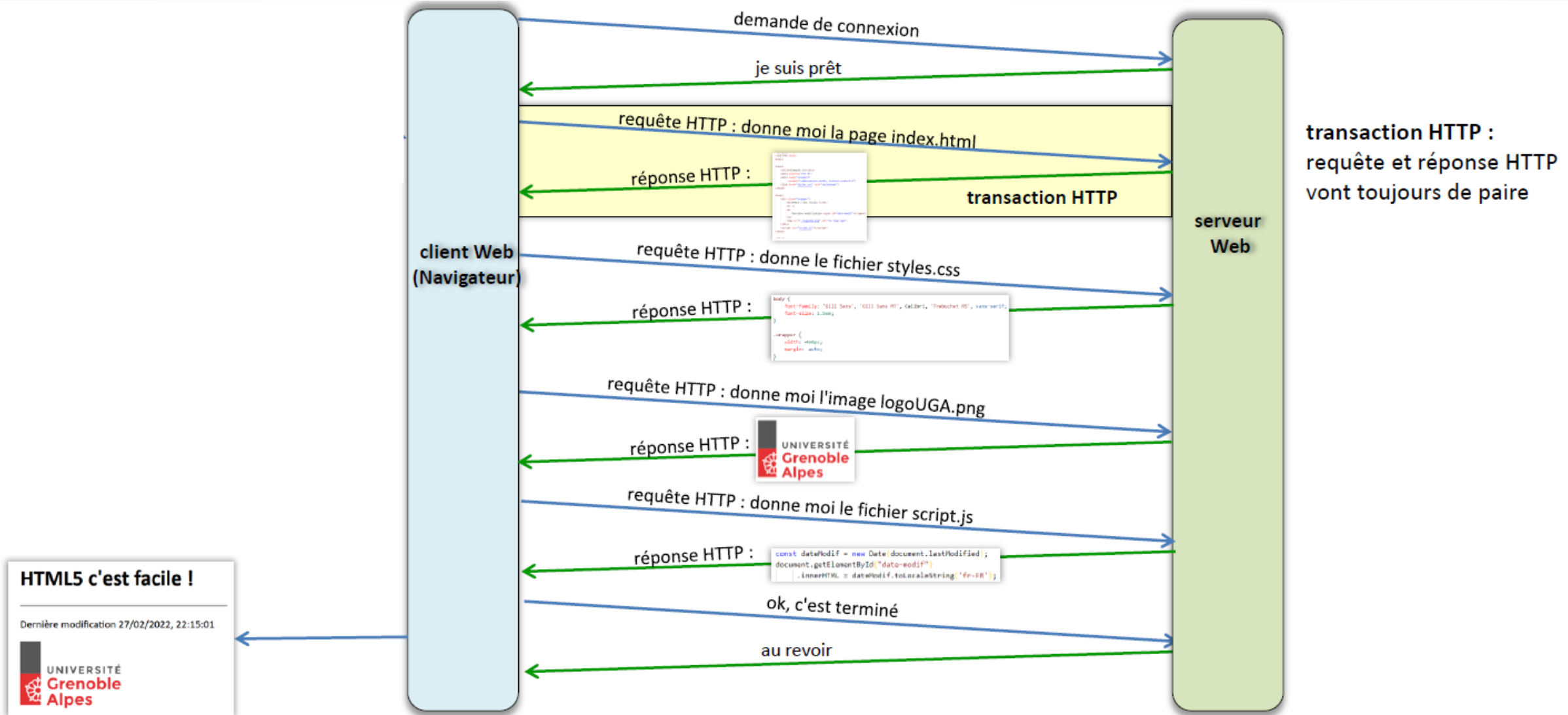
Le serveur web

14

- Serveur HTTP : logiciel qui écoute les requêtes de pages Web formulées par les clients et renvoie les documents demandés
 - Ex : Apache, Microsoft IIS (Internet Information Server), Nginx, ...
- Il vérifie la validité de la requête...
 - Le document demandé peut ne pas exister
 - L'accès à un document peut être restreint (accès restreint)
- ... et y répond si la requête est valide : envoi du texte, des images, de la feuille de styles, du code à exécuter sur le client (Javascript).
- Il peut renvoyer un message d'erreur, une demande d'authentification...
- Il peut exécuter un programme ou un script localement (PHP, javascript 'node js') qui va générer une réponse
 - HTML (page dynamique) en fonction des arguments transmis par le navigateur.
 - d'un autre format tel que text, json, ...

Déroulement du chargement d'une page web

15



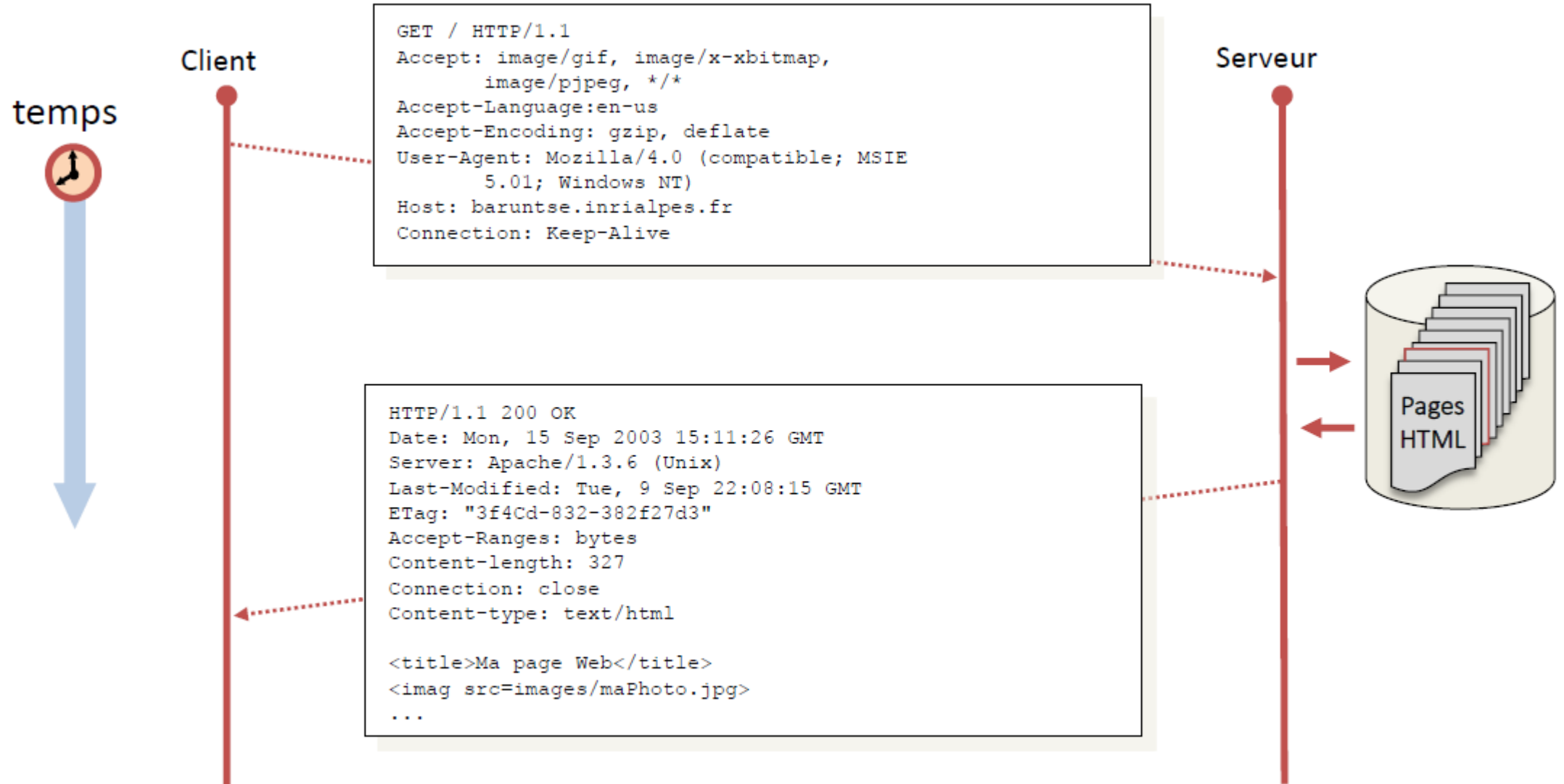
Le protocole HTTP

16

- Message (ou transaction) HTTP :
 - **Une requête HTTP** (envoyée par le client) et **une réponse HTTP** (envoyée par le serveur)
 - Au début la seule opération possible en tant qu'utilisateur était la récupération et la visualisation d'une ressource (document HTML)
 - Avec l'évolution du web d'autres types de ressources (HTML, images, CSS, JavaScript, vidéos, données XML/JSON...) peuvent être récupérées.
 - Le client peut également envoyer ou modifier des informations (formulaires HTML, appels AJAX....).
 - Une requête ne peut récupérer qu'une seule ressource à la fois / n'effectuer qu'une action à la fois

Transaction HTTP

17



Format des messages HTTP

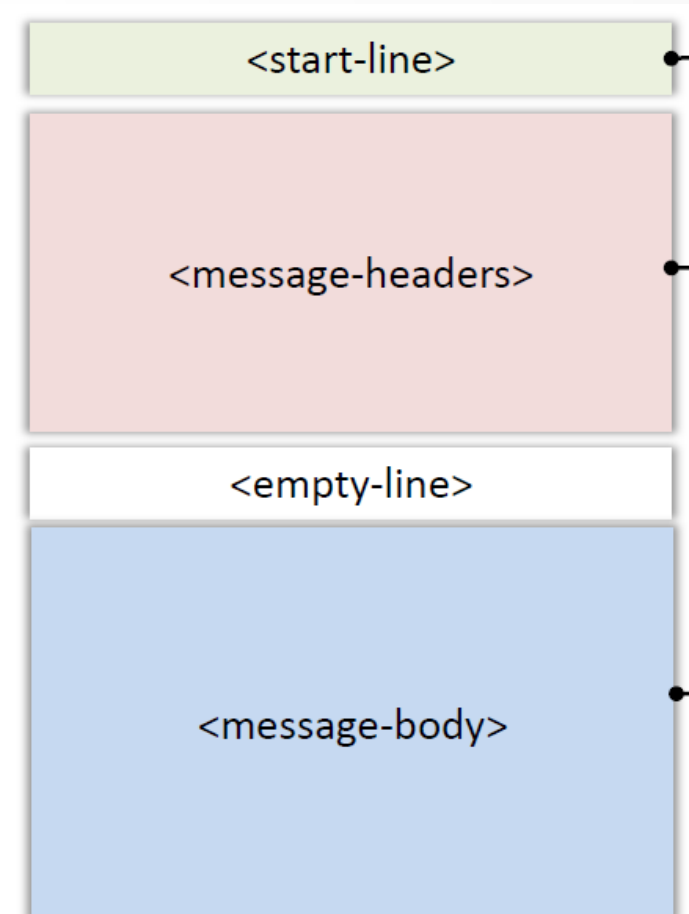
18

- Structure commune partagée par tous les messages HTTP (requêtes et réponses)

Détermine si le message est une requête ou une réponse

Contient l'entête du message

Contient le corps du message. Il peut être vide



Requête HTTP

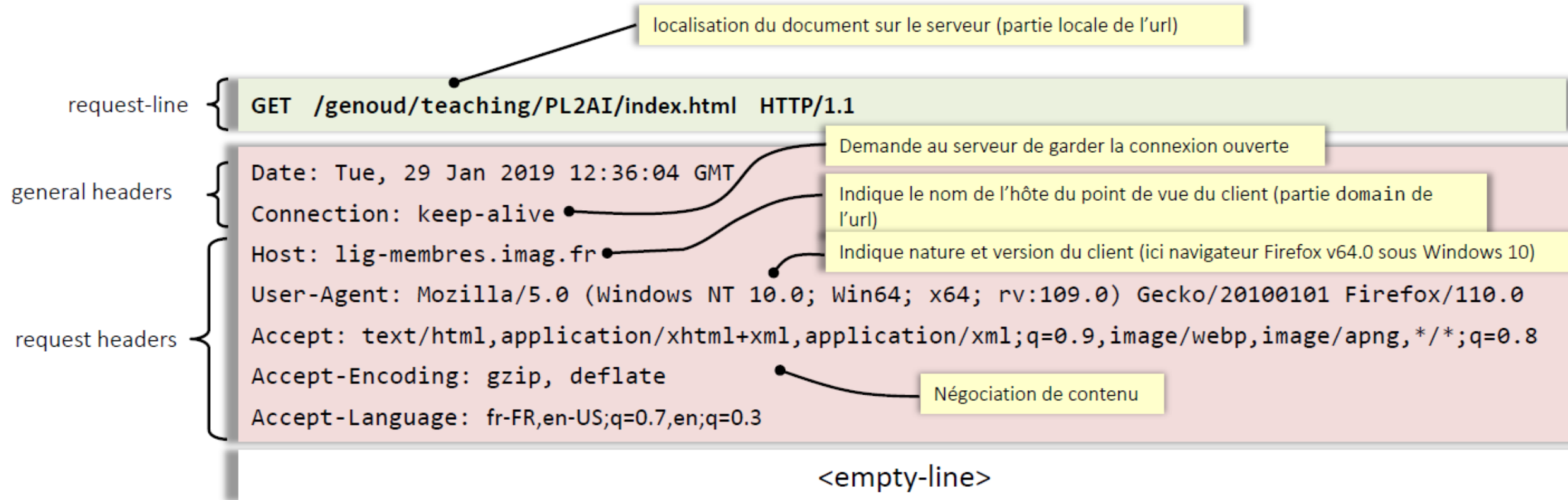
19

Requête HTTP :

- <start-line> est composé de trois informations suivantes : **méthode URI version**
 - URI est l'URI de la ressource
 - Version : version du protocole HTTP
 - Méthode :
 - GET méthode la plus courante pour demander une ressource.
 - POST : utilisée pour soumettre des données en vue d'un traitement à une ressource (typiquement depuis un formulaire HTML). L'URI fournie est l'URI d'une ressource à laquelle s'appliqueront les données envoyées.
 - PUT : permet de remplacer ou d'ajouter une ressource sur le serveur.
 - DELETE : permet de supprimer une ressource du serveur
 - HEAD : ne demande que des informations sur la ressource, sans demander la ressource elle-même.
 - PATCH : permet de modifier partiellement une ressource

Exemple d'une requête Get

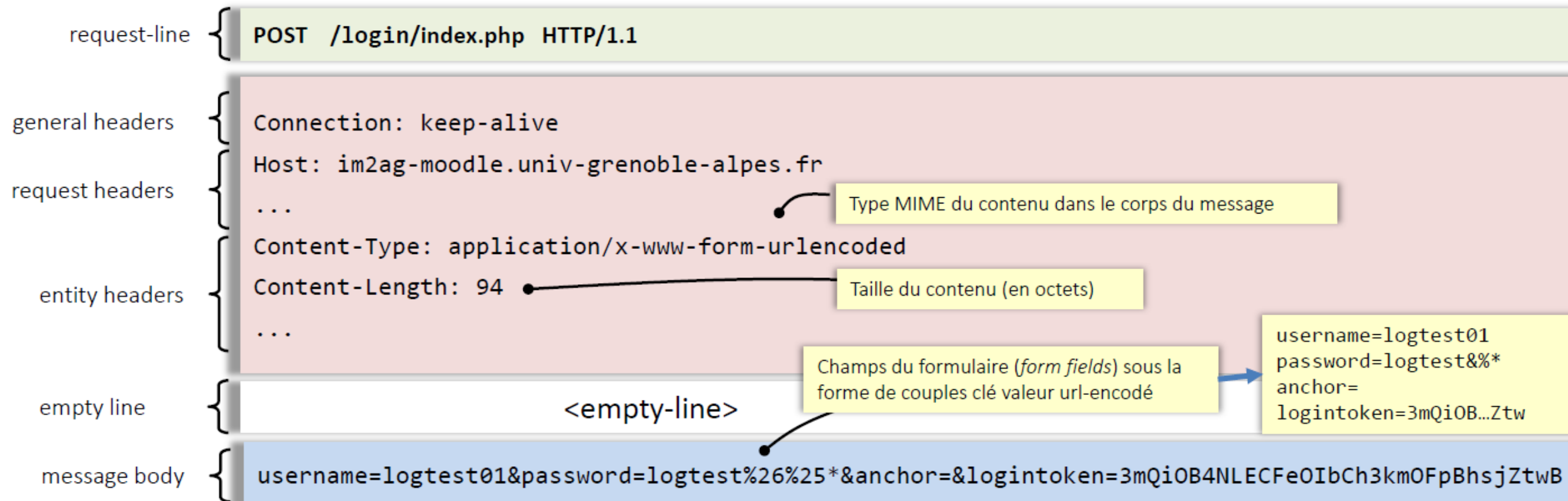
20



entity-headers et message-body sont vides

Exemple d'une requête POST

21



Réponse envoyée par le serveur

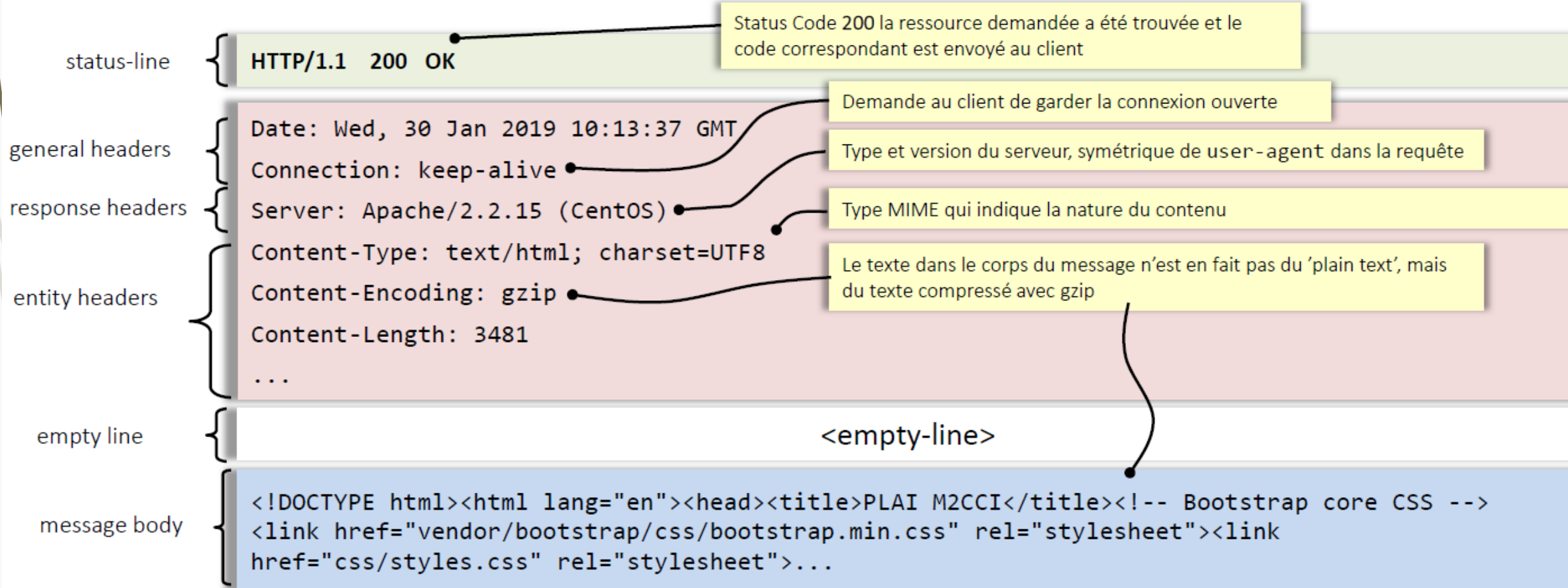
22

- Start-line est composé de trois parties : **version - status code - texte réponse**
 - Version : version de HTTP
 - Texte réponse : texte explicatif
 - Status code est formé de trois chiffres :
 - 2xx : requête du client accomplie avec succès
 - Exemple: 200 OK
 - 3xx : requête du client redirigée, d'autres actions sont nécessaires
 - Exemple: 303 Moved Permanently
 - 4xx : erreur du client
 - Exemple: 404 Resource Not Found
 - 403 il vous est interdit d'accéder à cette page
 - 5xx : erreurs du serveur.
 - Exemple: 503 Service Unavailable, le serveur ne peut répondre à la requête, par exemple suite à une surcharge

Réponse envoyée par le serveur

23

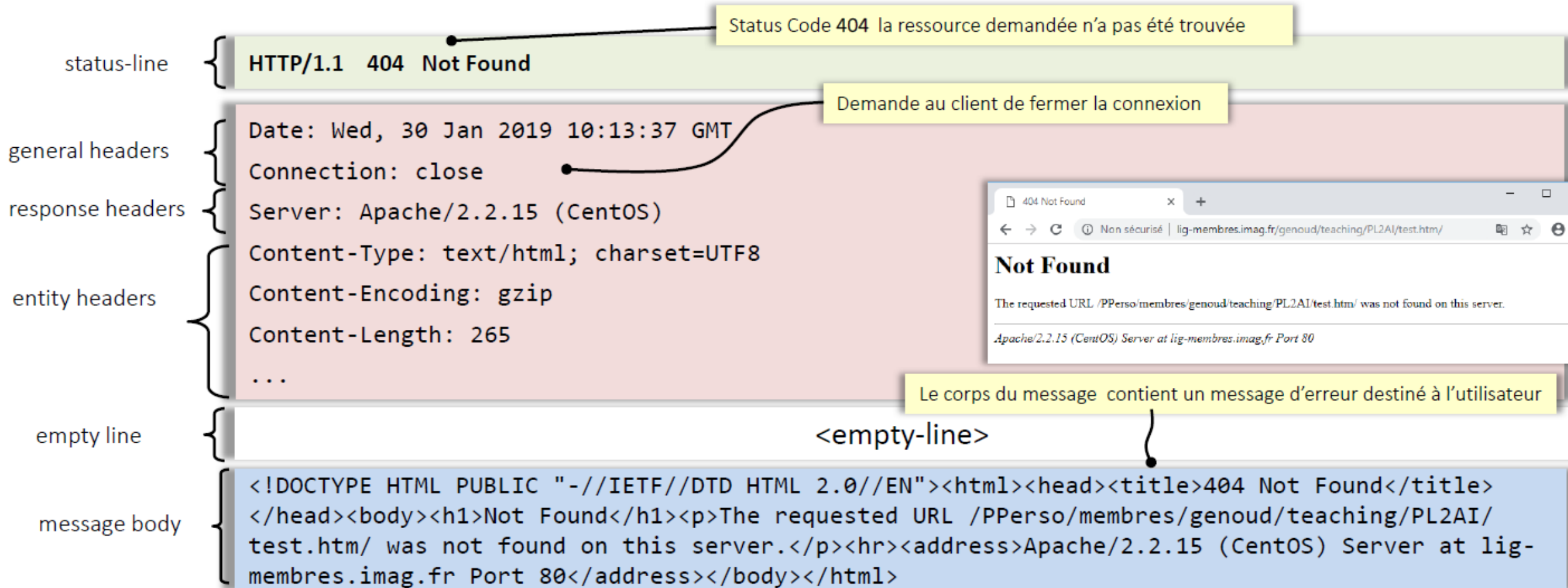
- La réponse réelle dépendra de la méthode de requête utilisée. Dans une requête GET, la réponse contiendra une entité correspondant à la ressource demandée.



Réponse envoyée par le serveur

24

➤ Réponse pour une page qui n'a pas été trouvée



Outils de développement

25

- ▶ Les outils de développement des navigateurs permettent de suivre l'activité réseau et de tracer les requêtes HTTP (Inspecteur ⇒ onglet réseau)
- ▶ Quand une page est chargée, possibilité de voir le détail de toutes les requêtes

The screenshot displays the Chrome DevTools Network tab for the URL `http://lig-membres.imag.fr/genoud/teaching/PL2AI/`. The interface includes a toolbar with icons for Inspecteur, Console, Débogueur, Éditeur de style, Performances, Mémoire, Réseau (selected), Stockage, and Accessibilité. Below the toolbar is a filter bar for URLs. The main area shows a table of network requests with columns for Status, Method, File, Domain, Source, Type, Transfer, Size, and Time. A timeline at the bottom of the table shows the duration of each request. The right sidebar provides detailed information for the selected request, including the request URL, method, address, status code (200 OK), version (HTTP/1.1), and headers (response and request).

État	Méthode	Fic...	Domaine	Source	Type	Transfert	Taille	0 ms	160 ms	320 ms	480 ms	640 ms
200	GET	/gen...	lig-membres.i...	document	html	3,71 Ko	13,54 Ko	16 ms				
200	GET	boot...	lig-membres.i...	stylesheet	css	20,85 Ko	137,63 Ko	25 ms				
200	GET	style...	lig-membres.i...	stylesheet	css	1,04 Ko	2,34 Ko	0 ms				
200	GET	ban...	lig-membres.i...	img	png	12,12 Ko	11,87 Ko	37 ms				
200	GET	jque...	lig-membres.i...	script	js	29,74 Ko	84,63 Ko	17 ms				
200	GET	boot...	lig-membres.i...	script	js	20,51 Ko	69,30 Ko	17 ms				
200	GET	js?id...	www.go...	script	js	33,00 Ko	91,30 Ko	424 ms				
200	GET	favic...	lig-membres.i...	img	vnd.micros...	259 o	0 o	16 ms				
200	GET	anal...	www.go...	script	js	17,63 Ko	43,10 Ko	88 ms				
200	GET	colle...	www.go...	img	gif	441 o	35 o	25 ms				

10 requêtes | 453,75 Ko / 139,28 Ko transférés | Terminé en : 560 ms | DOMContentLoaded: 131 ms | load: 503 ms

En-têtes de la réponse (519 o)

- Accept-Ranges: bytes
- Connection: close
- Content-Encoding: gzip
- Content-Length: 3481
- Content-Type: text/html; charset=UTF-8
- Date: Wed, 30 Jan 2019 11:11:22 GMT
- ETag: "2bda64-362d-57b281477423c"
- Last-Modified: Wed, 21 Nov 2018 07:55:55 GMT
- Server: Apache/2.2.15 (CentOS)
- Vary: Accept-Encoding

En-têtes de la requête (372 o)

- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
- Accept-Encoding: gzip, deflate
- Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
- Connection: keep-alive
- Host: lig-membres.imag.fr
- Upgrade-Insecure-Requests: 1
- User-Agent: Mozilla/5.0 (Windows NT 10.0; ...) Gecko/20100101 Firefox/54.0

Langages du web

26

- Dans une page web, on distingue bien
 - **l'aspect information** , relatif au contenu (les textes, images, sons , vidéos, liens) et à la structuration (l'organisation) de cette information
 - **l'aspect présentation** de l'information, relatif à la forme, car plusieurs «formes» sont possibles (sur grand écran, smartphone)
 - **l'aspect comportement** qui permet d'ajouter un comportement (animation, interaction, ...) aux pages web
- HTML le langage de base pour décrire des pages web, MAIS il y a plusieurs langages qui peuvent être présents dans le développement des pages web
 - HTML : Pour décrire et structurer les informations de la page
 - CSS : pour présenter les informations de la page (couleurs, encadrés, taille du texte, positionnement des divisions, des images,...)
 - Javascript : Langage de programmation qui permet d'ajouter un comportement (animation, gestion de l'interaction...) aux pages web
 - ...

Pages statiques et dynamiques

27

➤ Pages statiques

- Documents HTML sous forme de fichiers
- Transmis tels quels par le serveur HTTP
- La page visualisée N fois sur le même navigateur donnera toujours le même résultat

➤ Pages dynamiques

- La page visualisée dépend des manipulations de l'utilisateur
- Elle est générée dynamiquement (à la volée) par le serveur web selon les actions de l'utilisateur dans la page
- Nécessite de la programmation côté serveur pour prendre en compte les actions
 - PHP, **node js**, ...

Programmation Web côté client

28

- Le développement de scripts qui sont exécutés sur le navigateur du poste client
- Des scripts embarqués dans la page HTML qui sont transférés depuis le serveur vers le client
 - exemples : javascript, vbscript, ...
- Ces scripts décrivent le comportement du site qui peut être :
 - Modification d'apparence de la page, des animations, validation des formulaires, ...
- **Développement front-end** : consiste à créer des applications, des logiciels ou des sites web dont le rendu s'effectue côté client. Dans le domaine du développement web, les développeurs utilisent principalement **HTML, le CSS et le JavaScript.**

Programmation Web côté serveur

29

- Le développement de scripts qui sont exécutés sur le serveur
- Le résultat d'exécution de ces scripts est
 - soit une page HTML (page dynamique) qui est renvoyée au navigateur client
 - soit des données de plusieurs formats tels que json, ...
- Langages de scripts coté serveur : PHP, ASP, Perl, **javascript (dans l'environnement d'exécution Node js)**, ASP (Active Server Pages), ...
- Généralement, les scripts exécutés coté serveur font de traitement sur des bases de données en réponse aux requêtes des navigateurs clients.
- **Le back end** : désigne les parties du code d'une application ou d'un logiciel permettant son fonctionnement et inaccessibles à l'utilisateur.
 - Le développement back-end concerne les applications dont le rendu s'effectue coté serveur.
 - Elles accèdent généralement aux bases de données, serveurs de fichier, ...

Les CMS

30

- CMS (Content Management System) ou SGC (Système de Gestion de Contenu)
 - programme qui facilite la création et la gestion d'un site web
 - interface WYSIWYG (What You See Is What You Get)
 - WordPresse, Drupal, Joomla, ...

Les frameworks

31

- Framework (de développement informatique)
 - un ensemble d'outils et de composants logiciels éprouvés et réutilisables (bibliothèques de programmes, classes, ...) organisés conformément à un plan d'architecture bien défini (« squelette » ou « canevas »).
 - Il est souvent fourni sous la forme d'une bibliothèque logicielle et accompagné d'une documentation
- Intérêts :
 - Bonne organisation du code source grâce à l'architecture proposée (imposée ?) par le framework
 - Rapidité de développement
 - au lieu de faire un développement complet à partir zéro on s'appuie sur les composants du framework et on peut plus facilement se concentrer que sur la partie métier
- Exemples :
 - Bootstrap, React JS, Angular, Vue JS, ...

Chapitre 2 :

Le langage HTML

Rôles de HTML et CSS

33

➤ **HTML** (*HyperText Markup Language*) :

- Il a fait son apparition dès 1991 lors du lancement du Web.
- Son rôle est de gérer et organiser le contenu.
- C'est donc en HTML que vous écrirez ce qui doit être affiché sur une page web : du texte, des liens, des images etc..

➤ **CSS** (*Cascading Style Sheets*, aussi appelées *Feuilles de style*) :

- Son rôle est de gérer l'apparence de la page web (agencement, positionnement, décoration, couleurs, taille du texte...).
- Ce langage est venu compléter le HTML en 1996.

➤ Dans une page web, il faut bien séparer

- **l'aspect information** , relatif au contenu (les textes, images, sons , vidéos, liens) et à la structuration (l'organisation) de cette information ⇒ **HTML**
- **l'aspect présentation** de l'information, relatif à la forme, car plusieurs «formes» sont possibles (sur grand écran, smartphone) ⇒ **CSS**

Le langage HTML

34

- Markup langage : langage de balisage
- HTML N'EST PAS un langage de programmation
 - Langage descriptif : qui décrit le contenu de la page
- Langage pour créer des pages web
- Brique de base du Web
- La version actuelle est HTML5.2
- On obtient un fichier qui fournit une description de la page et c'est **au navigateur d'interpréter le fichier et produire la page web**

HTML : un langage à balises (tags)

35

- Les balises HTML servent principalement à
 - structurer le texte : divisions, titres, paragraphes, tableaux, liste, section, article, barre de navigation, en-tête, ...
 - inclure des objets externes : images, sons, vidéos
 - réaliser des liens hypertextes pour naviguer entre différentes pages

Syntaxe des balises

36

- `<nombalise > contenu </nombalise>`
- Balises
 - noms entourés de `<...>`
 - qualifient des portions de texte
- Fonctionnent (en général) par paires
 - `<nombalise>` : balise ouvrante
 - `</nombalise>` balise fermante
- Certaines balises sont "auto fermantes"
 - `<nombalise />` (/ facultatif en HTML 5)
- Les balises ouvrantes et balises auto fermantes peuvent être personnalisées à l'aide d'attributs
 - `<nombalise attr 1 ="valeur 1 " att 2 ="valeur 2 ...>`

Exemple des balises

37

- `<h1>Ceci est un titre</h1>`
- `<p>Ceci est un paragraphe</p>`
- `<hr />` (balise auto-fermante) (`<hr>` valide en HTML 5)
- ``

Structure d'une page HTML

38

```
<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Document</title>

</head>

<body>

    contenu de la page

</body>

</html>
```

- **DOCTYPE** indique que le navigateur doit interpréter le code de la page comme HTML (la dernière version du langage).
- **Head** Entête du document
 - méta informations utilisées par le navigateur et/ou les moteurs de recherche
- **<body>...</body>**: Corps du document, le contenu qui sera affiché dans la fenêtre du navigateur
- Une page HTML est enregistrée dans un fichier se terminant par l'extension « **.html** ».

La balise head

39

- La balise `<title>`
 - Contient le titre affiché dans la barre de titre du navigateur Web.
- Les balises `<script>`, `<link>` et `<style>` pour inclure ou définir des styles et des scripts
- La balise `<meta>`
 - Contient l'information sur votre page pour un navigateur, un moteur de recherche, etc.
 - La balise `<meta />` spécifie (en général) à la fois deux attributs : `name` et `content`.
 - Ce sont des suggestions pour les moteurs de recherche sur comment indexer la page web.
- `<meta name = "keywords"`
 - `content = "harry potter, harry potter and the deathly`
 - `hallows, deathly hallows, ministry of magic, resurrection`
 - `stone, clock of invisibility" />`

Éléments HTML

40

- Les balises et le texte
 - Le texte du document est ce qui est « entre » ou autour des balises
 - Les balises servent à déterminer la « fonction » du texte dans une structure
- Exemple
 - `ce texte est un lien`
- Les balises peuvent simplement altérer la « forme »
 - Exemple
 - `en gras`, `<i>en italiques</i>`, `<s>texte barré</s>`
- Les primitives premières du HTML ont une sémantique « documentaire »
 - `<p>`, `<h1>`, `<cite>`, `<address>`, `<blockquote>`, ``, ``, ``, ...

Éléments HTML

41

- Un deuxième jeu important du HTML concerne l'établissement des liens hypertextes :
 - `<a>`, ...
- Un troisième jeu important du HTML concerne la gestion de l'organisation visuelle (l'agencement) :
 - `<table>`, `<thead>`, `<tbody>`, `<tr>`, `<td>`, `<colgroup>`, `<div>`, ``, ...
- Un quatrième jeu du HTML concerne les nouvelles balises sémantiques :
 - `<section>`, `<article>`, `<header>`, `<footer>`, ...
- Un cinquième jeu du HTML concerne la définition des formulaires :
 - `<form>`, `<input>`, `<select>`, `<textarea>`, `<label>`, ...
- Un sixième jeu du HTML concerne la gestion de multimédias :
 - `<video>`, `<audio>`, ...

Les liens hypertextes

42

- balise <a>, l'attribut href précise l'URL de destination
 - `Plan`
- 3 types de liens
 - Vers document externe
 - Ex: lien vers une page HTML d'un autre site
 - Vers un document interne
 - Ex: lien vers une page HTML du même site que celui de la page HTML où se trouve le lien
 - A l'intérieur d'un même document (ancres)
 - Ex: lien vers une portion donnée d'une page HTML
- Le lien peut servir à envoyer un e-mail
 - `Envoyez-moi un mail`

Les liens hypertextes

43

- Liens vers une page externe
 - ` université ubs `
- Liens vers une page interne
 - On utilise le chemin relatif de la page
 - ` Ceci est un test `
- Liens à l'intérieur d'un document (ancres)
 - lien vers un élément précis d'une page (dans un endroit précis),
 - élément identifié par un attribut **id**
 - `<section id="profil" > ...</section>`
 - Utilisation d'un fragment (#) dans l'URL
 - `Mon profil `
 - ` Mon profil`
 - ` Mon profil`

Les images

44

- La balise ``, image à afficher est spécifiée avec l'attribut `src`
- Attributs
 - `src` : URL de l'image (obligatoire)
 - un fichier situé sur la même machine contenant la page courante
 - ``
 - un fichier situé sur une machine distante
 - ``
 - `alt` : texte alternatif, à afficher si l'image ne peut pas l'être (recommandé)
 - `title` : bulle d'aide
 - `width, height` : taille de l'image en pixel

Exemple : ``

Les images

45

- Si l'image est placée à l'intérieur d'une balise `<a>`, elle devient un lien hypertexte.
- ``
 ``
``

Les balises <h1>, <h2>, ... <h6>

46

➤ Les balises de titres <h1>, <h2>, ... <h6>

...

```
<body>
```

```
  <h1> Harry Potter </h1>
```

```
  <h2> Books </h2>
```

```
  <h3> Harry Potter and the Philosopher's Stone </h3>
```

```
</body>
```

...

Harry Potter

Books

Harry Potter and the Philosopher's Stone

output

La balise <p>

47

- La balise <p> permet d'insérer du texte dans un paragraphe.
- Chaque paragraphe pourra contenir une idée, un concept.
- Un paragraphe commence sur une nouvelle ligne

<body>

<p>

Harry Potter and the Deathly Hallows, the last book in the series, begins directly after the events of the sixth book. Voldemort has completed his ascension to power and gains control of the Ministry of Magic.

</p>

</body>

Harry Potter and the Deathly Hallows, the last book in the series, begins directly after the events of the sixth book. Voldemort has completed his ascension to power and gains control of the Ministry of Magic

output

La balise <!-- ... -->

48

- balise <!-- ... --> insère du commentaire dans une page html
- Les commentaires ne peuvent pas être imbriqués et ne peuvent pas contenir --.

```
<body>
```

```
<!-- My web page, by Bob Student CSE 380, Fall 2048 -->
```

```
<p>CS courses are <!-- NOT --> a lot of fun!</p>
```

```
</body>
```

CS courses are a lot of fun!

output

Les balises <hr> et

49

- Les deux balises sont auto-fermantes
-
 fait un retour à la ligne
- <hr> dessine une ligne horizontale

```
<body>
```

```
  <p> First paragraph </p>
```

```
  <hr>
```

```
  <p> Second Paragraph </p>
```

```
</body>
```

First Paragraph

Second Paragraph

output

Les balises ,

50

- La balise : texte souligné (habituellement en italique).
- La balise : texte fortement souligné (habituellement en gras).
- Les balises doivent être correctement imbriquées pour une page valide.

```
<body>  
  <p>  
    HTML is <em>really</em>, <strong>REALLY</strong> fun!  
  </p>  
</body>
```

HTML is *really* **REALLY** fun!

output

Liste non ordonnée : balises ,

51

- La balise représente une liste à puces d'éléments (bloc).
- La balise représente un élément unique dans la liste (bloc).
- Ces balises peuvent contenir l'attribut **type** qui pourra prendre les valeurs : *square, circle, disc, none*

```
<body>
```

```
    <ul>
```

```
        <li>No shoes</li>
```

```
        <li>No shirt</li>
```

```
        <li>No problem!</li>
```

```
</body>
```

- No shoes
- No shirt
- No problem!

output

Liste non ordonnée : balises ,

52

```
<body>
  <ul>
    <li>Harry Potter characters:
      <ul>
        <li>Harry Potter</li>
        <li>Hermione</li>
        <li>Ron</li>
      </ul>
    </li>
    <li>LOTR characters:
      <ul>
        <li>Frodo</li>
        <li>Bilbo</li>
        <li>Sam</li>
      </ul>
    </li>
  </ul>
</body>
```

- Harry Potter characters:
 - Harry Potter
 - Hermione
 - Ron
- LOTR characters:
 - Frodo
 - Bilbo
 - Sam

output

Liste ordonnée : balise

53

- La balise représente une liste numérotée des éléments.
- Elle peut contenir l'attribut **type** qui pourra prendre les valeurs : 1 | A | a | I | i.

```
<body>  
  <p>Apple business model:</p>  
  <ol>  
    <li>Beat Microsoft</li>  
    <li>Beat Google</li>  
    <li>Conquer the world!</li>  
  </ol>  
</body>
```

Apple business model:

1. Beat Microsoft
2. Beat Google
3. Conquer the world

output

Liste de définitions : balises <dl>, <dt>, <dd>

54

- La balise <dl> représente une liste de descriptions de termes.
- La balise <dt> représente chaque terme, et la balise <dd> sa définition.

```
<body>
```

```
    <dl>
```

```
        <dt>newbie</dt> <dd>one who does not have mad skills</dd>
```

```
        <dt>jaded</dt>  <dd>tired, bored, or lacking enthusiasm </dd>
```

```
        <dt>frag</dt>   <dd>a kill in a shooting game</dd>
```

```
    </dl>
```

```
</body>
```

```
newbie
```

```
    one who does not have mad skills
```

```
jaded
```

```
    Tired, bored, or lacking enthusiasm
```

```
frag
```

```
    a kill in a shooting game
```

output

Texte préformaté : balise <pre>

55

- La balise <pre> permet d'afficher le texte avec tous les espaces et les passages à la ligne.

```
<pre>
```

```
Bill Gates speaks
```

```
    You will be assimilated
```

```
Microsoft fans delirious
```

```
</pre>
```

```
Bill Gates speaks
```

```
    You will be assimilated
```

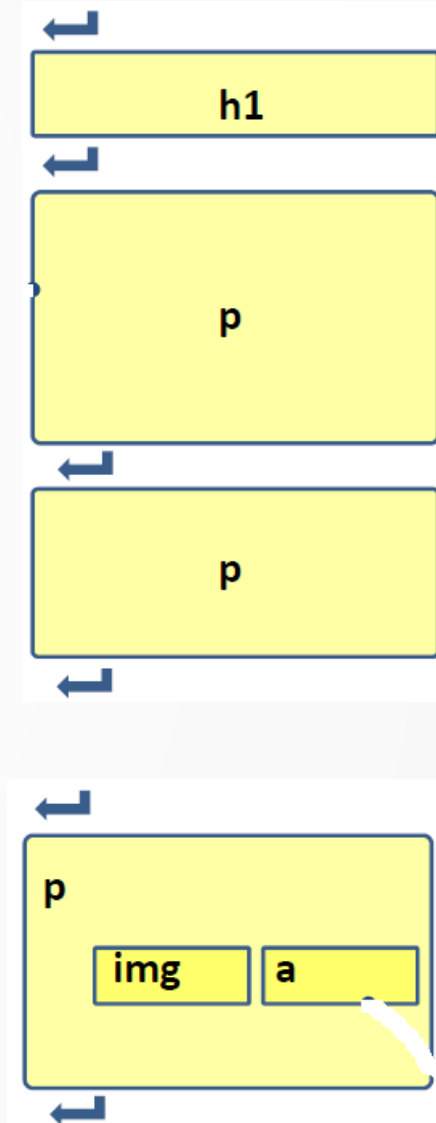
```
Microsoft fans delirious
```

output

Éléments de bloc et éléments en ligne

56

- Deux types d'éléments
 - bloc (block) et en ligne (inline)
- Élément de bloc : par défaut affichage vertical
 - commence toujours sur une nouvelle ligne
 - prend la largeur totale disponible.
 - un élément bloc prend la largeur de son conteneur
 - Conteneur : un élément HTML qui peut englober d'autres éléments ou du texte
- Élément en ligne : par défaut affichage au fil du texte
 - ne démarre pas sur une nouvelle ligne et ne prend que toute la largeur nécessaire.
 - On ne peut pas utiliser les attributs width et height



Éléments de bloc et éléments en ligne

57

Éléments de bloc : Exemples

- `<p>` paragraphe
- `<h1>` `<h2>` ... `<h6>` titres
- `` liste ordonnée
- `` liste non ordonnée
- `` élément d'une liste
- `<table>` tableau
- `<tr>` ligne d'un tableau
- `<div>` conteneur générique servant à regrouper d'autres éléments de type bloc ou en ligne
- ...

Éléments de bloc et éléments en ligne

58

Éléments en ligne : Exemples

- `<a>` lien hypertexte
- `` inclusion d'une image
- ``
- ``
- `` conteneur générique servant à regrouper d'autres éléments de type en ligne (équivalent à `<div>`)
- ...

Éléments de bloc et éléments en ligne

59

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    * {
      border: 2px solid black;
    }
    p, ul, li {
      background-color: bisque;
    }
    strong, em, span {
      background-color: aqua;
    }
  </style>
</head>
```

```
<body>
  <p> c'est un paragraphe  c'est un
paragraphe  c'est un paragraphe  c'est un
paragraphe  <em>emphasis</em>'est un paragraphe
c'est un paragraphe  c'est un
paragraphe  <strong>inline </strong>c'est un
paragraphe  c'est un paragraphe  c'est un
paragraphe
</p>
  <ul>
    <li>item 1</li>
    <li>item2 <span>texte inline </span></li>
  </ul>
</body>
</html>
```

c'est un paragraphe c'est un paragraphe c'est un paragraphe c'est un
paragraphe **emphasis** est un paragraphe c'est un paragraphe c'est un
paragraphe **inline** c'est un paragraphe c'est un paragraphe c'est un
paragraphe

- item 1
- item2 **texte inline**

Tableaux : balises `<table>`, `<tr>`, `<td>`

60

- La balise **`<table>`** définit le tableau; la balise **`<tr>`** chaque ligne du tableau, et la balise **`<td>`** les données de chaque cellule du tableau.
- Utile pour l'affichage de grands ensembles de données organisées en lignes et colonnes.
- Les cellules d'une ligne entourées par la balise **`<th>`** sont considérées comme des entêtes.
- Une légende au début de la table est réalisée à l'aide de la balise `<caption>`.
- REMARQUE : les tables sont parfois utilisées par des débutants pour la mise en page web, mais ce n'est pas sémantiquement approprié en HTML et cela doit être évité.

Tableaux : balises <table>, <tr>, <td>

61

```
<table border="1px">  
  <caption>My important data</caption>  
  <tr>  
    <th>Titre 1</th>  
    <th>Titre 2</th>  
  </tr>  
  <tr>  
    <td>cellule 1 </td>  
    <td>cellule 2 </td>  
  </tr>  
  <tr>  
    <td>cellule3</td>  
    <td>cellule4</td>  
  </tr>  
  <tr>  
    <td>foot 1</td>  
    <td>foot 2</td>  
  </tr>  
</table>
```

My important data

Titre 1	Titre 2
cellule 1	cellule 2
cellule3	cellule4
foot 1	foot 2

Tableaux : balises <thead>, <tbody> et <tfoot>

62

```
<table border="1px">  
  <caption>My important data</caption>  
  <thead>  
    <tr>  
      <th>Titre 1</th>  
      <th>Titre 2</th>  
    </tr>  
  </thead>  
  <tbody>  
    <tr>  
      <td>cellule 1 </td>  
      <td>cellule 2 </td>  
    </tr>  
    <tr>  
      <td>cellule3</td>  
      <td>cellule4</td>  
    </tr>  
  </tbody>
```

```
<tfoot>  
  <tr>  
    <td>foot 1</td>  
    <td>foot 2</td>  
  </tr>  
</tfoot>  
</table>
```

My important data

Titre 1	Titre 2
cellule 1	cellule 2
cellule3	cellule4
foot 1	foot 2

Tableaux : balises <table>, <tr>, <td>

63

- Il est possible de fusionner des lignes ou des colonnes dans un tableau.
- L'attribut **colspan** permettra de fusionner des colonnes
- L'attribut **rowspan** permettra de fusionner des lignes

```
<table border="1px">  
  <tr>  
    <td rowspan=2>Titre 1</td>  
    <td>Titre 2</td>  
    <td>Titre 3</td>  
  </tr>  
  <tr>  
    <td>cellule 1 </td>  
    <td>cellule 2 </td>  
  </tr>  
  <tr>  
    <td>cellule3</td>  
    <td colspan=2>cellule4</td>  
  </tr>  
</table>
```

Titre 1	Titre 2	Titre 3
	cellule 1	cellule 2
cellule3	cellule4	

La balise <div> (1)

64

- La balise <div> est utilisée pour indiquer une section logique ou une zone d'une page.
- C'est un conteneur qui peut contenir des éléments HTML de types différents
- La balise <div> est un élément neutre et n'a pas d'apparence par défaut, mais vous pouvez lui appliquer des styles.
- C'est un élément de bloc
- Avant HTML5, et même jusqu'à maintenant, cette balise était utilisée pour la présentation d'une page. Et ceci en découpant la page en divisions (div) et de leur appliquer les styles CSS nécessaires pour produire la forme souhaitée de la page

La balise <div> (2)

65

<div id='header'>

<div id='menu_navigation'>

<div class='article'>

<div class='section'>

<div id='right_side'>

<div id='footer'>

La balise

66

- La balise permet de définir un élément "in-line" de texte (ou conteneur) auquel s'appliquera des styles CSS
- La balise est un élément neutre et n'a pas d'apparence par défaut, mais vous pouvez lui appliquer des styles.
- Les éléments se placent les uns à côté des autres et ont la dimension qu'occupe leur contenu

```
<h2>Coding Horror! Coding Horror!</h2>
<p>See our <span class="special">spectacular</span> deal
on Droids!</p>
<p>We'll beat <span class="shout"> any advertised
price</span>!</p>
```

HTML

Coding Horror! Coding Horror!

See our **spectacular** deal on Droids!

We'll beat **any advertised price!**

output

Les nouvelles balises sémantiques en HTML5

67

- `<header>` : Qui indique que l'élément est une en-tête
- `<footer>` : Qui indique que l'élément est un pied-de-page
- `<nav>` : Qui indique un élément de navigation tel qu'un menu
- `<aside>` : Qui correspond à une zone secondaire non liée au contenu principal de la page
- `<article>` : Qui représente une portion de la page qui garde un sens même séparé de l'ensemble de la page (comme un article de blog par exemple)
- `<section>` qui permet de regrouper des éléments partageant une même thématique. Répétées plusieurs fois, elles représentent des sections ou des chapitres dans une page

Les nouvelles balises sémantiques en HTML5

68

- Ces balises sont des éléments de bloc qui n'ont pas une présentation prédéfinie
- Sont des améliorations sémantiques qui remplacent la balise <DIV>
- Elles sont utilisées comme conteneurs d'autres éléments HTML
- Leur utilisation peut simplifier la présentation de la page en leur associant des styles CSS

Les éléments `<article>` et `<section>`

69

- L'élément `<article>`
 - est un fragment indépendant du contenu général.
 - Convient pour du contenu comme des nouvelles ou des articles de blog, des messages de forum
- L'élément `<section>`
 - n'est pas une partie indépendante
 - C'est soit une manière de subdiviser une page en plusieurs sujets soit une manière de sectionner un article en sections
 - est assez trompeur car il est largement trop employé par les développeurs Web comme une alternative à `<div>`.

Les éléments <header> et <footer>

70

■ L'élément <header>

- représente un groupe d'aide de navigation ou d'introduction de la page Web ou d'un article
- <header> peut être un entête de la page ou d'un article ou d'une section

■ La balise <footer>

- Elle est utilisée pour créer la structure de pied de page de votre document web.
- Vous pouvez également utiliser cet élément plusieurs fois sur une seule page pour des blocs différents.
- Cette balise peut être utilisée, par exemple, pour marquer le droit d'auteur, conditions d'utilisation ...

La balise <nav>

71

- L'élément <nav> est utilisé pour créer des menus avec des liens vous permettant de naviguer sur les pages du site Web.
- Elle peut également être utilisée plusieurs fois.
- Bien souvent cette balise est associée à une liste, ordonnée ou non, de liens : et .

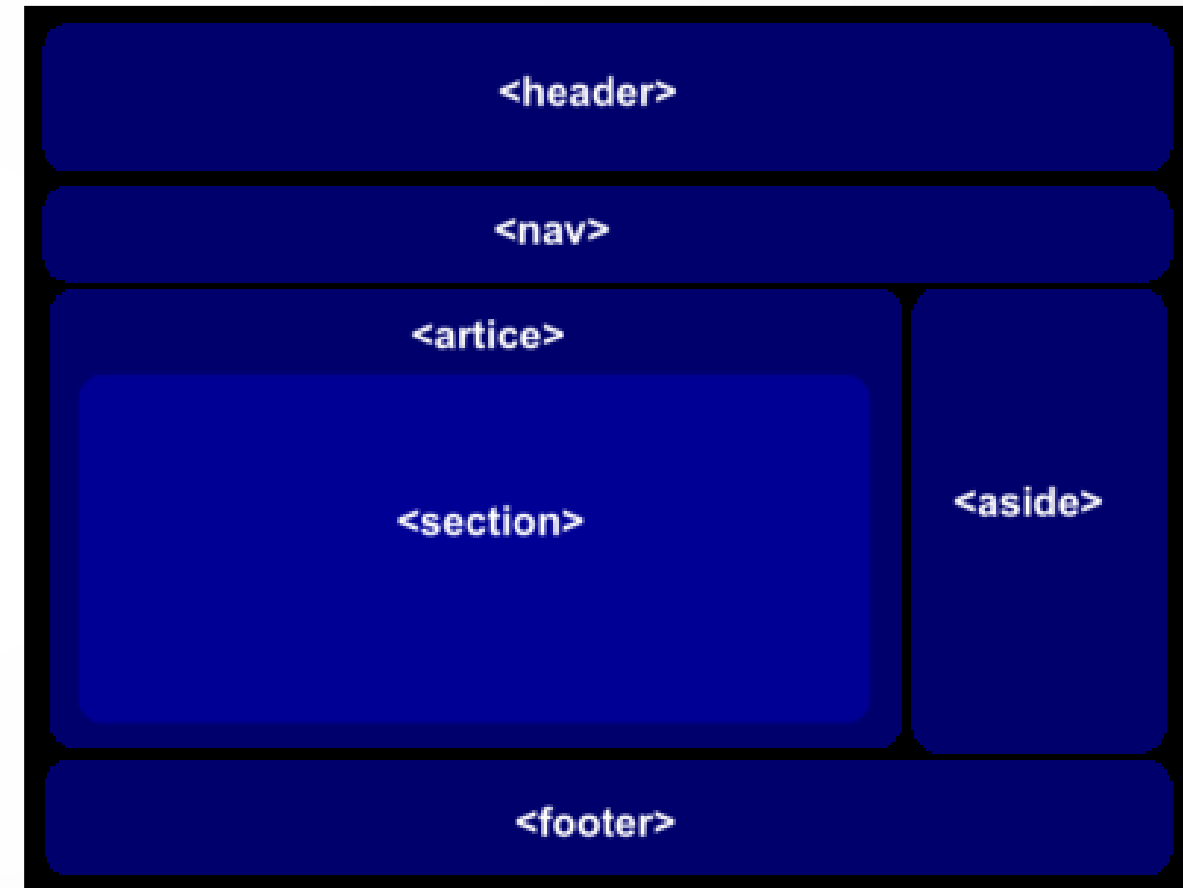
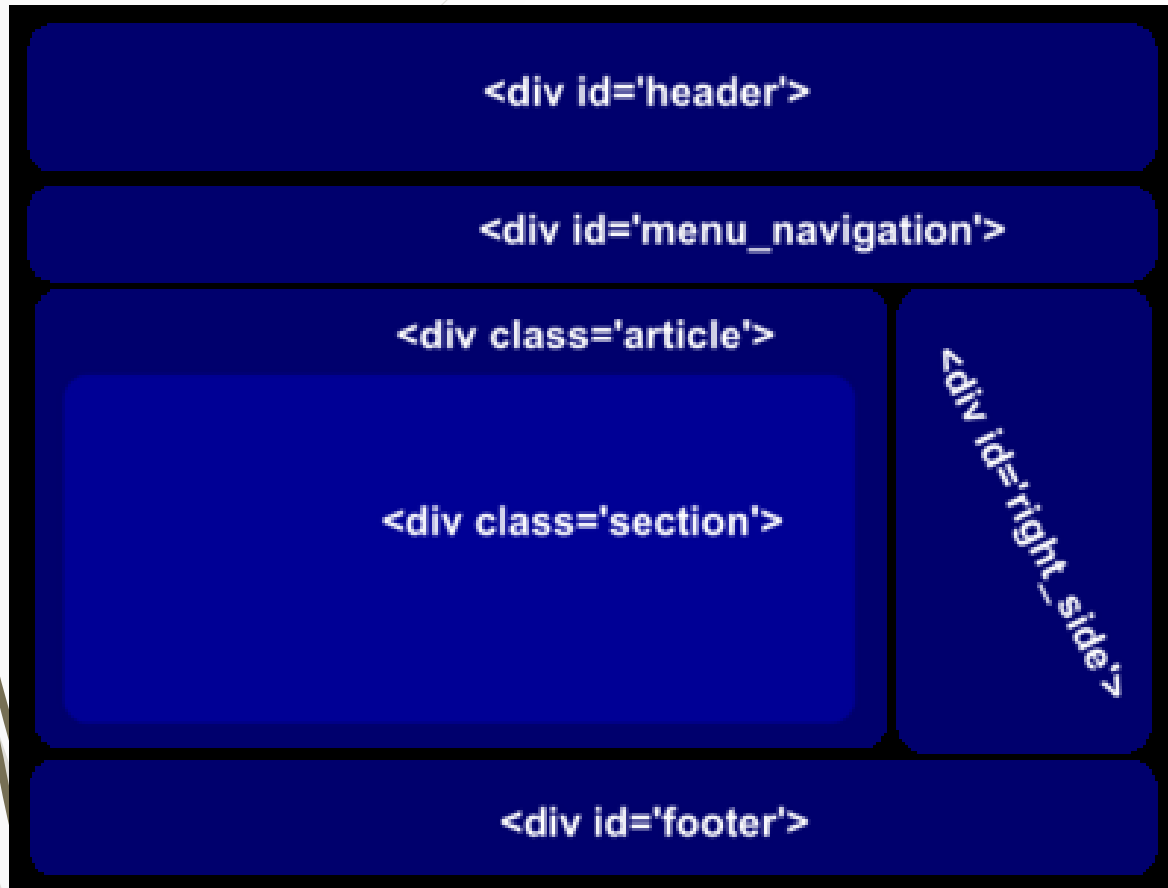
La balise <aside>

72

- Est utilisée pour un contenu secondaire qui n'est pas emboîtée dans aucun élément du document (<article> ...).
- L'exemple le plus approprié pour utiliser l'élément <aside> est la fameuse Sidebar ou colonne latérale.
- Nous pouvons utiliser cet élément pour contenir les bannières publicitaires ou tout simplement pour le contenu qui doit être placé séparément du contenu principal.

Présentation des pages

73



- Les nouvelles balises sémantiques remplacent `<div>` dans la structuration des pages web. Néanmoins, `div` reste toujours utilisé dans certains cas

Les éléments `<figure>` et `<figcaption>`

74

- Les éléments `<figure>` et `<figcaption>` sont utilisés pour présenter un bloc de contenu avec une légende.
- La balise `<figure>` définit un contenu autonome, comme des illustrations, des diagrammes, des photos, des listes de codes, etc
- La balise `<figcaption>` définit une légende pour un élément `<figure>`.

Les éléments <figure> et <figcaption>

75

```
<html>
<head>
<style>
figure {
  border: 1px #cccccc solid;
  padding: 4px;
  margin: auto;
  width : 50%;
}
h1 {text-align: center;}
figcaption {
  background-color: black;
  color: white;
  font-style: italic;
  padding: 2px;
  text-align: center;
}
</style>
</head>
<body>
<h1>The figure and figcaption elements + CSS</h1>

<figure>
  
  <figcaption>Fig.1 - Trulli, Puglia, Italy</figcaption>
</figure>
</body>
</html>
```

The figure and figcaption elements + CSS



Fig.1 - Trulli, Puglia, Italy

Les formulaires (1)

76

- Apporte de l'interactivité avec l'utilisateur en proposant des zones de dialogue : un formulaire n'est qu'une interface de saisie
- Selon les choix de l'utilisateur, il faut y associer un traitement
 - Sur le client avec JavaScript par exemple
 - Sur le serveur par l'intermédiaire de PHP, node js, ...
- Exemples typiques d'utilisation de formulaire
 - Moteurs de recherche
 - Interactions avec une base de données

Les éléments du formulaire (2)

77

- `<FORM>...</FORM>` début et fin du formulaire
- **Attributs** de la balise `<Form>`
 - **METHOD** : valeurs GET ou POST qui indiquent la façon dont les données sont transmises au serveur
 - Les champs du formulaire sont transmis via l'URL (méthode GET) ou dans le contenu de la requête HTTP (méthode POST).
 - **ACTION** : URL du script coté serveur qui sera exécuté quand l'utilisateur clique sur un bouton de soumission
 - **NAME** : désigne le nom du formulaire
 - **TARGET** : cible dans laquelle la réponse du script sera affichée
 - **ENCTYPE** : spécifie l'encodage utilisé pour l'envoi des données du formulaire dans le cas de la méthode POST (les données sont transmises dans le corps de la requête)

Les éléments du formulaire (3)

78

L'attribut ENCTYPE

- **ENCTYPE = "application/x-www-form-urlencoded"** : valeur par défaut ; url-encode le contenu du formulaire de la même façon que par la méthode GET
- **ENCTYPE = "text/plain"** : le contenu du formulaire est envoyé en format texte lisible par le destinataire (action = 'mailto: ' par exemple)
- **ENCTYPE = "multipart/form-data"** : permet d'expédier un fichier attaché dans le corps de la requête (<input type="file">)

Les éléments du formulaire (4)

79

Un formulaire peut contenir quatre catégories principales d'éléments :

- `<label>` : permet d'ajouter un label ou un texte
- `<input...>` : champs de saisie de texte et divers types de boutons
 - `type="text"` - zone de texte (type par défaut)
 - `type="password"` - zone de texte caché
 - `type="checkbox"` - cases à cocher
 - `type="radio"` - minimum 2, un seul sélectionnable
 - `type="submit"` – bouton de soumission du formulaire (on peut utiliser à la place `<button type="submit">...</button>`, on peut utiliser une image comme button)
 - `type="reset"` - bouton de remise à zéro des champs
 - `type="hidden"` - bouton caché mais le couple name/value est envoyé au traitement

Les éléments du formulaire (5)

80

- `type="file"` - upload d'un fichier vers le serveur
- `Type= "email"` - zone de saisie d'un e-mail
- `Type= "number"` - zone de saisie d'un nombre
- `Type= "date"` - zone de saisie d'une date
- `Type= "tel"` - zone de saisie d'un numéro de téléphone
- `type= "range"` – ajout d'un curseur
- `type= "month"` - zone de saisie d'un mois et une année
- `type= "time"` - zone de saisie d'un temps
- `type= "color"` – permet choisir une couleur
- `type= "url"` – zone de saisie d'une url
- ...

Les éléments du formulaire (6)

81

- `<select...>` : menus déroulants, listes à ascenseurs
 - `size="1"` - pop liste, 1 seul élément sélectionnable
 - `size="n"`, $n > 1$ - liste à choix multiples
- `<textarea...>` : zone de saisie d'un texte "long"

NB : Certains types de la balise `input` ont été introduits dans `html5` tels que les types `number`, `date`, `tel`, `range`, `email`, ... C'est particulièrement utile afin d'effectuer une première validation du contenu avant l'envoi des informations au serveur.

La balise input (1)

82

- type="text" - zone de texte (type par défaut)
- type="password" - zone de texte caché
- type="submit" - bouton de soumission du formulaire
- type="reset" - bouton de remise à zéro des champs

```
<!DOCTYPE html>
<html>
<body>
<h2>Password field</h2>
<form action="/action_page.php">
  <label for="username">Username:</label><br>
  <input type="text" id="username" name="username"><br>
  <label for="pwd">Password:</label><br>
  <input type="password" id="pwd" name="pwd"><br><br>
  <input type="submit" value="Submit"> <input type="reset" value="Réinitialiser">
</form>
</body>
</html>
```

Password field

Username:

Password:

La balise input (2)

83

- type="file" - upload d'un fichier vers le serveur
- Type= "email" - zone de saisie d'un e-mail. L'@ doit être présente pour que la saisie soit valide
- Type= "number" - zone de saisie d'un nombre. Il n'accepte que des nombres qui sont compris entre min et max si ceux-ci sont précisés.

```
<!DOCTYPE html>
<html>
<body>
<form action="/action_page.php">
  <label for="quantity">Quantity (between 1 and 5):</label>
  <input type="number" id="quantity" name="quantity" min="1" max="5"><br>
  <label for="email">Enter your email:</label>
  <input type="email" id="email" name="email"><br>
  <label for="myfile">Select a file:</label>
  <input type="file" id="myfile" name="myfile"><br><br>
  <input type="submit" value="Submit">
</form>
</body>
</html>
```

Quantity (between 1 and 5):

Enter your email:

Select a file: Aucun fichier choisi

La balise input (3)

84

- type = "radio" : insertion d'un élément radio. Choix d'une et une seule option parmi n

```
<!DOCTYPE html>
<html>
<body>
<h2>Radio Buttons</h2>
<p>Choose your favorite Web language:</p>
<form action="/action_page.php">
  <input type="radio" id="html" name="fav_language" value="HTML" checked>
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaScript">
  <label for="javascript">JavaScript</label><br><br>
  <input type="submit" value="Submit">
</form>
</body>
</html>
```

Radio Buttons

Choose your favorite Web language:

- ☒ HTML
- ☐ CSS
- ☐ JavaScript

Submit

La balise input (4)

85

- type = "checkbox" : Insertion d'une case à cocher. Cases à cocher permettant un choix multiple, plusieurs cases peuvent être cochées.

```
<!DOCTYPE html>
<html>
<body>

<h2>Checkboxes</h2>
<form action="/action_page.php">
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike" checked>
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label><br><br>
  <input type="submit" value="Submit">
</form>
</body>
</html>
```

Checkboxes

- ☒ I have a bike
- ☐ I have a car
- ☐ I have a boat

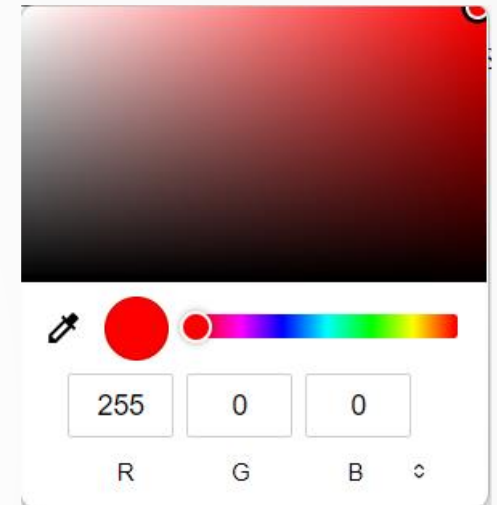
Submit

La balise input (5)

86

- Les Types "month", "date", "time", "range" et "color"

```
<!DOCTYPE html>
<html>
<body>
<form action="/action_page.php">
  <label for="favcolor">Select your favorite color:</label>
  <input type="color" id="favcolor" name="favcolor" value="#ff0000"><br>
  <label for="appt">Select a time:</label>
  <input type="time" id="appt" name="appt"><br>
  <label for="birthday">Birthday:</label>
  <input type="date" id="birthday" name="birthday"><br><br>
  <label for="vol">Volume (between 0 and 50):</label>
  <input type="range" id="vol" name="vol" min="0" max="50"><br>
  <input type="submit" value="Submit">
</form>
</body>
</html>
```



Select your favorite color:

Select a time:

Birthday:

Volume (between 0 and 50):

Contrôler la saisie

87

checked	Specifies that an input field should be pre-selected when the page loads (for type="checkbox" or type="radio")
disabled	Specifies that an input field should be disabled
max	Specifies the maximum value for an input field
maxlength	Specifies the maximum number of character for an input field
min	Specifies the minimum value for an input field
pattern	Specifies a regular expression to check the input value against
readonly	Specifies that an input field is read only (cannot be changed)
required	Specifies that an input field is required (must be filled out)
size	Specifies the width (in characters) of an input field
step	Specifies the legal number intervals for an input field
value	Specifies the default value for an input field

Select et option

88

- `<SELECT>` : définit des menus déroulants, listes à ascenseurs
- `<OPTION>` : définit un membre de la liste

```
<!DOCTYPE html>
<html>
<body>
<h2>The select Element</h2>
<p>The select element defines a drop-down list:</p>
<form action="/action_page.php">
  <label for="computer">Choose a computer:</label><br>
  <select id="computer" name="computer" size=3 multiple>
    <option value="hp" selected>hp</option>
    <option value="Dell">Dell</option>
    <option value="Acer" selected>Acer</option>
    <option value="Sony">Sony</option>
  </select><br>
  <label for="cars">Choose a car:</label>
  <select id="cars" name="cars">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
    <option value="fiat">Fiat</option>
    <option value="audi">Audi</option>
  </select><br>
  <input type="submit">
</form>
</body>
</html>
```

The select Element

The select element defines a drop-down list:

Choose a computer:

Choose a car:

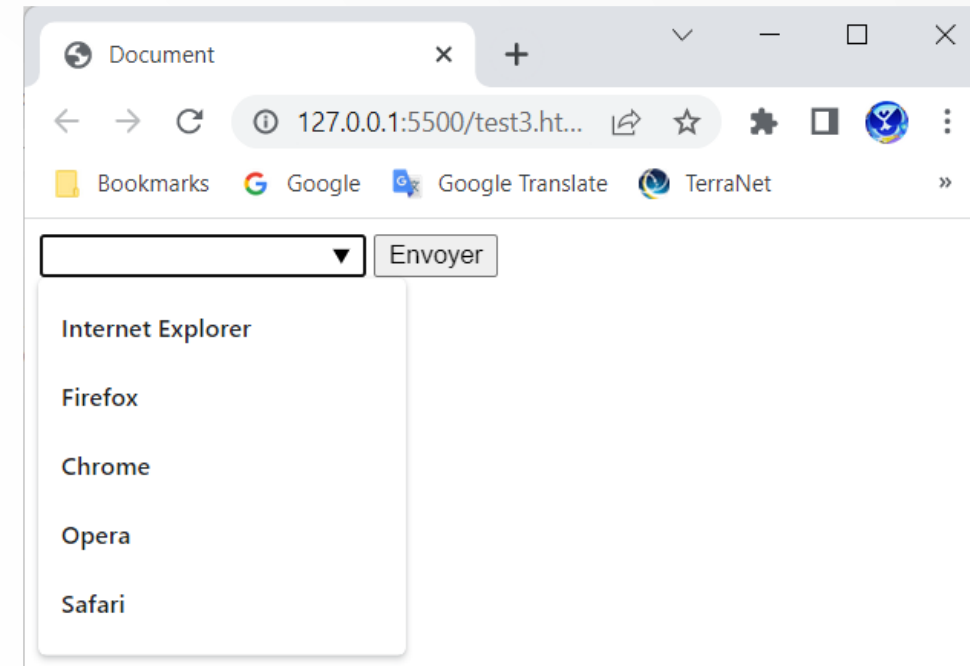
La balise datalist

89

➤ La balise <datalist> en HTML5

```
<input list="navigateur" />
<datalist id="navigateur">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

- L'élément <datalist> spécifie une liste pré-définie d'options pour un élément <input>.
- L'élément <datalist> est utilisé pour fournir une « saisie semi-automatique » sur les éléments <input>. Utiliser l'attribut **id** de la balise <datalist> et l'attribut **list** de la balise <input> pour les relier ensemble.



Textarea

90

- Zone de saisie de texte libre sur plusieurs lignes

```
<!DOCTYPE html>
<html>
<body>
<h2>Textarea</h2>
<p>The textarea element defines a multi-line input field.</p>
<form action="/action_page.php">
  <textarea name="message" rows="10" cols="30" placeholder="votre commentaire">
</textarea>
  <br><br>
  <input type="submit">
</form>
</body>
</html>
```

Textarea

The textarea element defines a multi-line input field.

votre commentaire

Envoyer

Fieldset

91

- Permet de regrouper plusieurs éléments dans une formulaire (et a une légende)

```
<!DOCTYPE html>
<html>
<body>
<form action="/action_page.php">
  <fieldset>
    <legend>Personalia:</legend>
    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="fname" value="John"><br>
    <label for="lname">Last name:</label><br>
    <input type="text" id="lname" name="lname" value="Doe"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
</body>
</html>
```

Personalia:

First name:

John

Last name:

Doe

Submit

```
<form action="validationFormulaire" method="GET">
```

Firefox - Exemple de formulaire (GET) x

localhost:8084/Exemp

MIREOT Transmettre des donn...

Exemple de formulaire (avec méthode GET)

Prénom : Amélie Nom : LE POULAIN

Sexe : féminin masculin

J'accepte les conditions d'utilisation de ce site : ☐

Valider

http://localhost:8084/ExempleFormulaire/validationFormulaire?prenom=Am%C3%A9lie&nom=LE+POULAIN&sexe=feminin&accept=ON

Request Headers 23:55:04.608

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7

Accept-Encoding: gzip, deflate

Accept-Language: en-us,en;q=0.5

Connection: keep-alive

Host: localhost:8084

Referer: http://localhost:8084/ExempleFormulaire/FormulaireAvecGet.xhtml

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:7.0.1) Gecko/20100101 Firefox/7.0.1

Les données sont envoyées dans l'URL

affiche la page

Navigateur

Serveur web

[HTTP/1.1 200 OK 14ms]

Response Headers Δ21ms

Content-Length: 443

Content-Type: text/html; charset=UTF-8

Date: Tue, 04 Oct 2011 21:55:04 GMT

Server: Apache-Coyote/1.1

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>JSP Page</title>
</head>
<body>
  <h2>
    Bonjour
    Mme Amélie LE POULAIN</h2>
  <p>
    Vous avez accepté les conditions du site.
  </p>
</body>
</html>
```

Firefox - JSP Page

localhost:8084/Exem

MIREOT Transmettre des donn...

Bonjour Mme Amélie LE POULAIN

Vous avez accepté les conditions du site.

```
<form action="validationFormulaire" method="POST">
```

POST <http://localhost:8084/ExempleFormulaire/validationFormulaire>

Les variables définies dans le formulaire ont été associées avec les valeurs entrées par l'utilisateur et passées au serveur dans le format URL-Encoded

Navigateur

affiche la page

Request Headers 23:38:10,536

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
 Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
 Accept-Encoding: gzip, deflate
 Accept-Language: en-us,en;q=0.5
 Connection: keep-alive
 Host: localhost:8084
 Referer: http://localhost:8084/ExempleFormulaire/FormulaireAvecPost.xhtml
 User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:7.0.1) Gecko/20100101 Firefox/7.0.1

Content-Type: application/x-www-form-urlencoded
 Content-Length: 56
 prenom=Am%C3%A9lie&nom=LE+POULAIN&sexe=feminin&accept=ON

données envoyées avec le corps de la requête

le serveur traite la requête POST et ses en-têtes puis passe le corps de la requête au programme spécifié par l'URL qui le traite.

Serveur web

[HTTP/1.1 200 OK 14ms]

Response Headers Δ20ms

Content-Length: 443
 Content-Type: text/html; charset=UTF-8
 Date: Tue, 04 Oct 2011 21:38:10 GMT
 Server: Apache-Coyote/1.1

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<h2>
  Bonjour
  Mme Amélie LE POULAIN</h2>
<p>
  Vous avez accepté les conditions du site.
</p>
</body>
</html>
```



Les balises multimédia

94

- **<video>**: Cette balise intègre directement un lecteur vidéo dans la page, avec des boutons Lecture, Pause, une barre de progression, du volume...
- **<audio>**: Cette balise est l'équivalent de la balise video mais pour l'audio. En 3 lignes de code vous avez un lecteur MP3 !

Exemple d'utilisation de la balise <video>

95

```
<video width="320" height="240" autoplay muted>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogg" type="video/ogg">  
Your browser does not support the video tag.  
</video>
```



- La première ligne sert à insérer la balise **<video>** et donc indiquer au navigateur qu'il devra afficher un contenu vidéo.
- Source indique l'emplacement des fichiers vidéo.
- Dans ce cas, le premier format supporté sera joué. Sinon le message sera affiché.

Exemple d'utilisation de la balise <video>

96

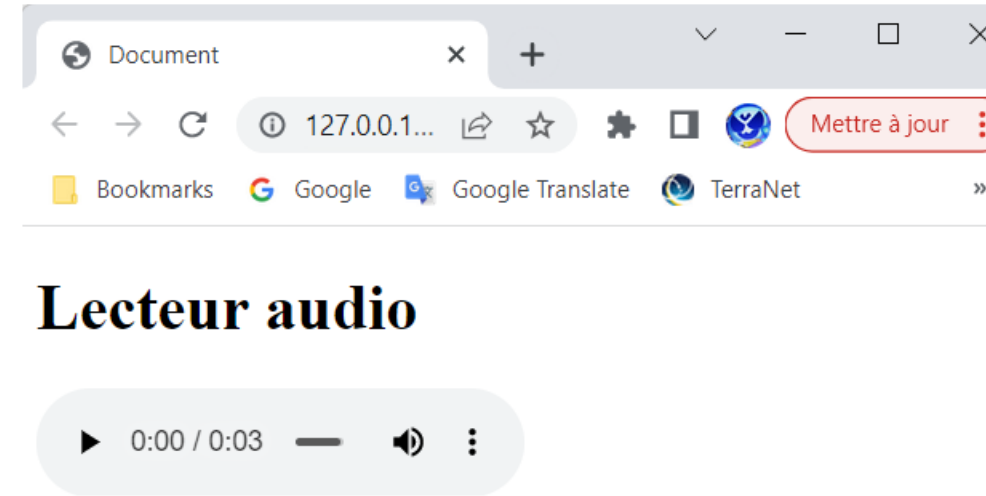
Attribute	Value	Description
<u>autoplay</u>	autoplay	Specifies that the video will start playing as soon as it is ready
<u>controls</u>	controls	Specifies that video controls should be displayed (such as a play/pause button etc).
<u>height</u>	<i>pixels</i>	Sets the height of the video player
<u>loop</u>	loop	Specifies that the video will start over again, every time it is finished
<u>muted</u>	muted	Specifies that the audio output of the video should be muted
<u>poster</u>	<i>URL</i>	Specifies an image to be shown while the video is downloading, or until the user hits the play button
<u>preload</u>	auto metadata none	Specifies if and how the author thinks the video should be loaded when the page loads
<u>src</u>	<i>URL</i>	Specifies the URL of the video file
<u>width</u>	<i>pixels</i>	Sets the width of the video player

- ▶ autoplay, controls, loop, muted, preload et src sont des attributs communs avec la balise <audio>

Exemple d'utilisation de la balise <audio>

97

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>
```



- La première ligne sert à insérer la balise **<audio>** et donc indiquer au navigateur qu'il devra afficher un contenu audio.
- La deuxième ligne indique l'emplacement du fichier audio.
- On utilise Source si l'on peut mettre le clip sous plusieurs formats. Dans ce cas, le premier format supporté sera joué.