

Université Sorbonne Paris Nord
IUT de Villetaneuse

BUT R&T – S1 – Cours R109



Introduction aux technologies Web

Laure Petrucci

Laure.Petrucci@univ-paris13.fr

<http://www.lipn.fr/~petrucci/R109/>



Objectifs

- Découvrir la structure d'un site Web : **URL**, fonctionnement du **modèle client-serveur**
- Comprendre et écrire **une page Web** dans un langage à balises normalisé de description de contenus
- Savoir modifier les **attributs de présentation** d'une page Web
- S'initier à l'**utilisation avancée d'un navigateur** Web
- Comprendre les notions de base de **responsive design** pour adapter un contenu Web à un **navigateur mobile**

Organisation du module

Volume horaire

Cours	4×1h
Travaux pratiques	4×1h30

Évaluation

Tous les travaux pratiques sont notés

Organisation pratique

- Support de cours : diapositives
- Sujets de TP et diapositives accessibles **en ligne**

Adresse des ressources

`http://lipn.fr/~petrucci/R109/`

Plan: Architecture client-serveur

- 1 Architecture client-serveur
 - Généralités
 - Client-serveur et Web
- 2 HTML
- 3 CSS
- 4 Sites Web adaptatifs

L'analogie du café



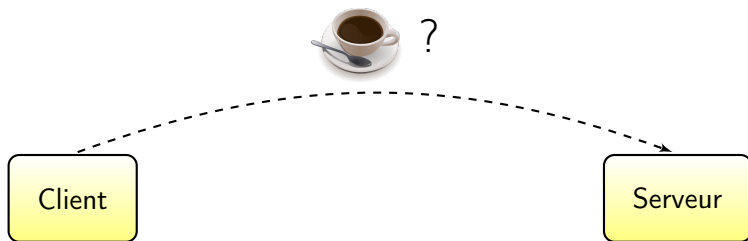
Client

The diagram consists of two yellow rounded rectangular boxes with black outlines. The box on the left contains the word 'Client' and the box on the right contains the word 'Serveur'. There are no lines or arrows connecting the two boxes.

Serveur

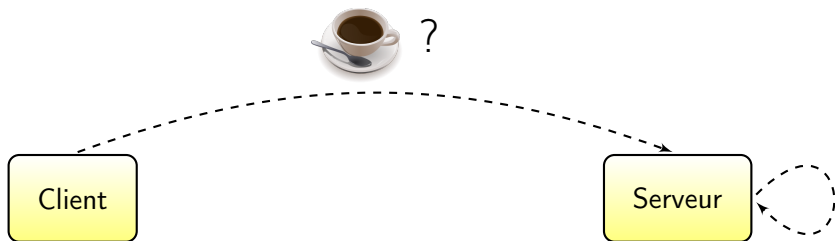
L'analogie du café

Client : « Bonjour, je voudrais un café. »



L'analogie du café

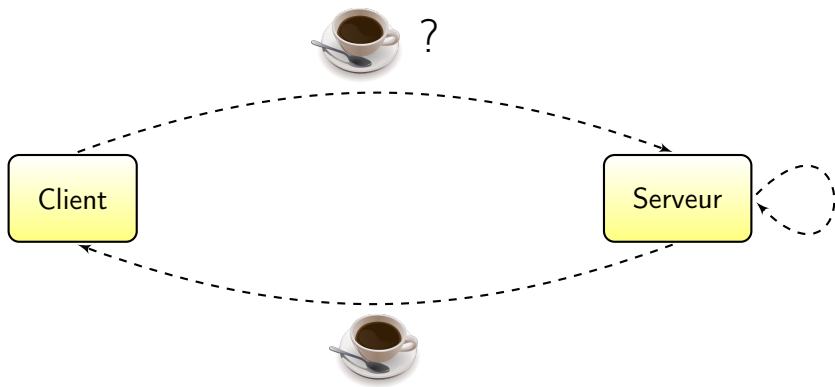
Client : « Bonjour, je voudrais un café. »
(préparation du café)



L'analogie du café

Client : « Bonjour, je voudrais un café. »
(préparation du café)

Serveur : « Bonjour, voici votre café. »



Définition

Modèle apparu dans les années 1970 (Xerox PARC)

Définition (Serveur)

Un **serveur** est un ordinateur (et/ou un programme informatique) offrant un **service** ou une **ressource** sur un réseau.

Définition (Client)

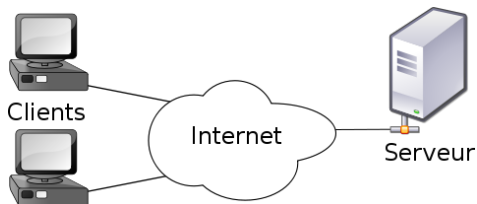
Un **client** est un programme informatique contactant un serveur via un réseau, afin de bénéficier d'un **service** ou d'une **ressource**.

Communication client-serveur

Le client et le serveur communiquent via un **protocole de communication**.

Les protocoles clients-serveurs se situent au niveau de la couche **application** du modèle OSI.

Plusieurs clients peuvent se connecter à un même serveur.



Applications du modèle client-serveur

- Serveur de **courrier électronique**
 - ▶ Protocoles courants : SMTP, POP, IMAP
- Serveurs de **pages Web**
 - ▶ Protocole : généralement **HTTP** (parfois XML Socket)

1 Architecture client-serveur

- Généralités
- Client-serveur et Web

2 HTML

3 CSS

4 Sites Web adaptatifs

Serveurs Web : matériel

Serveur Web : ordinateur affecté au stockage des pages Web et au traitement des requêtes provenant des clients



Le premier serveur Web



Salle de serveurs (2009)

- **Petits sites Web** : un serveur **partagé** (ou mutualisé), affecté à plusieurs sites Web
- **Grands sites Web** : un ou plusieurs serveurs **dédiés** à ce site
 - ▶ Google reposerait sur près d'**un million de serveurs** dans le monde

Serveurs Web : logiciel

Systèmes d'exploitation courants :

- Variantes de **Linux** (67%)
- Windows Server (Microsoft)
- Plus rarement : OSX (Apple)

Serveur HTTP courant :

- Apache

Logiciels courants :

- MySQL (base de données), PHP (pages dynamiques)

Un exemple de client : le navigateur

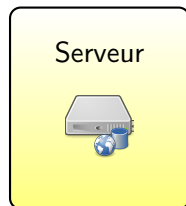
Le client est le **navigateur Web**

- Il supporte au minimum le protocole HTTP

Clients les plus courants aujourd'hui

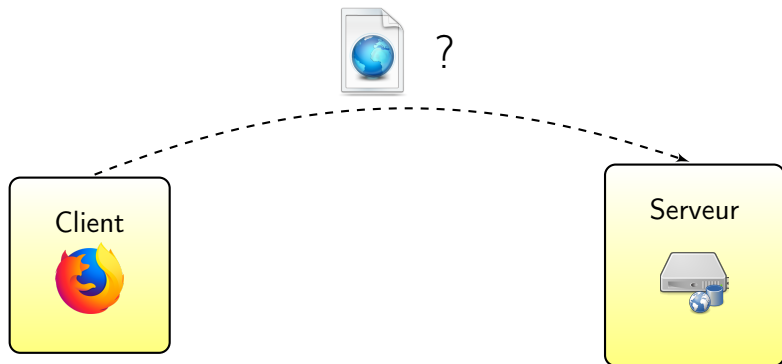
Nom	Depuis	Licence
Android (navigateur)	2008	GNU GPL
Chromium	2008	BSD (etc.)
Google Chrome	2008	propriétaire
Internet Explorer (et Edge)	1995	propriétaire
Mozilla Firefox	2002	MPL
Opera	1995	propriétaire
Safari	2003	propriétaire

Requête client-serveur



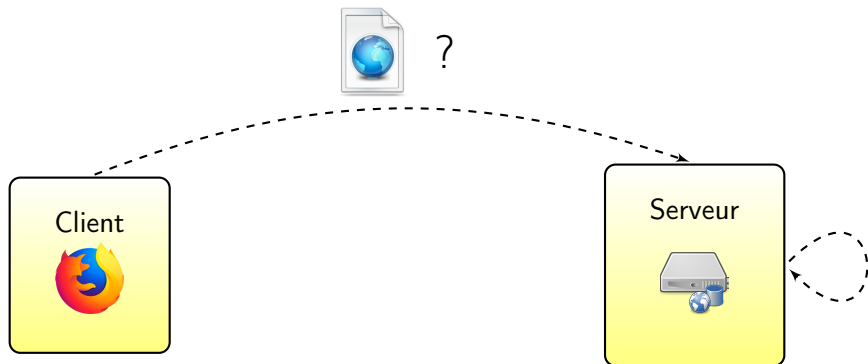
Requête client-serveur

Client : « Bonjour, je voudrais `www.univ-paris13.fr` »



Requête client-serveur

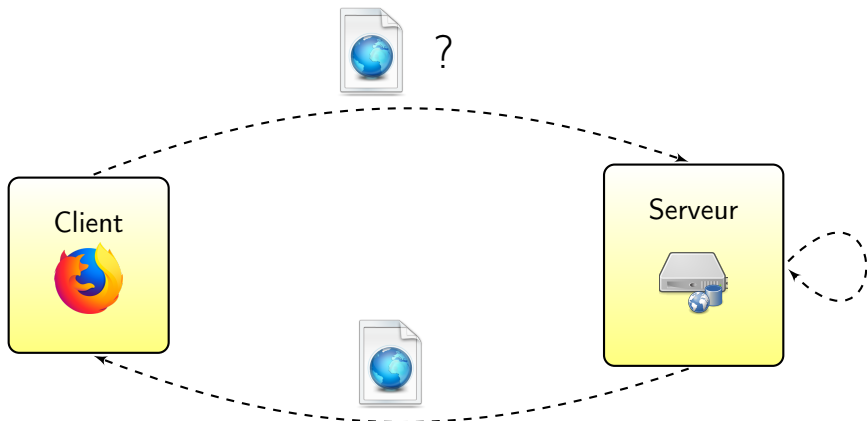
Client : « Bonjour, je voudrais `www.univ-paris13.fr` »
(préparation de la page Web)



Requête client-serveur

Client : « Bonjour, je voudrais `www.univ-paris13.fr` »
(préparation de la page Web)

Serveur : « Bonjour, voici la page Web »



Protocole HTTP et URL

Protocole HTTP

- *Hypertext Transfer Protocol*
- Port 80
- Couche application du modèle Internet

Basé sur les adresses Web ou URL

- *Uniform Resource Locator*
- Transformées en adresse IP par un système de résolution de noms (DNS)
 - ▶ `www.univ-paris13.fr` \Rightarrow `194.254.164.6`

Avantages et inconvénients du modèle client-serveur

- 😊 Calculs à la charge du serveur
 - ▶ Création de pages dynamiques (sauf Javascript)
 - ▶ Recherche d'information dans des bases de données immenses (moteurs de recherche)
- 😊 Client léger possible
 - ▶ Fonctionnant sur téléphone ou tablette
- 😞 Disponibilité du serveur : **point critique**
 - ▶ « Comment faire si l'on veut redémarrer le serveur correspondant à `www.google.fr` ? »
 - ▶ **Exemple** : OVH garantit une disponibilité de 99,9%

Avantages et inconvénients du modèle client-serveur

- 😊 Calculs à la charge du serveur
 - ▶ Création de pages dynamiques (sauf Javascript)
 - ▶ Recherche d'information dans des bases de données immenses (moteurs de recherche)
- 😊 Client léger possible
 - ▶ Fonctionnant sur téléphone ou tablette
- ☹️ **Disponibilité** du serveur : **point critique**
 - ▶ « Comment faire si l'on veut redémarrer le serveur correspondant à `www.google.fr` ? »
 - ▶ **Exemple** : OVH garantit une disponibilité de 99,9%
 - ⇒ près de **9h** de panne par an

Avantages et inconvénients du modèle client-serveur

😊 Calculs à la charge du serveur

- ▶ Création de pages dynamiques (sauf Javascript)
- ▶ Recherche d'information dans des bases de données immenses (moteurs de recherche)

😊 Client léger possible

- ▶ Fonctionnant sur téléphone ou tablette

☹ Disponibilité du serveur : point critique

- ▶ « Comment faire si l'on veut redémarrer le serveur correspondant à `www.google.fr` ? »
- ▶ Exemple : OVH garantit une disponibilité de 99,9%
 - ⇒ près de 9h de panne par an
 - ★ double panne géante d'OVH du 9 novembre 2017
 - ★ incendie du 10 mars 2021 sur le site de Strasbourg

Plan: HTML

- 1 Architecture client-serveur
- 2 **HTML**
 - Définitions
 - Histoire du HTML
 - Un langage standardisé
 - Structure d'un document HTML5
 - Syntaxe de HTML5
- 3 CSS
- 4 Sites Web adaptatifs

Internet

Définition (Internet)

Internet est un ensemble de réseaux interconnectés, utilisant un ensemble de protocoles de communication et d'échanges de données standardisés.

Internet est donc un **réseau de réseaux**.

Orthographe recommandée : **Internet**

(Variantes : l'Internet, l'internet, internet, les Internets)

World Wide Web

Définition (World Wide Web)

Le **World Wide Web** est un système fonctionnant au-dessus d'Internet, et basé sur des **liens hypertextes**. Il est constitué de **pages Web** organisées en **sites Web**, lesquelles sont consultables grâce à un **navigateur Web**.

Autres noms :

- Le Web
- La toile

Remarque :

- Le World Wide Web n'est qu'une application d'Internet (aux côtés d'autres comme le courrier électronique, la voix sur IP, etc.)

HTML

Définition (HTML)

HTML (HyperText Markup Language), ou langage de balisage d'hypertexte, est un langage standardisé utilisé pour la description des pages Web.

Hyperlien : permet de passer d'un document à un autre

HTML est aujourd'hui *de facto* le seul langage de description de pages Web.

HTML : remarques

HTML ne doit pas être confondu avec :

- **PHP** : langage dont une application possible est d'être utilisé côté serveur pour générer du code HTML
- **(My)SQL** : langage de requêtes d'accès à une base de données dont une application possible est d'être utilisé côté serveur, par exemple avec PHP, pour générer du code HTML
- **JavaScript** : langage orienté objet pouvant être utilisé côté client pour générer du code HTML
- **jQuery** : bibliothèque libre de JavaScript
- **Ajax** : architecture combinant notamment JavaScript, XML et CSS pour construire des applications Web

1 Architecture client-serveur

2 HTML

- Définitions
- **Histoire du HTML**
- Un langage standardisé
- Structure d'un document HTML5
- Syntaxe de HTML5

3 CSS

4 Sites Web adaptatifs

Au commencement : le texte

Années 1980 : nécessité de publier, partager et retrouver des documents

```
Le titre de mon document
```

```
Un paragraphe dans mon document.
```

```
Un autre paragraphe qui mentionne une autre page Web.
```

Les liens hypertextes

Fin des années 1980 : les liens hypertextes

- Permettent de pointer vers une autre page

```
Le titre de mon document
```

```
Un paragraphe dans mon document.
```

```
Un autre paragraphe qui mentionne une
```

```
<A HREF="http://www.cern.fr">autre page Web</A>.
```

(Code évidemment non conforme à la norme HTML actuelle.)

Les balises

1991 : proposition de **balises**

```
<TITLE>Le titre de mon document</TITLE>
```

Un paragraphe dans mon document.

```
<P>
```

Un autre paragraphe qui mentionne une

```
<A HREF="http://www.cern.fr">autre page Web</A>.
```

(Code évidemment non conforme à la norme HTML actuelle.)

L'influence des navigateurs

1991–1993 : en l'absence de standard, ce sont les **navigateurs Web** (Mosaic, Netscape) qui « décident » *de facto* de l'évolution de HTML

L'influence des navigateurs

1991–1993 : en l'absence de standard, ce sont les **navigateurs Web** (Mosaic, Netscape) qui « décident » *de facto* de l'évolution de HTML

Ajouts notables des navigateurs :

- les images (balise ``)
- diverses balises qui seront « supprimées » par la suite (texte clignotant, centré, etc.)

```
<TITLE>Le titre de mon document</TITLE>
```

```
Un paragraphe dans mon document.
```

```
<P>
```

```
<BLINK>Du texte clignotant</BLINK>
```

```
<CENTER>Du texte centré</CENTER>
```

```
<IMG SRC="mon_image.gif">
```

(Code évidemment non conforme à la norme HTML actuelle.)

HTML 2.0 : la première spécification

- 1995 : spécification de HTML 2.0 par l'**IETF**
 - ▶ *Internet Engineering Task Force*, qui élabore les standards d'Internet
- 1996 : prise en main par le **W3C**
 - ▶ *World Wide Web Consortium* (448 organisations membres en 2021) qui élabore les standards du Web
- 1997 : publication de **HTML 3.2** par le W3C (sous forme de recommandation)

HTML 4.01 : la stabilité

- 1997–1998 : publication de **HTML 4.0** par le W3C (sous forme de recommandation)
- 3 variations
 - ▶ *Strict* : éléments obsolètes interdits
 - ▶ *Transitional* (transitoire) : éléments obsolètes autorisés
 - ▶ *Frameset* (cadre) : assemblage d'éléments de type cadre
- Les **éléments obsolètes** sont des éléments autorisés par les navigateurs Web de l'époque, mais destinés à être remplacés par les feuilles de style
- 1999 : publication par le W3C de **HTML 4.01**

HTML5 : le nouveau standard

- 2008 : publication par le W3C d'un **premier brouillon** (*draft*) de la spécification de **HTML5**
 - ▶ Début de la création de pages Web conformes à la pré-spécification HTML5
- 2011 : **dernier appel** pour contribuer à la spécification
- 28 octobre 2014 : publication du **standard HTML5** comme recommandation W3C

HTML5 : le nouveau standard

- 2008 : publication par le W3C d'un **premier brouillon** (*draft*) de la spécification de **HTML5**
 - ▶ Début de la création de pages Web conformes à la pré-spécification HTML5
- 2011 : **dernier appel** pour contribuer à la spécification
- 28 octobre 2014 : publication du **standard HTML5** comme recommandation W3C

Ce cours se concentre sur **HTML5**.

1 Architecture client-serveur

2 HTML

- Définitions
- Histoire du HTML
- **Un langage standardisé**
- Structure d'un document HTML5
- Syntaxe de HTML5

3 CSS

4 Sites Web adaptatifs

L'intérêt du respect du standard

- **Interopérabilité, compatibilité** entre navigateurs
- **Meilleur référencement**
 - ▶ Google favoriserait le référencement des sites Web dont le code HTML est conforme aux standards

Limite du respect du standard

Limites de l'interopérabilité

Une page Web peut ne pas s'afficher de la même façon selon les navigateurs, même si elle respecte entièrement le standard HTML5, et même si les navigateurs respectent eux aussi le standard.

Il convient donc d'en **tester l'affichage** sur les principaux navigateurs.

Néanmoins, respecter le standard limite les différences potentielles d'affichage !

Validation du code HTML

À la différence de langages de programmation, un non-respect de la syntaxe de HTML n'empêchera pas l'affichage sur le navigateur.

Plus généralement, le navigateur affichera quelque chose (de non spécifié) ou n'affichera rien, mais **n'affichera jamais d'erreur.**

Validation du code HTML

À la différence de langages de programmation, un non-respect de la syntaxe de HTML n'empêchera pas l'affichage sur le navigateur.

Plus généralement, le navigateur affichera quelque chose (de non spécifié) ou n'affichera rien, mais **n'affichera jamais d'erreur**.

Validation

Il faut impérativement **valider** son code HTML avant de le publier.

Valdateur du W3C

Un outil indispensable : le **valdateur** du W3C et de la fondation Mozilla

<http://validator.w3.org/>

W3C Markup Validation Service
Check the markup (HTML, XHTML, ...) of Web documents

Jump To: [Notes and Potential Issues](#) [Congratulations · Icons](#)

This document was successfully checked as HTML5!

Result: Passed, 2 warning(s)

Source:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Le titre de mon document</title>
  </head>
  <body>
    Ce document est constitué d'une seule phrase.
  </body>
</html>
```

Encoding: utf-8 (detect automatically)

Doctype: HTML5 (detect automatically)

Root Element: html



Try now the [W3C Validator Suite™](#) premium service that checks your entire website and evaluates its conformance with W3C open standards to quickly identify those portions of your website that need your attention.

1 Architecture client-serveur

2 HTML

- Définitions
- Histoire du HTML
- Un langage standardisé
- **Structure d'un document HTML5**
 - Principe des balises
 - En-tête
- Syntaxe de HTML5

3 CSS

4 Sites Web adaptatifs

Un langage à base de balises

HTML repose sur la notion de **balises** (*tags*)

Deux types de balises :

- 1 Les balises qui sont ouvertes puis fermées, et encadrent du **contenu**
 - ▶ Exemple : `Un peu d'italique`
- 2 Les balises qui s'ouvrent et se ferment en même temps
 - ▶ Exemple : ``

```
<h1>Un exemple de titre</h1>
<hr />
<div>
  <p>
    Du texte <b>gras</b>, puis <em>italique</em>,
    puis <b><em>gras et italique</em></b>.
  </p>
  
</div>
```

Balises et attributs

Certaines balises peuvent posséder des **attributs**.

Exemples :

- `<meta charset="utf-8" />`

- ``

- `<p style="text-align:right">
 Du texte aligné à droite
</p>`

Balises et hiérarchie

Les balises sont structurées de façon **hiérarchique**.

Chaque balise **héríte** des propriétés de la balise de niveau supérieur – à moins qu'elle ne les redéfinisse.

```
<p>Ce texte est <b>seulement en gras</b>.</p>
```

```
<p>Ce texte est <b>seulement en gras <i>et  
cette partie est en gras et en italique</i></b>.</p>
```


Balises et hiérarchie

Les balises sont structurées de façon **hiérarchique**.

Chaque balise **hérite** des propriétés de la balise de niveau supérieur – à moins qu'elle ne les redéfinisse.

```
<p>Ce texte est <b>seulement en gras</b>.</p>  
<p>Ce texte est <b>seulement en gras <i>et  
cette partie est en gras et en italique</i></b>.</p>
```

Ordre de fermeture

Il est impératif de fermer les balises dans l'ordre inverse duquel on les a ouvertes.

Commentaires

Les commentaires sont du code HTML qui n'est pas exécuté, donc **invisible sur le navigateur**.

Syntaxe : `<!-- Texte entre commentaires -->`

```
<h1>Un titre (qui va s'afficher)</h1>
```

```
<p>Cette phrase va s'afficher sur le navigateur.
```

```
  <!-- En revanche, cette phrase ne s'affichera pas -->
```

```
</p>
```

Commentaires

Les commentaires sont du code HTML qui n'est pas exécuté, donc **invisible sur le navigateur**.

Syntaxe : `<!-- Texte entre commentaires -->`

```
<h1>Un titre (qui va s'afficher)</h1>

<p>Cette phrase va s'afficher sur le navigateur.
  <!-- En revanche, cette phrase ne s'affichera pas -->
</p>
```

Attention

Les commentaires restent visibles dans la **source** de la page Web, qui est accessible depuis le client.

- Attention à ne pas y laisser d'informations confidentielles !

Structuration d'un document HTML

Deux grandes parties :

- **En-tête** : définition du titre, du codage, des mots-clés, etc.
 - ▶ Information destinée aux machines (navigateur, robots, etc.)
- **Corps** : contenu de la page Web
 - ▶ Information destinée à l'humain (et aux machines)

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <!-- Ici, l'en-tête -->
  </head>

  <body>
    <!-- Ici, le corps de la page -->
  </body>
</html>
```

Un exemple minimal (syntaxe conforme à HTML5)

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8" />
    <title>Le titre de mon document</title>
  </head>

  <body>
    <p>Ce document est constitué d'une seule phrase.</p>
  </body>
</html>
```

Document Type Definition

Un document conforme à la syntaxe de HTML5 commence nécessairement par `<!DOCTYPE html>`.

Il s'agit de la **définition du type du document** (DTD), qui décrit un modèle de document XML ou SGML.

La balise `title` (1/2)

- Située dans l'**en-tête** du document
 - ▶ Entre les balises `<head>` et `</head>`
- Seule **balise obligatoire** de l'en-tête
- Syntaxe :

```
...  
  <head>  
    <title>Le titre de ma page Web</title>  
  </head>  
...
```

La balise `title` (2/2)

Son contenu définit le **titre** du document

- Auparavant systématiquement affiché dans **le titre de la fenêtre du navigateur** ou **de l'onglet**
- De moins en moins aujourd'hui
- Améliore grandement le **référencement**



Les balises meta

- Situées dans l'**en-tête** du document
 - ▶ Entre les balises `<head>` et `</head>`
- Ordre non spécifié
- Permettent de donner des informations au **navigateur** et aux **moteurs de recherche**
 - ▶ Codage des caractères
 - ▶ Auteur, description, mots-clés de la page
- Ne s'affichent jamais directement sur la page

Codage des caractères

- Syntaxe :

```
<meta charset="NOM_DU_CODAGE" />
```

- Valeurs possibles

- ▶ utf-8 : unicode (recommandé)

```
<meta charset="utf-8" />
```

- ▶ iso-8859-1 : alphabet latin
- ▶ et bien d'autres

- La présence de cette balise n'est pas obligatoire selon le W3C, mais **très très très fortement recommandée** (comprendre obligatoire).

Liste complète des valeurs :

<https://www.iana.org/assignments/character-sets/character-sets.xhtml>

Description, mots-clés, auteur

Syntaxe :

```
<meta name="description" content="Introduction
aux technologies Web" />
<meta name="keywords" content="cours,HTML,IUT de
Villetaneuse,R109" />
<meta name="author" content="Le prof" />
```

Description, mots-clés, auteur

Syntaxe :

```
<meta name="description" content="Introduction
aux technologies Web" />
<meta name="keywords" content="cours,HTML,IUT de
Villetaneuse,R109" />
<meta name="author" content="Le prof" />
```

Important

Remplir ces champs est important pour le référencement dans les moteurs de recherche.

Un exemple d'en-tête

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8" />
    <meta name="description" content="Une page Web
      vraiment intéressante" />
    <meta name="keywords" content="cours,HTML,IUT de
      Villetaneuse,R109" />
    <meta name="author" content="Le prof" />
    <title>Le titre de ma page Web</title>
  </head>

  <body>
    ...
  </body>
</html>
```

En-tête : bonnes pratiques

Bonnes pratiques

Il est très fortement conseillé de toujours spécifier, en plus de la balise

`<title>` :

- le codage des caractères,
- l'auteur,
- la description,
- les mots-clés.

1 Architecture client-serveur

2 HTML

- Définitions
- Histoire du HTML
- Un langage standardisé
- Structure d'un document HTML5
- **Syntaxe de HTML5**
 - Structure de la page
 - Styles de texte
 - Listes et énumérations
 - Tableaux
 - Médias

3 CSS

4 Sites Web adaptatifs

Sections

Balise `<div>` : élément structurel (division ou section)

Bonnes pratiques

Il est d'usage d'organiser la page Web en `<div>` imbriquées.

```
<div>
  <div>
    <!-- Une première partie -->
  </div>
  <div>
    <!-- Une seconde partie -->
  </div>
</div>
```


Paragraphes

- Balise `<p>` : paragraphe
- Balise `
` ou `
` : saut de ligne

Bonne pratique

Il est d'usage de créer un second paragraphe (`<p>`) plutôt qu'un saut de ligne (`
`).

```
<div>
  <p>Un premier paragraphe.</p>
  <p>Un second paragraphe.</p>
</div>
```

Espaces

- Les espaces affichées en HTML s'affichent également sur le navigateur.
 - ▶ Attention donc à ne pas multiplier les espaces !
- En revanche, les sauts de ligne en HTML deviennent des espaces sur le navigateur.
 - ▶ Pour afficher un saut de ligne, utiliser `
`

```
<p>Cette phrase va s'afficher
```

```
sur une seule ligne.<br />Cette phrase sera dessous.</p>
```

Cette phrase va s'afficher sur une seule ligne.

Cette phrase sera dessous.

Espaces insécables

Les **espaces insécables** s'affichent en utilisant ` `;

- Pour « *non-breakable blank space* »

Usage en français :

- Après «
- Avant **? ! ; : »**

```
< &nbsp; Comment allez-vous &nbsp; ? &nbsp; », demanda-t-elle.
```

```
« Comment allez-vous ? », demanda-t-elle.
```

Quelques caractères spéciaux

Certains caractères sont **interdits** dans le code HTML (en dehors des balises).

Il faut donc utiliser un codage du caractère :

- `<` : `<`;
- `>` : `>`;
- `&` : `&`;

Exemple :

```
Le DUT R&amp;T &gt; le reste du monde
```

```
Le DUT R&T > le reste du monde
```

Les titres

Syntaxe :

- `<h1>` : titre de premier niveau
- `<h2>` : titre de second niveau
- ...
- `<h6>` : titre de sixième niveau

Bonnes pratiques

Il est d'usage de n'avoir qu'un seul titre `<h1>` dans le document, et de respecter l'ordre croissant des niveaux.

Les titres : exemple

```
<h1>Le titre général</h1>
<h2>Une première grande section</h2>
<p>Un texte d'introduction</p>
<h3>Un sous-titre dans la première section</h3>
<h3>Un autre sous-titre dans la première section</h3>

<h2>Une seconde grande section</h2>
```

affichera

```
Le titre général
Une première grande section
Un texte d'introduction
Un sous-titre dans la première section
Un autre sous-titre dans la première section
Une seconde grande section
```

Les liens hypertextes externes

- Syntaxe :

```
<a href="LA_CIBLE_DU_LIEN">Le texte à afficher</a>
```

Affiche

[Le texte à afficher](#)

- Deux types de liens :

- ▶ Liens **relatifs**

```
<a href="index.html">Texte</a>
```

Pointe sur la page [index.html](#) dans le répertoire courant

- ▶ Liens **absolus**

```
<a href="http://lipn.fr/~petrucci/R109/">Texte</a>
```

Les liens hypertextes internes

- On peut effectuer un lien vers **une autre partie du même document**
- Syntaxe pour définir l'endroit dans le document (destination)
 - ▶ À l'aide de `id`
 - ▶ Exemple : `<h2 id="macible">Mes films préférés</h2>`
- Syntaxe pour définir le lien
 - ▶ `Remonter aux films préférés`

Affiche

Mes films préférés

...

[Remonter aux films préférés](#)

Les liens hypertextes vers une adresse électronique

- On peut effectuer un lien vers **une adresse électronique**
- Syntaxe :
 - ▶ `Écrivez-moi`

Affiche

[Écrivez-moi](mailto:moi@univ-paris13.fr)

Attention

Cette utilisation est fortement déconseillée, du fait de la présence sur Internet de nombreux **robots malveillants** qui récupèrent les adresses électroniques en clair pour envoyer des **pourriels**.

Un exemple un peu plus fourni (ex1.html)

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>Le titre de mon document</title>
    <meta charset="utf-8" />
    <meta name="author" content="L'auteur de la page" />
  </head>

  <body>
    <h1>Le titre de la page</h1>
    <p>Une phrase avec <a href="index.html">un lien</a>.</p>
    <h2>Une section</h2>
    <p>Une autre phrase.</p>
    <h2>Une seconde section</h2>
    <p>Encore une autre phrase.</p>
  </body>
</html>
```

Nouvelles balises HTML5

- Peuvent être vues comme des variations **sémantiques** de `<div>`
 - ▶ Pas d'incidence sur l'affichage ou la mise en page
 - ▶ Peuvent être **imbriquées**
- Syntaxe :
 - ▶ `<section>` : un bloc de contenu cohérent
 - ▶ `<article>` : un article au contenu cohérent
 - ▶ `<aside>` : contenu accessoire
 - ▶ `<header>` : en-tête (de page, de section, etc.)
 - ▶ `<footer>` : pied (de page, de section, etc.)
 - ▶ `<nav>` : informations de navigation (liens)

Nouvelles balises HTML5 : exemple (ex2.html)

```
...
<header>Le haut de la page</header>
<article>
  <h2>Le titre de mon article</h2>
  <header>Un chapeau pour mon article</header>
  <p>Premier paragraphe.....</p>
  <p>Second paragraphe.....</p>
  <footer>Infos finales sur mon article</footer>
  <aside>Voir un autre article sur le même sujet</aside>
</article>
<footer>
  <nav>
    <a href="index.html">Retour à l'accueil</a>
  </nav>
</footer>
...
```

Nouvelles balises HTML5 : attention

En-tête : attention aux confusions

Il convient de bien différencier :

- 1 l'**en-tête syntaxique** du document HTML (balise `<head>` unique, contenu non affiché), et
- 2 les **en-têtes sémantiques** dans le corps (`<body>`) du document (balises `<header>` éventuellement multiples).

Gras, italique, souligné

Syntaxe :

- `...` (*bold*) : **gras**
- `<i>...</i>` (*italique*) : *italique*
- `...` (*emphasis*) : *emphase*
(s'affiche en général comme italique)
- `<u>...</u>` (*underline*) : souligné

Gras, italique, souligné

Syntaxe :

- `...` (*bold*) : **gras**
- `<i>...</i>` (*italique*) : *italique*
- `...` (*emphasis*) : *emphase*
(s'affiche en général comme italique)
- `<u>...</u>` (*underline*) : souligné

Bonne pratique

Le soulignage ne devrait jamais être utilisé en informatique, à l'exception (éventuelle) des liens hypertextes.

Couleurs

- Syntaxe :

- ▶ Attribut `style="color:COULEUR"`
- ▶ Utilisé avec la balise `` au sein d'un paragraphe

```
Le grand <span style="color:blue">bleu</span>
```

- ▶ Peut également être utilisé avec `<p>`, `<div>`, etc.

```
<p style="color:green">Le rayon vert</p>
```


Couleurs

- Syntaxe :

- ▶ Attribut `style="color:COULEUR"`
- ▶ Utilisé avec la balise `` au sein d'un paragraphe

```
Le grand <span style="color:blue">bleu</span>
```

- ▶ Peut également être utilisé avec `<p>`, `<div>`, etc.

```
<p style="color:green">Le rayon vert</p>
```

- Valeurs des couleurs :

- ▶ **Prédéfinies** : `green`, `blue`, `red`, etc.
 - ★ Voir liste complète par exemple sur :
<http://www.standardista.com/css3/cssnamed-hsl-and-rgb-colors/>
- ▶ **Hexadécimales** : format `#RRVVB` (rouge vert bleu)
 - ★ Exemple : `#FFFF00` =

Couleurs

- Syntaxe :

- ▶ Attribut `style="color:COULEUR"`
- ▶ Utilisé avec la balise `` au sein d'un paragraphe

```
Le grand <span style="color:blue">bleu</span>
```

- ▶ Peut également être utilisé avec `<p>`, `<div>`, etc.

```
<p style="color:green">Le rayon vert</p>
```

- Valeurs des couleurs :

- ▶ **Prédéfinies** : `green`, `blue`, `red`, etc.
 - ★ Voir liste complète par exemple sur :
<http://www.standardista.com/css3/cssnamed-hsl-and-rgb-colors/>
- ▶ **Hexadécimales** : format `#RRVVBB` (rouge vert bleu)
 - ★ Exemple : `#FFFF00` = jaune

Bonne pratique

En général, le formatage du texte est géré non par HTML mais par une **feuille de style CSS** (voir plus loin)

Listes à puces

Syntaxe : (``=unordered list, ``=line)

```
<p>J'aime bien :</p>
<ul>
  <li>Le HTML</li>
  <li>Le CSS</li>
</ul>
```


Listes à puces : remarques

Remarques

- 1 Les listes à puces peuvent être **imbriquées**.
- 2 L'affichage des listes à puces peut être considérablement personnalisé par les **feuilles de style**.
- 3 Il est d'usage d'utiliser l'environnement `` pour la création des **menus** (même si ceux-ci n'ont pas l'apparence d'une liste à puces).

Énumérations

Syntaxe : (=ordered list)

```
<p>J'aime bien :</p>
<ol>
  <li>Le HTML</li>
  <li>Le CSS</li>
</ol>
```


Descriptions

Syntaxe : (<dl>=description list, <dt>=description type, <dd>=description data)

```
<dl>
  <dt>HTML</dt>
  <dd>Langage de description de pages Web</dd>
  <dt>CSS</dt>
  <dd>Langage décrivant la présentation de pages HTML</dd>
</dl>
```

Affichage par défaut :

HTML

Langage de description de pages Web

CSS

Langage décrivant la présentation de pages HTML

Tableaux : exemple (1/2)

Syntaxe : (<tr>=table row, <th>=table head, <td>=table data)

```
<table border="1">
  <tr>
    <th>Titre du film</th>
    <th>Année de sortie</th>
  </tr>
  <tr>
    <td><em>Drive</em></td>
    <td>2011</td>
  </tr>
  <tr>
    <td><em>The Neon Demon</em></td>
    <td>2016</td>
  </tr>
</table>
```

Tableaux : exemple (2/2)

Affichage :

Titre du film	Année de sortie
<i>Drive</i>	2011
<i>The Neon Demon</i>	2016

Un tableau plus complexe : syntaxe (1/2)

Syntaxe : (<thead>=table header, <tbody>=table body)

```
<table border="1">
  <thead>
    <tr>
      <th colspan="2">Films de Wong Kar-Wai</th>
    </tr>
    <tr>
      <th>Année</th>
      <th>Titre</th>
    </tr>
  </thead>
```

Un tableau plus complexe : syntaxe (2/2)

```
<tbody>
  <tr>
    <td rowspan="2">1994</td>
    <td><em>Les Cendres du temps</em></td>
  </tr>
  <tr>
    <td><em>Chungking Express</em></td>
  </tr>
  <tr>
    <td>2000</td>
    <td><em>In the Mood for Love</em></td>
  </tr>
</tbody>
</table>
```

Un tableau plus complexe : affichage (ex3.html)

Affichage :

Films de Wong Kar-Wai	
Année	Titre
1994	<i>Les Cendres du temps</i>
	<i>Chungking Express</i>
2000	<i>In the Mood for Love</i>

Tableaux : bonne pratique

Bonne pratique

Il est d'usage de n'utiliser l'environnement `<table>` que pour les tableaux. Pour aligner des éléments lors de la mise en page, on utilise les environnements `<div>` et des feuilles de style.

Images (ex4.html)

- Syntaxe :

```

```

- La plupart des extensions courantes sont prises en charge par HTML5 (.jpg, .png, .gif, etc.).

Exemple :

```

```

Attention : texte de remplacement

- Le texte de remplacement (attribut `alt`) est **obligatoire**.
- Il est affiché par le navigateur si l'image n'est pas trouvée (fichier manquant), ou dans certains cas spécifiques (navigateur Web pour non-voyants).

Audio (ex5.html)

- Syntaxe :

```
<audio controls>  
  <source src="Reflection.ogg" type="audio/ogg">  
  <source src="Reflection.mp3" type="audio/mpeg">  
  Votre navigateur ne prend pas en charge les formats  
    audio proposés.  
</audio>
```

- Résultat :



- Les différents formats sont essayés dans l'ordre descendant, jusqu'au premier format pris en charge par le navigateur.

Audio et navigateurs

Attention

Tous les navigateurs ne prennent pas tous les formats en charge.

Navigateur	AAC	MP3	Ogg	Wav
Chrome	Oui	Oui	Oui	Oui
Chromium	Non	Oui	Oui	Oui
Firefox	Partiel	Partiel	Oui	Oui
Internet Explorer	Oui	Oui	Non	Non
Opera	Partiel	Partiel	Oui	Oui
Safari	Oui	Oui	Non	Oui

Bonne pratique

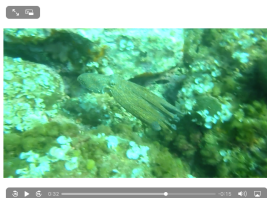
Prévoir les sons dans suffisamment de formats (au moins Ogg et MP3).

Vidéo (ex6.html)

- Syntaxe :

```
<video width="640" height="480" controls>  
  <source src="poulpe.webm" type="video/webm">  
  <source src="poulpe.mp4" type="video/mp4">  
  Votre navigateur ne prend pas en charge les formats  
    vidéo proposés.  
</video>
```

- Résultat :



- Les différents formats sont essayés dans l'ordre descendant, jusqu'au premier format pris en charge par le navigateur.

Vidéo et navigateurs

Attention

Tous les navigateurs ne prennent pas tous les formats en charge.

Navigateur	MP4	Ogg	WebM
Chrome	Partiel	Partiel	Oui
Chromium	Non	Partiel	Oui
Firefox	Partiel	Oui	Oui
Internet Explorer	Oui	Non	Non
Opera	Partiel	Oui	Oui
Safari	Oui	Non	Non

Bonne pratique

Prévoir les vidéos dans suffisamment de formats (au moins WebM et MP4).

Plan: CSS

1 Architecture client-serveur

2 HTML

3 **CSS**

- Définition et historique
- Syntaxe de CSS3
- Un exemple

4 Sites Web adaptatifs

CSS

Définition (CSS)

CSS (*Cascading Style Sheets*, ou feuilles de style en cascade) est un langage permettant la description de la présentation des pages HTML.

- Standard développé par le W3C.
- CSS est incrémenté par **niveaux** et non par versions
 - ▶ Le niveau 3 réutilise et étend les niveaux 1 et 2
 - ▶ Différent de HTML !

Principe : séparation de la forme et du fond

Principe

Le principe est de séparer le fond (**HTML**) de la forme (**CSS**).

Une feuille de style CSS va donc **mettre en forme** la page HTML.

Bonne pratique

Un site HTML bien conçu devrait ne contenir aucune information de style dans le document HTML.

Bref historique

- Début des années 1990 : concept de feuille de style
- 1995 : démonstration au congrès WWW
- 1996 : **CSS1**
- 1997 : **CSS2**
- 1999 : premiers brouillons pour CSS3
- 2001 : **CSS2.1** (correction de CSS2)
- 2010 : début de la réflexion pour **CSS4**
- 2011–12 : début des recommandations officielles du W3C pour des modules de **CSS3**

CSS3 est toujours **en développement**.

1 Architecture client-serveur

2 HTML

3 CSS

- Définition et historique
- Syntaxe de CSS3
 - Principes
 - Affichage du texte
 - Mise en forme de la page
- Un exemple

4 Sites Web adaptatifs

Exemple simple

Page HTML :

```
...  
<h1>Happy Together</h1>  
<p>Film de Wong Kar-Wai  
réalisé en 1997</p>  
...
```

Feuille de style :

```
body {  
    color: #000080;  
}  
  
h1 {  
    font-weight: bold;  
    font-style: italic;  
    color: red;  
    font-size: 24px;  
}
```

Exemple simple

Page HTML :

```
...  
<h1>Happy Together</h1>  
<p>Film de Wong Kar-Wai  
réalisé en 1997</p>  
...
```

Feuille de style :

```
body {  
    color: #000080;  
}  
  
h1 {  
    font-weight: bold;  
    font-style: italic;  
    color: red;  
    font-size: 24px;  
}
```

Affichage :

Happy Together

Film de Wong Kar-Wai réalisé en 1997

Syntaxe générale d'une feuille CSS

Sélecteur

```
h1 {  
    font-weight: bold;  
    font-family: Times New Roman, Times, serif;  
}
```

Diagram illustrating the syntax of a CSS rule with annotations:

- Sélecteur** (Selector): `h1`
- Propriété** (Property): `font-weight`
- Valeur** (Value): `bold`
- Propriété** (Property): `font-family`
- Valeurs** (Values): `Times New Roman, Times, serif`

Remarques :

- Fichier ayant l'extension `.css`
- Syntaxe insensible aux espaces et retours à la ligne.
 - ▶ Néanmoins, l'**indentation est fortement recommandée!**

Style inclus ou séparé ? (1/2)

☹ Moche :

```
...
  <h1 style="color:red">Happy Together</h1>
...
```

☹ Vaguement mieux :

```
<!DOCTYPE html>
<html lang="fr">
<head>
...
  <style type="text/CSS" >
    h1 {color:red;}
  </style>
...
</head>
<body>
  <h1>Happy Together</h1>
</body>
</html>
```

Style inclus ou séparé ? (2/2)

☺ **Bien :**

```
<head>
...
  <link rel="stylesheet" type="text/css"
        media="screen" href="mafeuille.css" />
...
</head>

<body>
  <h1>Happy Together</h1>
</body>
```

... où `mafeuille.css` est un **fichier séparé** définissant les propriétés de `<h1>` (et des autres balises).

Commentaires

Les commentaires sont du code CSS qui n'est pas exécuté, donc **non pris en compte par le navigateur**.

Syntaxe : `/* Texte entre commentaires */`

```
body {  
  color: #000080; /* la couleur */  
  /* text-decoration: line-through; */  
}
```

Commentaires

Les commentaires sont du code CSS qui n'est pas exécuté, donc **non pris en compte par le navigateur**.

Syntaxe : `/* Texte entre commentaires */`

```
body {  
  color: #000080; /* la couleur */  
  /* text-decoration: line-through; */  
}
```

Attention

Les commentaires restent visibles dans la **source** de la feuille de style, qui est accessible depuis le client.

- Attention à ne pas y laisser d'informations confidentielles !

Propriétés : styles

- Propriété `font-size`: **taille de police**
 - ▶ Valeurs prédéfinies absolues : `xx-small`, `x-small`, `medium`, `large`, `x-large`, `xx-large`
 - ▶ Valeurs prédéfinies relatives : `smaller`, `larger`
 - ▶ Valeur numérique : x unités, où unité $\in \{\text{pt}, \text{px}, \text{em}, \%\}$
- Propriété `font-weight`: **graisse** de la police
 - ▶ Valeurs prédéfinies : `lighter`, `normal`, `bold`, `bolder`
 - ▶ Valeurs numériques : $x \in [0; 1000]$ (1000 étant le plus gras)
- Propriété `font-style`: **inclinaison** de la police
 - ▶ Valeurs prédéfinies : `normal`, `italic`, `oblique`
- Propriété `text-decoration`: **décoration** (soulignage, surlignage)
 - ▶ Valeurs prédéfinies : `underline`, `overline`, `line-through`, `none`

Propriétés : police

- Propriété `font-family`: famille de police de caractères
 - ▶ Grandes familles : serif, sans-serif, mono, monospace
 - ▶ Nom de police : Arial, Courier New, Courier, Helvetica, Times New Roman, Times, et bien d'autres

vif zéphyr | vif zéphyr | vif zéphyr
serif | sans-serif | monospace

Bonnes pratiques typographiques

- 1 Les titres sont souvent en sans-serif, et le corps en serif.
- 2 Il convient d'éviter de mélanger outre mesure les familles de polices dans un même document.

Propriétés : capitales et petites capitales (1/2)

- Propriété `font-transform`: **police en capitales / minuscules**
 - ▶ Valeurs : `capitalize` (première lettre), `uppercase` (capitales), `lowercase` (minuscules), `none` (normal)
- Propriété `font-variant`: **petites capitales**
 - ▶ Valeurs : `normal`, `small-caps`

Propriétés : capitales et petites capitales (2/2)

```
<p>Un bout de texte en  
  <span style="font-transform:uppercase">capitales</span> et  
  en <span style="font-variant:small-caps">Petites  
  Capitales</span>.</p>
```

Affiche :

Un bout de texte en CAPITALES et en Petites Capitales.

Bonne pratique

On ne doit jamais écrire de code HTML directement en majuscules, mais plutôt utiliser les attributs CSS correspondants.

Propriétés : couleurs

- Propriété `color`: couleur de la police

- ▶ Couleurs prédéfinies : `red`, `blue`, `green`, `orange`, etc.

- ★ Voir par exemple web-color.aliasdmc.fr/

- ▶ Valeurs utilisateur :

- ★ Hexadécimal : `#RRVBBB`

- ★ RVB : `rgb(r, g, b)` avec valeurs de 0 à 255

- ★ RVB : `rgb(r%, g%, b%)` avec valeurs de 0 à 100

- ★ RVB + transparence : `rgba(r, g, b, a)` avec valeurs de 0 à 255, et transparence (`a`) entre 0 et 1

- ★ HSL (teinte / saturation / lumière) : `hsl` et `hsla`

Identifiants et classes

● Identifiants

- ▶ Syntaxe HTML : `<h1 id="titre">`
- ▶ Syntaxe CSS : `h1 #titre { ... }`
- ▶ Utilisation **unique** (un seul `id="titre"` dans la page HTML)

● Classes

- ▶ Syntaxe HTML : `<h1 class="titre">`
- ▶ Syntaxe CSS : `h1 .titre { ... }`
- ▶ Utilisation **multiple** (plusieurs `class="titre"` autorisés dans la page HTML)

Identifiants et classes : exemple

Page HTML :

```
<h2 id="premier">As Tears Go By</h2>
<h2>Nos années sauvages</h2>
<h2 class="bleu">Les Cendres du temps</h2>
<h2>Chungking Express</h2>
<h2 class="bleu">Les Anges déchus</h2>
```

Feuille de style :

```
h2 {
  font-style: italic;
}
#premier {
  color: red;
}
.bleu {
  color: blue;
}
```

Identifiants et classes : exemple (affichage)

As Tears Go By

Nos années sauvages

Les Cendres du temps

Chungking Express

Les Anges déchus

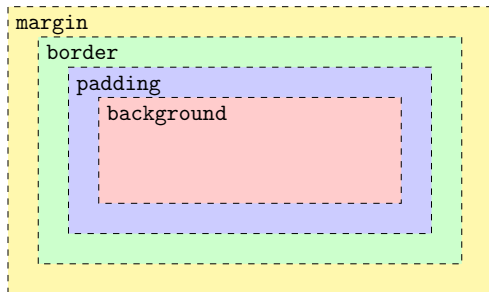
Héritage

- Les propriétés CSS **héritent** des propriétés parentes.
- Héritage par raffinement de propriété :
 - ▶ `h2 #premier` hérite de `h2`
 - ▶ `h2 .bleu` hérite de `h2` (mais pas de `h2 #premier`)
- Héritage par propriété sémantiquement parente :
 - ▶ `h2` hérite de `body`

Marges et bordures

Tout élément HTML est considéré comme contenu dans une **boîte**, dont on peut définir :

- l'aire intérieure (**background**)
- l'espace intérieur (**padding**)
- la bordure (**border**)
- la marge extérieure (**margin**)



Marges et bordures : exemple

```
<p class="boiteorange">Ce petit exemple est tiré du Wikibook  
sur CSS (Cascading Style Sheets).</p>
```

```
p .boiteorange {  
border: 1px dotted black;  
padding-top: 1em;  
padding-right: 2em;  
padding-bottom: 3em;  
padding-left: 4em;  
background-color: #fc9;  
width: 10em;  
text-align: justify  
}
```

Ce petit exemple
est tiré du
Wikibook sur CSS
(Cascading Style
Sheets).

Le validateur du W3C

Comme pour HTML, le W3C et la fondation Mozilla mettent à disposition un validateur de code CSS3.

<http://jigsaw.w3.org/css-validator/>

Un exemple sans CSS

Présentation de l'IUT de Villetaneuse

Menu

- [Le BUT R&T](#)
- [Le BUT en apprentissage](#)
- [La licence professionnelle](#)
- [Autres départements](#)

Le BUT R&T

Le [BUT](#) de département réseaux et télécommunications de l'[IUT de Villetaneuse](#) permet d'acquies plusieurs compétences :

1. Administrer et sécuriser les réseaux et l'Internet,
2. Connecter les entreprises et les usagers, et
3. Créer des outils et applications informatiques pour les R&T.

Deux parcours sont proposés :

- *Cyber* : Cybersécurité
- *ROM* : Réseaux Opérateurs et Multimédia

La licence professionnelle ASSR

Description

La licence professionnelle ASSR du département réseaux et télécommunications de l'[IUT de Villetaneuse](#) permet d'acquies plusieurs compétences en termes de :

- réseaux,
- sécurité et techniques d'attaques,
- administration de postes sous Linux ou Windows,
- communication et expression.

Deux versions

En formation initiale

La licence professionnelle peut être intégrée après un DUT réseaux et télécommunications, un BTS, ou d'autres formations similaires.

En alternance

Le contenu de la licence en alternance est le même que la licence en formation initiale, seules les modalités diffèrent. Dans la version en alternance, les étudiants ont le statut d'apprentis, et passent la moitié de leur temps en entreprise.

Pour en savoir plus, consultez le site Internet de la [licence professionnelle ASSR](#).

Sites intéressants

- [Wikipédia](#) : HTML3
- [Wikipédia](#) : CSS3
- [Validateur W3C](#) : HTML
- [Validateur W3C](#) : CSS

Licence

Ce contenu peut être réutilisé, republié et modifié selon les conditions de la licence [Creative Commons CC BY-NC-SA 3.0](#) ; auteurs : [Étienne André](#), [Laure Petrucci](#).



Le même exemple avec CSS

Présentation de l'IUT de Villetaneuse

Menu	Le BUT R&T
Le BUT R&T	<p>Le BUT du département réseaux et télécommunications de l'IUT de Villetaneuse permet d'acquérir plusieurs compétences :</p> <ol style="list-style-type: none"> 1. Administrer et sécuriser les réseaux et l'Internet, 2. Connecter les entreprises et les usagers, et 3. Créer des outils et applications informatiques pour les R&T. <p>Deux parcours sont proposés :</p> <ul style="list-style-type: none"> • <i>Cyber</i> : Cybersécurité • <i>ROM</i> : Réseaux Opérateurs et Multimédia
Le BUT en apprentissage	
La licence professionnelle	
Autres départements	
	La licence professionnelle ASSR
Description	<p>La licence professionnelle ASSR du département réseaux et télécommunications de l'IUT de Villetaneuse permet d'acquérir plusieurs compétences en termes de :</p> <ul style="list-style-type: none"> • réseaux, • sécurité et techniques d'attaques, • administration de postes sous Linux ou Windows, • communication et expression.
Deux versions	
En formation initiale	<p>La licence professionnelle peut être intégrée après un DUT réseaux et télécommunications, un BTS, ou d'autres formations similaires.</p>
En alternance	<p>Le contenu de la licence en alternance est le même que la licence en formation initiale, seules les modalités diffèrent. Dans la version en alternance, les étudiants ont le statut d'<i>apprentis</i>, et passent la moitié de leur temps en entreprise.</p> <p style="text-align: right; font-size: small;"><i>Pour en savoir plus, consultez le site Internet de la licence professionnelle ASSR.</i></p>
<p>SITES INTÉRESSANTS WIKIPÉDIA : HTML5 WIKIPÉDIA : CSS3 VALIDATEUR W3C : HTML VALIDATEUR W3C : CSS</p> <p style="font-size: x-small;">CE CONTENU PEUT ÊTRE RÉUTILISÉ, RÉPUBLIÉ ET MODIFIÉ SELON LES CONDITIONS DE LA LICENCE CREATIVE COMMONS CC BY NC SA 3.0 ; AUTEURS : ÉTHIENNE ANDRÉ, LAURIE PETRUCCI</p>	
	

Code HTML **identique**
 Simple ajout d'une feuille de style

Plan: Sites Web adaptatifs

- 1 Architecture client-serveur
- 2 HTML
- 3 CSS
- 4 Sites Web adaptatifs**

Problématique

La consultation d'un site Web peut se faire depuis des appareils très divers :

- Ordinateur de bureau à écran 34 pouces
- Ordinateur portable
- Netbook
- Tablette
- Téléphone intelligent (ou non)
- Télévision
- Navigateur pour aveugles

Problème

Comment assurer un affichage optimal pour tous ?

Solution : les sites Web adaptatifs

Conception de sites Web adaptatifs

La **conception de sites Web adaptatifs** (*responsive Web design*) regroupe différents principes et technologies ayant pour but de faciliter la consultation d'un site Web de façon optimale quel que soit l'appareil utilisé.



Principe technique

La conception de sites Web adaptatifs repose sur l'utilisation de la règle `@media` en CSS.

Utilisation :

- Import de différentes feuilles de style en fonction du navigateur (taille de l'écran, type de système, etc.)
- Redimensionnement de blocs ou d'images
- Personnalisation de parties de la feuille de style

Utilisation de @media (1/2)

Exemple d'import conditionnel de feuilles de style dans un fichier HTML :

```
<link rel="stylesheet"
      media="only screen and min-device-width:480px"
      href="bureau.css" type="text/css" />
<link rel="stylesheet"
      media="handheld, (max-device-width:480px)"
      href="smartphone.css" type="text/css" />
```

- Si appareil de type **screen** et taille de l'écran supérieure à **480 px** : utilisation de **bureau.css**
- Si appareil de type **handheld** ou taille de l'écran inférieure à **480 px** : utilisation de **smartphone.css**

Utilisation de @media (2/2)

Conditions à l'intérieur d'une feuille de style :

```
header {  
  width: 600px;  
  background-color: white;  
}  
@media only screen and (max-device-width:480px) {  
  header {  
    width: 100%;  
    background-color: #D0D0D0;  
  }  
}
```

Syntaxe de @media

Utilisation d'expressions booléennes :

- **and** : « et »
- **not** : « non »
- **,** : « ou »
- expressions imbriquées

Exemple :

```
@media screen and (max-device-width:480px) and not braille  
and not projection
```

Syntaxe : types de media

Type	Description
braille	Pour aveugle
embossed	Imprimantes en braille
handheld	Tenu à la main (ex : téléphone intelligent)
print	Impression
projection	Projecteurs
screen	Écran
speech	Synthèse vocale
tty	Terminal à chasse fixe
tv	Téléviseur
all	Tous

Syntaxe : types de fonctionnalités

Propriété	Valeur	Description
<code>color</code>	Entier	Nombre de couleurs (bits)
<code>color-index</code>	Entier	Nombre d'entrées dans la table des couleurs
<code>device-aspect-ratio</code>	Entier/Entier	Aspect ratio
<code>device-height</code>	Entier	Hauteur (appareil)
<code>device-width</code>	Entier	Largeur (appareil)
<code>grid</code>	Booléen	Vrai pour un appareil à grille
<code>height</code>	Entier	Hauteur (surface)
<code>max-device-width</code>	Entier	Largeur (appareil)
<code>monochrome</code>	Entier	Nombre de bits (monochrome)
<code>orientation</code>	<code>landscape</code> / <code>portrait</code>	Orientation de l'écran
<code>resolution</code>	Résolution	Résolution
<code>scan</code>	<code>progressive</code> / <code>interlaced</code>	Pour télévisions
<code>width</code>	Entier	Longueur (surface)

Bonnes pratiques

Quelques idées à prendre en considération :

- Menus non affichés par défaut sur terminaux de petite taille (utilisation de Javascript, ou de pages séparées)
- Éviter l'utilisation du zoom ou du défilement, peu pratique pour la navigation
- Privilégier les tailles **relatives**, en % ou **em**, plutôt qu'absolues (**pt**, **px**)
 - ▶ **1em** = taille de la police par défaut du navigateur
- Privilégier des images **vectérielles** (formats **.svg**, **.pdf**)
 - ▶ Sinon, aspect « pixelisé » en cas d'agrandissement (formats **.gif**, **.jpg**, **.png**)

Outils d'inspection de pages Web

Présentation de l'IUT de Villetaneuse

Le BUT R&T	
Menu	
Le BUT R&T	
Le BUT en apprentissage	
La licence professionnelle	<p>Le BUT du département réseaux et télécommunications de l'IUT de Villetaneuse permet d'acquérir plusieurs compétences :</p> <ol style="list-style-type: none"> 1. Administrer et sécuriser les réseaux et l'Internet, 2. Connecter les entreprises et les usagers, et 3. Créer des outils et applications informatiques pour les R.T.
Autres départements	<p>Deux parcours sont proposés :</p> <ul style="list-style-type: none"> • Cyber : Cybersécurité • ROM : Réseaux Opérateurs et Multimédia
La licence professionnelle ASSR	
Description	
	<p>La licence professionnelle ASSR du département réseaux et télécommunications de l'IUT de Villetaneuse permet d'acquérir plusieurs compétences en termes de :</p> <ul style="list-style-type: none"> • réseaux, • sécurité et techniques d'attaques, • administration de postes sous Linux ou Windows, • communication et expression.

The screenshot shows the developer tools interface. On the left, the HTML tree is expanded to show the `body` element. The right pane displays the CSS styles for the selected element, including the default browser styles for `body` and the `marker` pseudo-element. The `body` styles include `background-color: #e8f5e9`, `color: black`, and `font-family: serif`. The `marker` styles include `font-variant-numeric: tabular-nums` and `text-transform: none`. The bottom right pane shows the 'Modèle de boîte' (Box Model) with dimensions of approximately 1618.67 x 858.33 pixels.