

Technologies d'Internet

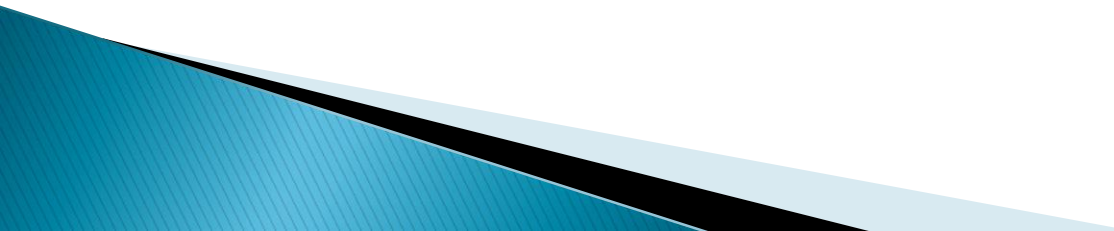
Frédéric LAGRANGE

frederic.lagrange@2la-software.com




ENSIM
École d'ingénieurs
Le Mans Université

Organisation

- ▶ Objectifs visés
 - ▶ Evaluation
 - ▶ Généralités
 - ▶ Architectures
 - ▶ Technologies
 - HTML 5
 - CSS
 - JavaScript
 - PHP
 - ▶ CMS (Content Management System)
- 

Objectifs visés

- ▶ Comprendre ce qu'est une application web
 - ▶ Savoir décrire l'architecture et l'organisation d'une application web
 - ▶ Connaître les technologies de développement web
 - ▶ Maîtriser la programmation de sites web complexes
- 

Déroulement de cette option

- ▶ 6 séances de cours de 1h15 (7h30)
- ▶ 6 séances de TD de 1h15 (7h30)
- ▶ 4 séances de TP de 3h (12h)
 - A la fin de la dernière séance de TP : présentation et remise du projet

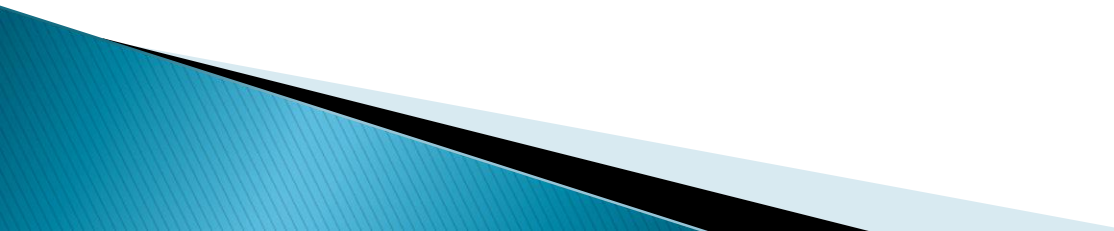
Espace cours pour ce module sur UMTICE

<http://umtice.univ-lemans.fr/course/view.php?id=3525>

Clé : HCJP5

- ▶ Supports de cours/TD en ligne, sujets de TP
- ▶ Liens vers des outils utiles à la conception de sites
- ▶ Liens vers des informations complémentaires

Evaluation

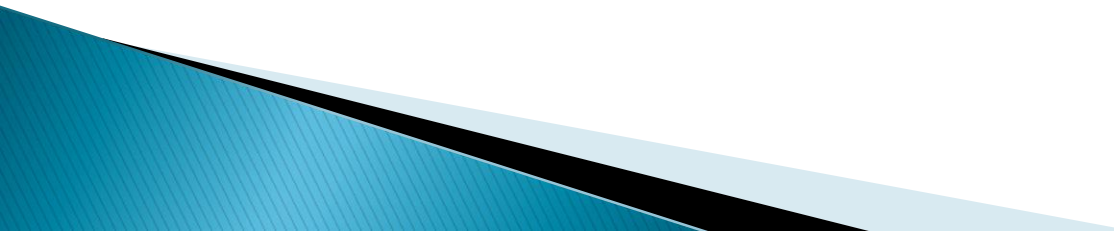
- ▶ Une note de projet
 - ▶ Une note de rapport de projet
 - ▶ Projet individuel
- 

Généralités



Vocabulaire

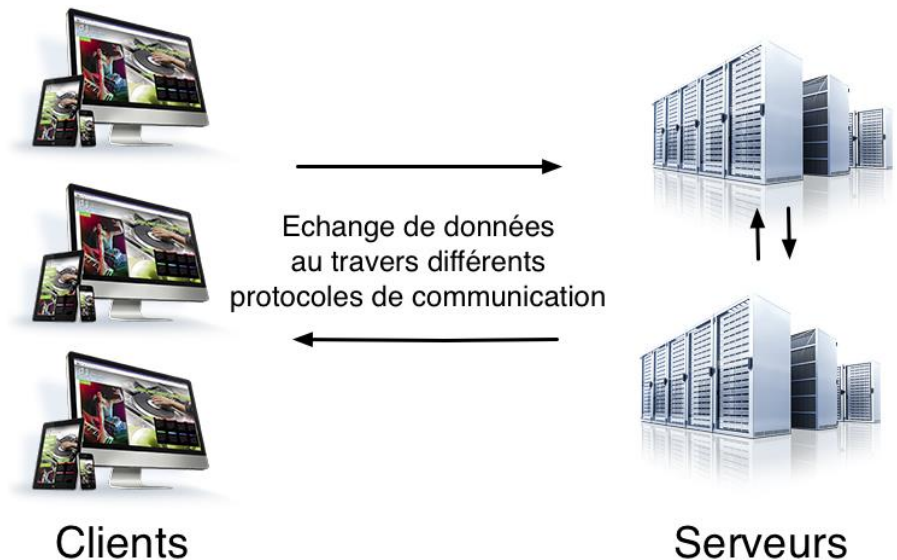
- ▶ **Que pensez-vous des termes suivants ?**
 - Internet
 - Toile
 - Réseau

 - ▶ **Pourquoi mettre des ordinateurs en réseau ?**
 - Echanger des informations
 - Sauvegarder des données
 - Travailler à plusieurs
 - ...
- 

Vocabulaire

Internet : réseau informatique mondial donnant accès à des **services** (courrier, world wide web, ...) et permettant à ses utilisateurs d'**échanger des données** et de **communiquer**.

- l'accès se fait via un matériel : ordinateur, smartphone, tablette, ...
- avec un moyen de communication : filaire ou sans fil (wifi, bluetooth)
- grâce à des protocoles de communication : http, ftp, ...



Historique d'Internet 1 / 2

► 1962 : les concepts

Joseph Carl Robnett Licklider du MIT (Massachusetts Institute of Technology) décrit les premières interactions possibles avec un réseau d'ordinateurs

► 1965 : Premier test

Première connexion longue distance (entre le Massachusetts et la Californie)

La DARPA (Defense Advanced Research Projects Agency) lance la conception d'ARPANET (Advanced Research Projects Agency Network), premier réseau à transfert de paquets.

► 1970 : Première utilisation

Déploiement d'ARPANET grâce au protocole de communication NCP

Historique d'Internet 2/2

- ▶ 1972 : Première application importante

Le courrier électronique

- ▶ 1980 : ARPANET accessible

ARPANET devient accessible à grande échelle

- ▶ 1990 : Internet d'aujourd'hui

Le web, pages en HTML avec textes, images liens, etc. via des URL et le protocole HTTP

- ▶ 2015 : Etat du marché

Plus d'un milliard de sites en ligne pour plus de trois milliards d'internautes. Ces chiffres croissent de manière importante en fil des ans

Protocoles de communication

HTTP : HyperText Transfer Protocol :

- Protocole pour accéder à un serveur contenant des données
- Clients HTTP : navigateurs Web ou robots d'indexation

HTTPS : HyperText Transfer Protocol Secure

- Protocole de transfert sécurisé disposant d'une couche de chiffrement (SSL, TLS)
- Vérifier l'identité du visiteur avec un certificat d'authentification
- Commerce électronique, Banques, etc.

Moteurs de recherches

Robots d'indexation pour les moteurs de recherche

- Parcourt des liens à partir d'une racine pivot
- Indexation des pages selon des mots-clés
- Robot = Crawler = Bot = ...
- GoogleBot (Google)

Mettre des aides dans nos pages pour qu'elles soient indexées/référencées, si on le souhaite : métadonnées.

Protocoles de communication

- ▶ **HTTP(S)** : HyperText Transfer Protocol (Secure)
- ▶ **FTP** : File Transfer Protocol
 - Protocole pour échanger des fichiers sur le réseau
- ▶ **TCP/IP** : Transmission Control protocol / Internet Protocol
- ▶ **IRC** : Internet Relay Chat
 - Protocole de communication instantanée et textuelle sur Internet
- ▶ **SMTP, POP3, IMAP**
 - Protocoles de messageries : permettant de récupérer les messages sur les serveurs

Navigateurs web

- ▶ Internet Explorer & Edge (Windows), Safari (Mac OS), Firefox, Google Chrome, Android Browser, etc.
- ▶ Reposent sur des **moteurs de rendu HTML** : traiter une ressource respectant les **standards du Web** et afficher le résultat sur une interface utilisateur :
 - **Gecko** : Mozilla Firefox, Netscape (depuis version 6)
 - **KHTML** : Konqueror
 - **WebKCore** : Safari
 - **Blink** : Google Chrome, Chromium, Microsoft Edge, Opera (depuis version 15)
 - **Trident** : Internet Explorer, Netscape (depuis version 8)

Applications web

- ▶ Des exemples ?
- ▶ Quelles sont les particularités ?
- ▶ Différence entre Internet et Intranet ?

Architectures d'applications



- ❑ Client
- ❑ Serveur
- ❑ Client / Serveur

Cas d'un programme client et Serveur



Envoi de la **requête**
(demande d'une page Web)



Envoi de la réponse
(la page Web demandée)



Serveurs

Traitement de la requête

Clients

Affichage de la réponse
via le navigateur Web

Cas d'un programme client et Serveur



Envoi de la **requête**
(demande d'une page Web)



PHP
SQL

Envoi de la réponse
(la page Web demandée)



HTML / CSS



Serveurs

Traitement de la requête

Clients

Affichage de la réponse
via le navigateur Web

La notion de serveur WEB

- ▶ Un logiciel ?
- ▶ Un ordinateur ?

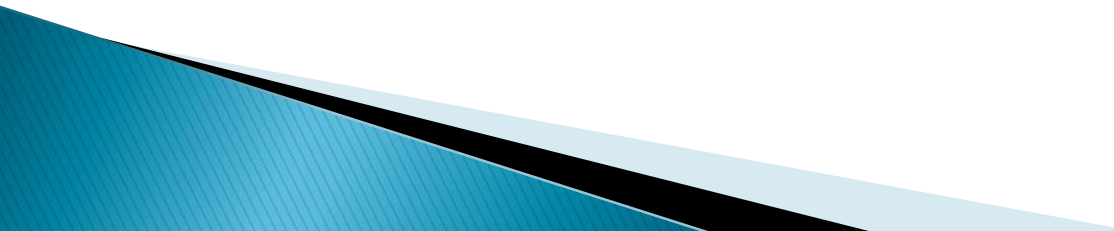
HTML 5
= évolution (HTML 4.01 &
XHTML 1.0)

» [http://www.w3.org/TR/2012/
CR-html5-20121217/](http://www.w3.org/TR/2012/CR-html5-20121217/)

HTML, XHTML et HTML5

- ▶ **HTML** : HyperText Markup Language
- ▶ **Langage à balises**
- ▶ Différentes versions/recommandations d'un même langage, le HTML
- ▶ Consortium W3C
- ▶ **XHTML 2.0** : un des standards, programmation rigoureuse, pour prendre de bonnes habitudes
- ▶ **HTML5.1** : dernier standard : améliorer l'interopérabilité et les performances des 'user agents'

[<https://www.w3.org/standards/history/html51> – Dernières recommandations du 15 septembre 2016]



Concepts

► Séparation forme et contenu

- HTML pour mettre le contenu
- CSS pour la mise en forme

► Langage à balises

- Exemple : `<p> </p>` – `<html></html> `
- Attributs aux balises

``

► Différentes version du langage (historique)

- HTML 4 en 1999
- Depuis 2006 coopération W3C et WHATWG sur HTML 5
- Depuis 2010 Apple supporte fortement HTML5

Concepts HTML 5

► Des “règles” ont été établies

- Toutes les nouvelles fonctionnalités seront basées sur HTML, CSS, DOM, et JavaScript
- Réduire au maximum la nécessité de plugins externes comme Flash
- Meilleure gestion des erreurs
- Essayer de remplacer par des tags ce qui se fait en masse avec du script (menus, etc)
- HTML5 doit supporter plusieurs devices (mobiles...)
- Le processus de développement doit être ouvert au public

XHTML 1.0 strict

Bonnes habitudes à respecter :

- ▶ Toute balise ouvrante doit être **fermée** : `<hr />` `
`
- ▶ Balises et attributs doivent être écrits en **minuscules**
- ▶ Les valeurs des attributs doivent être entre **guillemets** ou **apostrophes**
- ▶ Chaque **attribut** doit avoir une **valeur**
- ▶ Chaque élément doit être **imbriqué correctement**
- ▶ Améliorer l'**accessibilité** (pour lecture par un appareil de synthèse vocal) : attribut "alt" d'une image

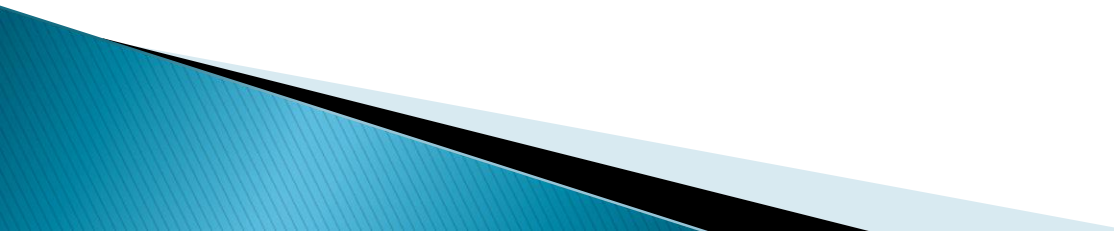
Ressources pour démos, tutos et exemples

- ▶ Page de ressources HTML5 régulièrement mise à jour :
 - <http://tinyurl.com/7px5gpe>
- ▶ Autres démonstrations utilisées
 - <http://www.w3.org/2010/Talks/0430-www2010-plh/>
 - <http://www.htmlfivewow.com>

Un document minimal

```
<!doctype html>
<html lang="fr">
<head>
<meta charset="utf-8">
<title>Titre de la page</title>
<link rel="stylesheet"
      href="style.css">
<script src="script.js"></script>
</head>
<body>
... <!-- Le reste du contenu --> ...
</body>
</html>
```

Encodage : UTF-8 en HTML5 !

- ▶ ISO-8859-1 : permet d'enregistrer presque tous les caractères du français
 - ▶ ISO-8859-15 : une variation de l'ISO-8859-1, qui rajoute le symbole « euro » et le « l'e dans l'o »
 - ▶ UTF-8 : permet théoriquement d'encoder toutes les langues, du français au japonais en passant par l'arabe
- 

Encodage : quelques exemples...

Quelques correspondances entre ISO-8859-1 et UTF 8 :

é = Ã© = é

è = Ã¨ = è

à = Ãà = à

ï = Ãï = ï

ô = Ã´ = ô

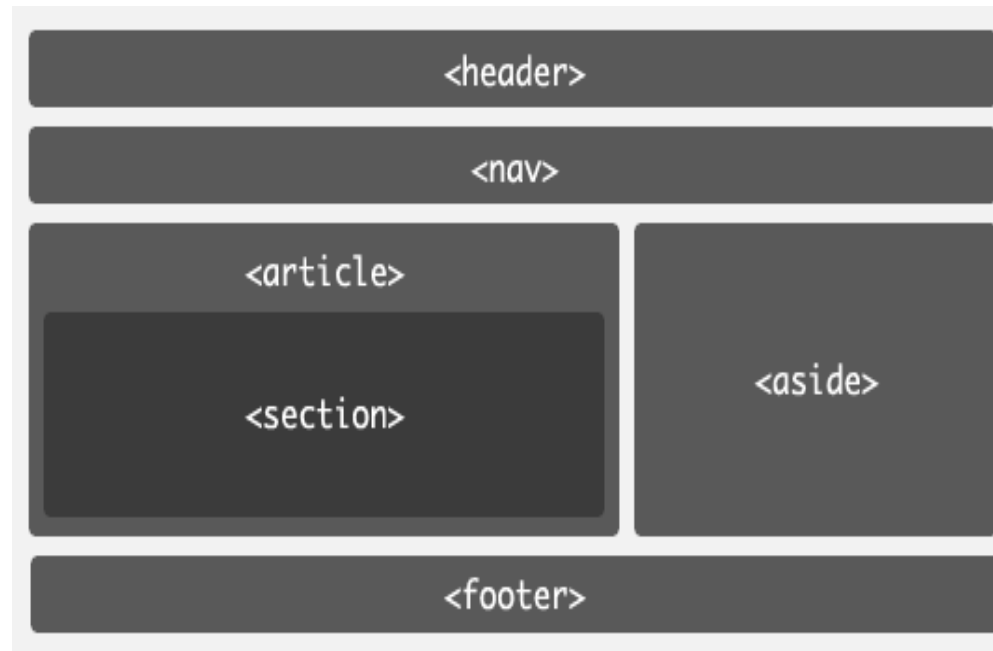
ç = Ãç = ç

ê = Ãê = ê

ù = Ãù = ù

Le Corps du document

```
<body>
<header>...</header>
<nav>...</nav>
<article>
  <section> ...
  </section>
</article>
<aside>... </aside>
<footer>...</footer>
</body>
```



Vérification de la syntaxe

- ▶ Navigateur != compilateur => pas d'informations précises en cas d'erreur
- ▶ Validateur W3C : <https://validator.w3.org/> permet de valider la syntaxe

Balises méta : Métadonnées

Informations disponibles pour les

- ▶ navigateurs web
- ▶ moteurs de recherche
- ▶ outils d'indexations

Exemples de métadonnées

```
<meta charset="utf-8"> <!-- en HTML5 -->
```

```
<meta name="description" content="Une description courte de ma page" />
```

```
<meta name="keywords" content="cours, HTML, Licence, SPI" /> <!-- deprecated -->
```

```
<meta name="robots" content="noindex" /> <!-- restreindre l'indexation -->
```

```
<!-- noindex : pas d'indexation de la page -->
```

Des balises à connaître

- La balise de paragraphe : `<p>`
- le saut de ligne (!= de paragraphe) `
`
- `...` : pour mettre en gras ("bold" en anglais)
- ` ... ` : pour mettre le texte en évidence
- `<i>...</i>` : pour mettre en italique
- `^{...}` : pour écrire du texte en exposant
- `_{...}` : pour écrire du texte en indice

Exemple : niveaux de titres

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title> Structure de base </title>
</head>
<body>
  <h1> Exemple de page Web</h1>
  <p> Nous allons nous concentrer sur les balises à l'intérieur de la balise body </p>
  <p><b> Ceci est du texte en gras </b>
    ce n'est pas parce que je vais à la ligne dans mon éditeur que
    cela va à la ligne dans mon navigateur. Il faut écrire ceci <br />
    <!-- Commentaires : break -->
    pour aller dans la ligne dans le navigateur.
  </p>
  <h2> Ceci est un titre de niveau 2 </h2>
  <h3> Ceci est un titre de niveau 3 </h3>
  <h6> Ceci est un titre de niveau 3 </h6>
  <p> Les titres vont du niveau 1 au niveau 6,
    le niveau 6 étant le plus petit.
  </p>
  <!-- Commentaires : horizontal rule -->
  <hr />
  <p> Je peux aussi tracer un trait. </p>
</body>
</html>
```

Exemple de page Web

Nous allons nous concentrer sur les balises à l'intérieur de la balise body

Ceci est du texte en gras ce n'est pas parce que je vais à la ligne dans mon éditeur que cela va à la ligne dans mon navigateur. Il faut écrire ceci pour aller dans la ligne dans le navigateur.

Ceci est un titre de niveau 2

Ceci est un titre de niveau 3

Ceci est un titre de niveau 3

Les titres vont du niveau 1 au niveau 6, le niveau 6 étant le plus petit.

Je peux aussi tracer un trait.

Les Tableaux

1. On déclare le tableau : `<table>`
2. On déclare la première ligne : `<tr>`
3. On déclare, une par une, les cellules de la première ligne :
 - en ouvrant chaque cellule avec la balise `<td>`
 - en fermant chaque cellule avec la balise `</td>`
4. On répète l'opération " 3 " autant de fois qu'il y a de cellules dans la ligne.
5. On conclut la ligne : `</tr>`
6. On répète les opérations " 2 à 4 " autant de fois qu'il y a de lignes dans le tableau.
7. On conclut le tableau : `</table>`

Exemple : Tableaux

```
<table border="1">
  <!-- Titre général au dessus du tableau -->
  <caption>Titre du tableau</caption>
  <thead><!-- En-tête du tableau -->
    <tr>
      <th>Titre Col 1</th>
      <th>Titre Col 2</th>
    </tr>
  </thead>
  <tfoot><!-- "pied de page" du tableau -->
    <tr>
      <td>Valeur Footer 1</td><td>Valeur Footer 2</td>
    </tr>
    <tr>
      <td>Valeur Footer 3</td><td>Valeur Footer 4</td>
    </tr>
  </tfoot>
  <tbody><!-- les informations principales du tableau -->
    <tr>
      <td>Valeur 1</td><td>Valeur 2</td>
    </tr>
    <tr>
      <td>Valeur 1</td><td>Valeur 2</td>
    </tr>
  </tbody>
</table>
```

Titre du tableau

Titre Col 1	Titre Col 2
Valeur 1	Valeur 2
Valeur 1	Valeur 2
Valeur Footer 1	Valeur Footer 2
Valeur Footer 3	Valeur Footer 4

Les Listes et Images

- `ul` – `ol` : unordered list ou ordered list
- `li` : list-item
- commentaires entre `<!--` et `-->`
- Imbrications possibles de listes

- Image : ``
- Lien hypertexte : `texte hyperactif`

Exemple : Listes

```
<!-- Commentaire : unordered list -->
<ul>
  <!-- Commentaire : list items -->
  <li>Listes à puces </li>
  <li>avec autant d'items voulus </li>
</ul>
<!-- Commentaire : ordered list -->
<ol>
  <li>Listes numérotées </li>
  <li>toujours avec des items </li>
</ol>
```

- Listes à puces
 - avec autant d'items voulus
-
1. Listes numérotées
 2. toujours avec des items

Nouvelles Balises HTML5

► Sémantique

Balise	Description
<code><article></code>	Définit un contenu autonome et indépendant.
<code><aside></code>	Définit un contenu « mis à part » dans un contenu du document
<code><bdi></code>	Isolé une partie du texte qui peut être formaté dans une direction différente du texte dont il est exclu.
<code><command></code>	Définit un bouton de commande que l'utilisateur peut appeler
<code><details></code>	Définit les détails supplémentaires que l'utilisateur peut afficher ou masquer
<code><summary></code>	Définit un élément visible pour un élément <code><details></code>
<code><figure></code>	Spécifie un contenu autonome, comme des illustrations, des diagrammes, des photos, des listes de codes, etc
<code><figcaption></code>	Définit une légende pour un élément <code><figure></code>
<code><footer></code>	Définit le pied de page d'un document ou d'un article
<code><header></code>	Définit une introduction, une en-tête ou un groupe d'éléments de navigation pour un document
<code><hgroup></code>	Regroupe un ensemble d'éléments à <code><h1></code> <code><h6></code> quand une rubrique a plusieurs niveaux
<code><mark></code>	Passé un coup de marker sur vote texte – mise en évidence
<code><meter></code>	Définit une mesure scalaire dans une gamme connue (une jauge)
<code><nav></code>	Définit un menu, des liens de navigation
<code><progress></code>	Représente l'état d'avancement d'une tâche
<code><ruby></code>	Définit une annotation ruby (pour la typographie d'Asie orientale)
<code><rt></code>	Définit une explication / prononciation des caractères (pour la typographie d'Asie orientale)
<code><rp></code>	Définit un texte d'erreur pour les navigateurs ne supportant pas la annotations Ruby
<code><section></code>	Définit une section dans un document
<code><time></code>	Définit une date / un horaire
<code><wbr></code>	Définit un saut de ligne possible

Nouvelles Balises HTML5

► Élément de formulaire

Balise	Description
<u><datalist></u>	Spécifie une liste de pré-définis pour les options des contrôles d'entrée
<u><keygen></u>	Définit un champ générateur de clés (pour les formulaires)
<u><output></u>	Définit le résultat d'une opération

► Multimédia

Balise	Description
<u><audio></u>	Définit un contenu audio – En savoir plus sur la lecture d'un fichier audio
<u><video></u>	Définit une vidéo ou un film
<u><source></u>	Définit de multiples ressources pour les médias <video> et <audio>
<u><embed></u>	Définit un conteneur pour une application externe ou un contenu interactif (un plug-in)
<u><track></u>	Définit des pistes de texte pour les médias <video> et <audio>
<u><canvas></u>	Utilisé pour dessiner des graphiques, des dessins, à la volée, via des scripts (habituellement en JavaScript)

Nouvelles Balises HTML5

► Éléments supprimés ou « deprecated »

<acronym>

<applet>

<basefont>

<big>

<center>

<dir>

<frame>

<frameset>

<noframes>

<strike>

<tt>

<u>

Exemple : gestion native vidéo

code

```
<video controls poster='poster320.jpg' width='320' height='180'>  
<source src='video.ogg' type='video/ogg' ; codecs="theora, vorbis" />  
<source src='video.mp4' type='video/mp4' ; codecs="avc1.42E01E, mp4a.40.2" />  
<source src="video.webm" type='video/webm; codecs="vp8.0, vorbis" />  
</video>
```

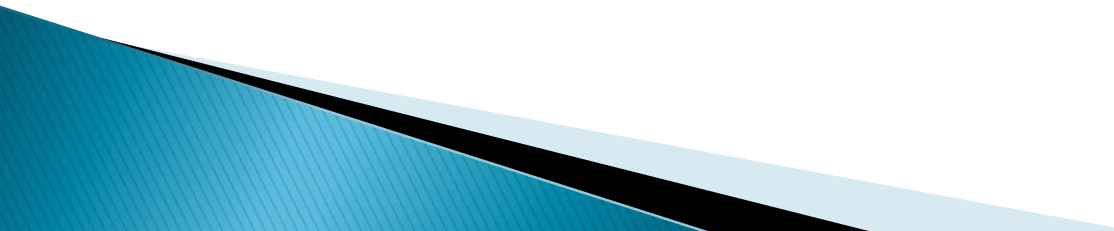
Les attributs pour la balise <video> :

- ▶ **height – width** : taille
- ▶ **controls** : contrôle de lecture (boutons de navigation, volume, etc, selon les navigateurs)
- ▶ **preload= "auto"** : débiter le téléchargement de la vidéo au chargement de la page
- ▶ **autoplay** : lancer la lecture automatiquement
- ▶ **poster="image.jpg"** : image par défaut
- ▶ **loop** : lecture en boucle
- ▶ **audio= "mute"** : désactiver le son de la vidéo

CSS

»» Cascading Style Sheets

Intérêt des CSS

- ▶ Gestion séparée du contenu et de la présentation
 - ▶ Meilleure portabilité du contenu
 - ▶ Portabilité du style : harmonisation de la charte graphique
 - ▶ Gestion du code simplifiée avec plus de lisibilité en cas de modification et de mise à jour
- 

Associations de CSS externes

- ▶ Inclure dans la balise <head> de la page html :

```
<link rel="stylesheet" href="style1.css" />
```

...

```
<link rel="stylesheet" href="style10.css" />
```

- ▶ Feuille css « style1.css » :

```
h1 {\nfont-family: Arial, Helvetica,sans-serif;\nfont-size: 14px; /* changer la taille de la police */\ncolor: #00ff00; /* changer la couleur de la police background-color :#000000; /* changer\nla couleur de fond\n}
```

- ▶ Validation du code css :

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

Associations de CSS internes

- Inclure dans la balise <head> de la page html :

```
<head>
<style type="text/css">
<!--
h1{
font-family:Arial, Helvetica, sans-serif ; /* style de la police */
font-size:14 px ; /* taille de la police */
color:#00ff00 ; /* couleur de la police */
background-color:#000000 ; /* couleur du fond */
}
-->
</style>
</head>
```

- **Inconvénients : maintenance et portabilité**

Déclarations dans les balises

- ▶ **Fonctionne mais très mauvaise pratique :**

```
<h2 style="text-align:center; color=red;">
```

le titre de la page

```
</h2>
```

- ▶ **Inconvénients :**

- Lisibilité
- Modularité
- Maintenance
- Portabilité

Le vocabulaire

► CSS

- Des sélecteurs
- Des déclarations, blocs de déclarations
- Des propriétés
- Des valeurs
- Des règles
- Feuille de style



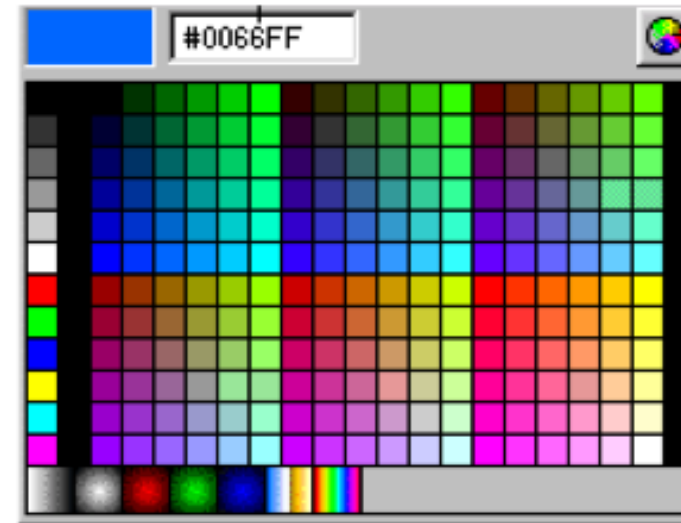
Couleurs et fond

► Le codage de la couleur

- Mélange de trois couleurs : RVB, codé en hexadécimal
- #000000 pour le noir, #FFFFFF pour le blanc et toutes les déclinaisons possibles
- quelques raccourcis : black, white, red, blue ...

► Exemples

#FF0000 Rouge
#00FF00 Vert
#0000FF Bleu
#ED00FF Turquoise
#000000 Noir
#FFFFFF Blanc



Couleurs et fond

► Les propriétés

- `color` pour définir la couleur du texte
- `background-color` pour définir la couleur de fond du document
- `background-image` pour spécifier une image de fond dans le document

Mise en forme du Texte

- ▶ **font-family** : police de caractères
valeur : Verdana, Arial, Helvetica, Geneva...
 - ▶ **font-size** : taille de la police
valeur : une taille par exemple 12px
 - ▶ **font-style** : style de la police
valeur : normal, italic...
 - ▶ **font-weight** : épaisseur de la police
valeur : normal, bold, bolder, lighter...
 - ▶ **text-decoration** : soulignement du texte
valeur : none, underline, overline, line-through...
 - ▶ **text-align** : alignement du texte
valeur : left, right, center, justify
 - ▶ **letter-spacing** : distance entre les lettres
valeur : une distance, par exemple 1px
- 

Mise en forme des listes

- ▶ **list-style-type** : type de puces à appliquer dans la liste

valeur : disc, circle, square, decimal, none

- ▶ **list-style-image** : spécification d'une image à utiliser comme puce à la place des puces habituelles utilisées dans des listes

valeur : url(image.gif)

Mise en forme des tableaux

► Les bordures de façon générale

- **border-color** : couleur de la bordure
valeur : une couleur
- **border-style** : forme de la bordure
valeur : none, hidden, dotted, dashed, solid, double groove, ridge, inset, outset
- **border-width** : épaisseur de la bordure
valeur : épaisseur ou thin, medium, thick

► Les propriétés spécifiques aux tables

- **padding** : espace entre la bordure et le contenu des cellules
valeur : la largeur
- **border-collapse** : fusionner la bordure du tour du tableau et la bordure des cases
valeur : *collapse* pour fusionner et *separate* pour séparer les bordures

Définition d'une classe CSS

- ▶ Permet d'appliquer un style précis pour une balise particulière
- ▶ Syntaxe pour déclarer une classe dans le CSS :

```
balise.NomClasse  
{  
  propriété1 : valeur1;  
  propriété2 : valeur2;  
  ...  
  propriétéN : valeurN ;  
}
```

- ▶ Appel dans le HTML :

```
<balise class="NomClasse">...</balise>
```

Exemple d'une classe CSS

► HTML :

```
<h1> un titre </h1>  
<table>  
  <tr>  
    <td class="grise"> une cellule grise </td>  
    <td class="bleue"> une cellule bleue </td>  
    <td class="grise"> une cellule grise </td>  
  </tr>  
</table>  
</div>
```

un titre

une cellule grise

une cellule bleue

une cellule grise

► CSS :

```
td.grise {background-color : grey ;}  
td.bleue {background-color : blue ;}
```

Difficulté : tenir compte de tous les navigateurs

En CSS on arrive à des choses comme :

- ▶ `border:1px solid #888; /* toutes les versions */`
- ▶ `border-radius:10px; /* Explorer 9, Firefox 4, Opera 10.5 */`
- ▶ `-moz-border-radius:10px; /* Firefox 3.5 */`
- ▶ `box-shadow:5px 5px #888; /* Explorer 9, Firefox 4, Opera 10.5 */`
- ▶ `-moz-box-shadow:5px 5px #888; /* Pour Firefox 3.5 */`
- ▶ `-webkit-box-shadow:5px 5px #888; /* Safari 3.0, Chrome 1.0 */`

La première ligne assure un service minimum tandis que les suivantes s'adressent à différents navigateurs qui comprendront s'ils sont concernés