

Technologies et Développement Web

Cours 2 – Le Langage HTML

Objectifs du module

- Création de pages web avec le langage HTML
- Utiliser Aptana Studio pour appliquer les connaissances acquises

Plan du module

- La Structure
- Le Texte
- Les Listes
- Les Liens
- Les Images
- Les Tableaux
- Les Formulaires
- Organisation
- Audio et Vidéo
- Nouveautés HTML 5

Section 1 : La Structure

Exemple d'une page web

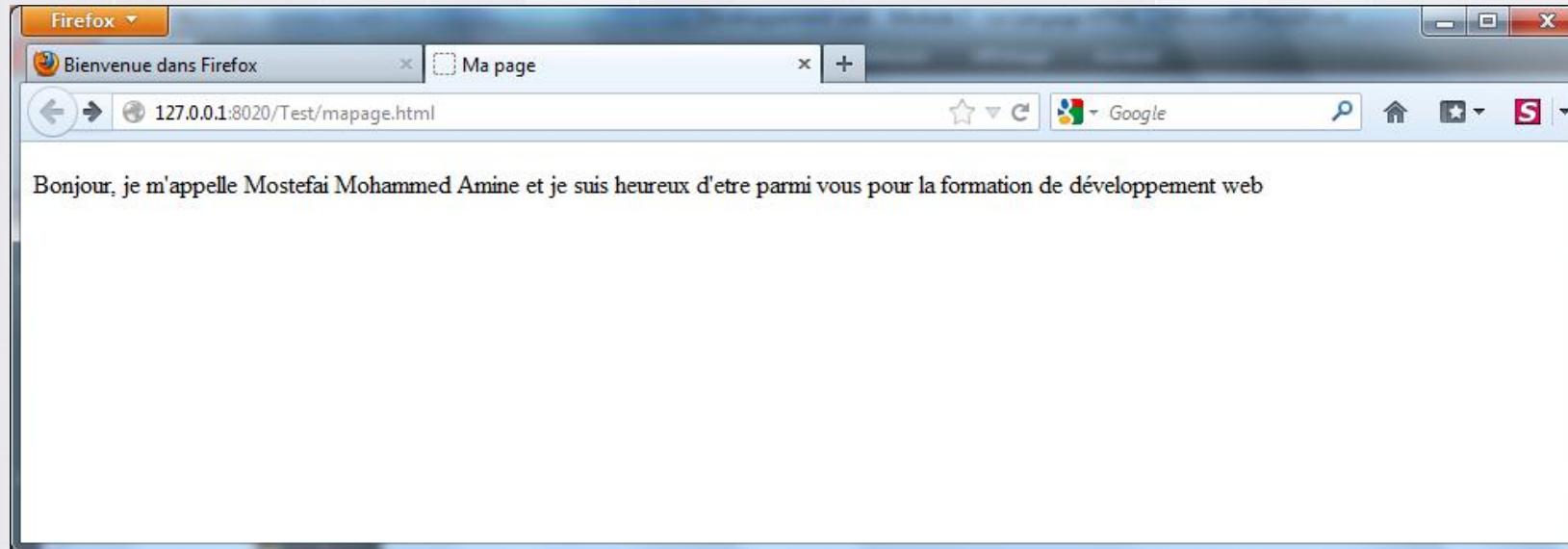
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
<head>
<title>Ma page</title>
</head>

<body>
<p>
Bonjour, je m'appelle Mostefai Mohammed Amine et je suis heureux d'etre
parmi vous pour le cours technologies et développement web
</p>
</body>
</html>
```

Exemple d'une page web

Bonjour, je m'appelle Mostefai Mohammed Amine et je suis heureux d'être parmi vous pour la formation de développement web



Un petit historique

- 1996 : HTML 4 est sorti. La plupart des balises de HTML 4 sont encore utilisées mais certaines balises orientées présentation (par exemple <center>,) sont obsolètes
- 2000 : XHTML 1.0 permet une syntaxe plus stricte de HTML et plus de conformité avec le langage XML
- 2000 : HTML 5 est sorti mais n'a pas encore été standardisé

Structure d'une page web

```
<?xml version="1.0" encoding="UTF-8"?>
```

- Indique que le document HTML est en réalité un document XML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"  
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

- Comme il y a plusieurs versions de HTML, chaque page doit inclure une balise <DOCTYPE> indiquant la version du HTML utilisée pour le document actuel

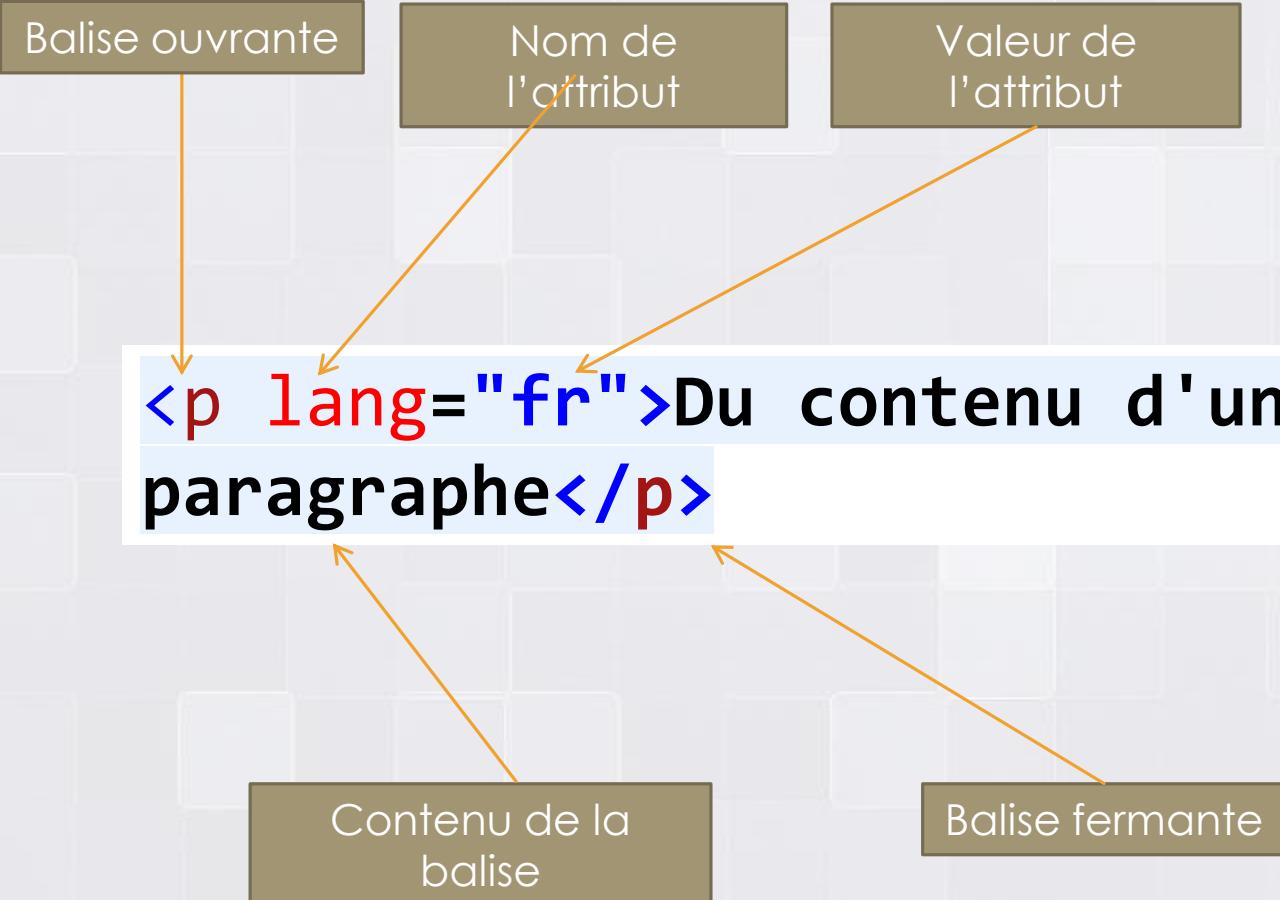
```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
```

- La balise html indique que tout ce qu'elle contient, c'est du code HTML
- **<body>**
- La balise « body » contient le contenu qui va être **affiché** dans la fenêtre du navigateur

```
<head>  
<title>Ma page</title>  
</head>
```

- La balise « head » contient des informations descriptives sur la page (par exemple, le titre)

Les balises



Les balises vides

10

Le titre d'une page

- Le titre est indiqué par la balise « title »

```
<title>Ma page</title>
```

- Le titre apparaît dans la barre de titre du navigateur

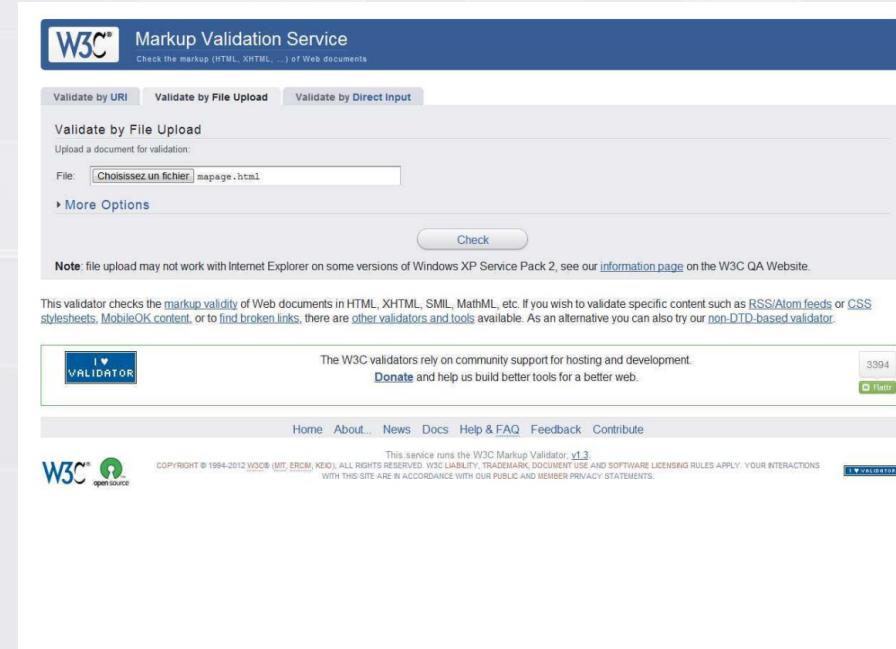
Voir comment les autres ont fait leurs pages HTML

12

- Chaque navigateur a des fonctions permettant de consulter le code source et d'inspecter un site
- Internet explorer : bouton droit + « afficher le source »
- Internet explorer : outils développeur F12
- Firefox : code source CTRL + U
- Menu développement web offre une pléthore d'outils
- Chrome : Bouton droit + « afficher le code source de la page »
- L'inspecteur est un élément puissant permettant d'analyser les éléments d'une page

Validation d'une page

- La validation inspecte les erreurs et vérifie que la page est conforme au standard (XHTML)
- La w3c met à disposition un site de validation : <http://validator.w3.org>



Exercice 1

- Créer une page conforme XHTML
- Valider la page sur le site de validation

Section 2 : Le Texte

Balises orientées texte

- HTML inclut un certain nombre de balises permettant de structurer le texte
- HTML 5 a introduit des balises dites sémantiques permettant de donner plus de sens au code HTML

Paragraphes

- Le paragraphe est défini par la balise `<p>`
- Par défaut, le paragraphe provoque un saut de ligne

`<p>`

Je ne comprenais pas pourquoi cent mille personnes téléchargeaient le programme et j'ai décidé de déchiffrer ce mystère.

Ça m'a pris trois ans pour en faire une théorie unifiée de l'expérience utilisateur".

`</p>`

`<p>`

Sans qu'il soit possible de déterminer l'importance de cette recherche dans sa fortune, il réussit plus que bien

`</p>`

Je ne comprenais pas pourquoi cent mille personnes téléchargeaient le programme et j'ai décidé de déchiffrer ce mystère. Ça m'a pris trois ans pour en faire une théorie unifiée de l'expérience utilisateur".

Sans qu'il soit possible de déterminer l'importance de cette recherche dans sa fortune, Vardi réussit plus que bien

Gras et italique

- Un texte en gras (ou qui mérite le focus) est inséré dans la balise **< b >**
- La balise *< i >* décrit un texte en italique

< p >

Je ne comprenais pas pourquoi cent mille personnes téléchargeaient le programme et j'ai décidé de déchiffrer ce mystère.

Ça m'a pris trois ans pour en faire une théorie unifiée de l'expérience utilisateur".

< /p >

< p >

Sans qu'il soit possible de déterminer l'importance de cette *< i > recherche < /i >* dans sa fortune, il réussit plus que bien

< /p >

Je ne comprenais pas pourquoi cent mille personnes téléchargeaient le programme et j'ai décidé de déchiffrer ce mystère. Ça m'a pris trois ans pour en faire une théorie unifiée de l'expérience utilisateur".

Sans qu'il soit possible de déterminer l'importance de cette *recherche* dans sa fortune, il réussit plus que bien

Indice et exposant

- Exposant : <sup>
- Indice : <sub>

Mostefai M^{ed} Amine trouve que les matchs de C₁ sont de plus haut niveau que ceux de la C₃

Mostefai M^{ed} Amine trouve que les matchs de C₁ sont de plus haut niveau que ceux de la C₃

Sauts de ligne et règles horizontales

- Un paragraphe provoque automatiquement un saut à la ligne mais si on veut forcer ce saut, on utilise la balise

- <hr> représente une règle horizontale
- Dans HTML5, <hr> représente un changement de la thématique du contenu

Mostefai M^{ed} Amine trouve que les matchs de C₁ sont de plus haut niveau que

ceux de la C₃
</p><hr/> <p>Mais le niveau de notre championnat est loin derrière !</p>

Mostefai M^{ed} Amine trouve que les matchs de C₁ sont de plus haut niveau que
ceux de la C₃

Mais le niveau de notre championnat est loin derrière !

Texte important

- La balise **indique un texte d'une grande importance, *un texte accentué***
- Dans HTML4, **et *représente un texte accentué*, dans HTML 5 il représente un texte important**

Mostefai M^{ed} Amine trouve que les matchs de C₁ sont de plus **haut niveau** que ceux de la C₃ *malheureusement* pour le spectacle

Mostefai M^{ed} Amine trouve que les matchs de C₁ sont de plus **haut niveau** que ceux de la C₃ *malheureusement* pour le spectacle

Citations

- La balise `<blockquote>` représente une citation. Le navigateur l'affiche par défaut en bloc et l'indente,
- `<q>` représente une courte citations

`<p>`

Voici une des citations les plus intriguantes sur la politique :

`<blockquote cite="Paul Valéry">`

La politique est l'art d'empêcher les gens de se mêler de ce qui les regarde`</blockquote>`

`</p>`

`<p>`Une autre citation plus drôle : `<q>`Le mariage est la principale cause du divorce`</q></p>`

Voici une des citations les plus intriguantes sur la politique :

La politique est l'art d'empêcher les gens de se mêler de ce qui les regard

Une autre citation plus drôle : "Le mariage est la principale cause du divorce"

Abréviations

- Les abréviations sont définies par la balise `<abbr>`

Le `<abbr title="Centre de Recherche en Information Scientifique et Technique">CERIST</abbr>` a été le premier provider internet en Algérie

Le CERIST a été le premier provider internet en Algérie

Référence

- La balise <cite> indique le titre d'une œuvre (livre, chanson, ...)

```
<p>Le film <cite>John Carter</cite> a nécessité un budget énorme mais était un vrai fiasco au box-office</p>
```

Le film *John Carter* a nécessité un budget énormé mais était un vrai fiasco au box-office

Définitions

- La balise `<dfn>` indique une définition

`<p>`

Une `<dfn>adresse IP</dfn>` (avec IP pour Internet Protocol) est le numéro qui identifie chaque ordinateur connecté à Internet, ou plus généralement et précisément, l'interface avec le réseau de tout matériel informatique (routeur, imprimante) connecté à un réseau informatique utilisant l'Internet Protocol`</p>`

Une *adresse IP* (avec IP pour Internet Protocol) est le numéro qui identifie chaque ordinateur connecté à Internet, ou plus généralement et précisément, l'interface avec le réseau de tout matériel informatique (routeur, imprimante) connecté à un réseau informatique utilisant l'Internet Protocol

Titres

- Les titres sont définis par la balise <hn>
- Dans HTML, il y a six niveaux de titre : de h1 à h6

<h1>Que faire pendant les longues journées du ramadhan ?</h1>

<h2>Travailler</h2>

<p>

Pour ceux qui ont la chance de ne pas avoir pris de congé.

</p>

<h2>Dormir</h2>

<p>Un choix inéluctable</p>

Que faire pendant les longues journées du ramadhan ?

Travailler

Pour ceux qui ont la chance de ne pas avoir pris de congé.

Dormir

Un choix inéluctable

Changements dans un contenu

27

- HTML permet de faire le suivi de modifications grâce aux balises ins et del
- Ins indique une insertion et del une suppression

<p>

Ma soeur est une médiocre<ins>excellente</ins> cuisinière

</p>

Ma soeur est une médiocre excellente cuisinière

Changements dans un contenu

- La balise <s> indique un texte qui n'est plus correct

```
<p>  
<s>Ma voiture est blanche</s>  
</p>  
<p>  
Ma voiture est grise  
</p>
```

Ma voiture est blanche

Ma voiture est grise

Changements dans la direction (HTML5)

29

- La balise `<bdi>` indique un changement dans la direction de texte (par exemple arabe / français)

```
p>Mon nom en arabe est <bdi>/> أمين<bdi> et c'est un joli prénom</p>
```

Mon nom en arabe est أمين et c'est un joli prénom

Emphase (HTML5)

- La balise **mark** indique une emphase sur son contenu

```
<p>Le développement web nécessite de la <mark>passion</mark> et de la  
patience</p>
```

Le développement web nécessite de la passion et de la patience

Texte pré-formaté

- La balise <pre> indique une zone de texte préformaté
- La balise <pre> est idéale pour la présentation de texte avec une syntaxe spéciale : par exemple du code source

```
public class NHibernateContractResolver : DefaultContractResolver {
    protected override List<MemberInfo> GetSerializableMembers(Type objectType) {
        if (typeof(INHibernateProxy).IsAssignableFrom(objectType)) {
            return base.GetSerializableMembers(objectType.BaseType);
        } else {
            return base.GetSerializableMembers(objectType);
        }
    }
}
```

Détails (HTML5)

- La balise **details** consiste à définir un contenu qui est caché par défaut que l'utilisateur peut montrer quand il le souhaite
- Seul la balise **summary** est visible par défaut

```
<details><summary>Formation web par Mr Mostefai</summary>
<p>La formation est composée des rubriques suivantes : HTML, CSS et JavaScript</p></details>
```

► Formation web par Mr Mostefai

▼ Formation web par Mr Mostefai

La formation est composée des rubriques suivantes : HTML, CSS et JavaScript

Caractères spéciaux

- Le standard HTML demande de respecter le codage des caractères ASCII 7 bits, c'est-à-dire que les caractères accentués ne sont pas autorisés. Il faut pour cela utiliser un codage particulier
- Pour coder un caractère accentué, il suffit de saisir une combinaison précédée du caractère & et terminée par un point-virgule (;).

Caractères spéciaux – Liste de quelques caractères

34

- Le tableau ci-dessous donne quelques exemples de caractères spéciaux

Caractère	Code	Caractère	Code
&	&	¶	¶
€	€	À	À
<	<	Ç	Ç
>	>	È	È
Espace	&nbsp	É	É
§	§	Ê	Ê
©	©	À	à
®	®	é	é
±	±	è	è

Caractères spéciaux – Exemple

```
<p>  
Il &eacute;tait un excellent &eacute;l&egrave;ve  
</p>
```

Il était un excellent élève

Exercice 1

- Dans la perspective de créer une page web sur son CV personnel, créer une page contenant une mini-biographie **SUR VOUS**

Section 3 : Les Listes

Listes dans HTML

- HTML inclut plusieurs types de listes : listes ordonnées, listes non ordonnées et listes de définitions
- Les listes ordonnées contient une séquence d'éléments dont la position dans la liste est importante
- Les listes non ordonnées contient des éléments dont l'ordre n'est pas important
- Les listes de définitions contiennent des listes de termes et leurs définitions
- Il est possible d'incorporer une liste dans une autre

Listes ordonnées

- La liste est définie par la balise `` et les éléments par la balise ``

`<p>`

Le classement des gagnants est le suivant :

``

`Mr X X`

`Mr Y Y`

`Mr Z Z`

``

`</p>`

Le classement des gagnants est le suivant :

1. Mr XX
2. Mr YY
3. Mr ZZ

Listes non ordonnées

- La liste est définie par la balise `` et les éléments par la balise ``

`<>p>`

Pour réussir son projet, il faut :

``

`Un peu de chance`

`Un peu de courage`

`Beaucoup d'argent`

`</p>`

Pour réussir son projet, il faut :

- Un peu de chance
- Un peu de courage
- Beaucoup d'argent

Listes de définitions

- La liste est définie par la balise `<dl>`
- La balise `<dt>` introduit un terme et la balise `<dd>` donne sa définition

`<dl>`

`<dd>Adresse IP</dd>`

`<dt>Numéro servant à identifier chaque Ordinateur connecté à Internet</dt>`

`<dd>HTTP</dd>`

`<dt>Protocole de transmission de documents hypermédia. Il est utilisé pour transférer des documents hypertextes ou des documents hypermédia entre un serveur et un client</dt></dl>`

Adresse IP

Numéro servant à identifier chaque Ordinateur connecté à Internet

HTTP

Protocole de transmission de documents hypermédia. Il est utilisé pour transférer des documents hypertextes ou des documents hypermédia entre un serveur et un client

Exercice 2

- Ajouter l'historique de votre expérience professionnelle ainsi que vos diplômes au CV commencé dans l'exercice 1
- La balise `<dt>` introduit un terme et la balise `<dd>` donne sa définition

Section 4 : Les Liens

Les liens

- Les liens sont l'essence même du web car ils permettent de relier des pages entre elles
- Les liens entre les pages sont la source de l'analogie entre internet et la toile d'araignée (web)
- Le bon choix du contenu et de la disposition des liens est primordial pour la **navigabilité** du site

Structure d'un lien

```
<a href="http://www.esi.dz">école</a>
```

Balise lien

Cible du lien

Contenu du
lien

Visitez notre [école](#) pour en savoir plus !

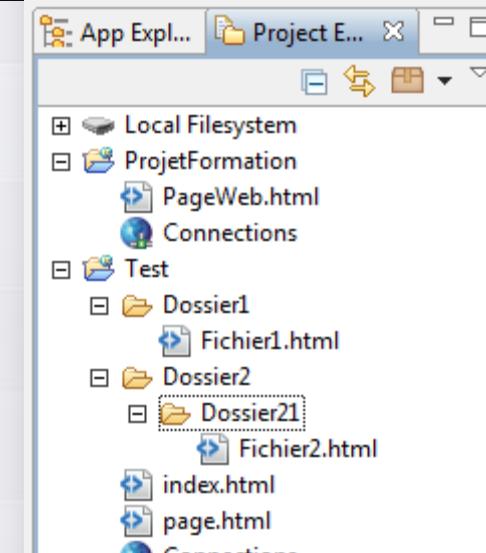
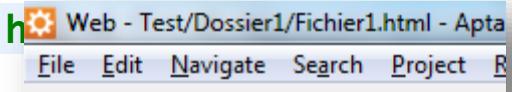
URLs absolues et URLs relatives

46

- Chaque ressource dans le web a un identifiant unique qui s'appelle URL (Unified Resource Locator)
- Une **URL absolue** commence par le nom de domaine du site contenant la ressource
- Quand les pages appartiennent au même site, on peut faire usage des **URLs relatives**
- Les URLs relatives ne commencent pas par le nom de domaine mais indiquent une position par rapport à la page en cours

URLs absolues et URLs relatives - Exemple

```
<a href="page.html">Une page dans le même dossier</a> <!-- index.html -->
<a href="Dossier1/Fichier1.html">Une page dans un sous-dossier</a> <!-- index.html -->
<a href="Dossier2/Dossier21/Fichier2.html">Une page dans un autre sous-dossier</a>
<a href="Dossier1/Fichier1.html">Une page dans un sous-dossier</a> <!-- index.html -->
<a href="../index.html">Lien vers un dossier parent</a><!-- Fichier1.h
```



Liens vers des mails

- Un lien peut déclencher le programme client mail au lieu de pointer vers une ressource, il suffit d'utiliser « mailto » dans la cible du lien

```
<body>  
Contactez-moi en cliquant <a href="mailto:mostefaiamine@gmail.com">ici</a>  
</body>
```

Contactez-moi en cliquant [ici](mailto:mostefaiamine@gmail.com)

Ouvrir un lien dans une nouvelle fenêtre

- Pour ouvrir un lien dans une nouvelle fenêtre, utiliser l'attribut « target » avec la valeur « _blank »

```
<body>
```

```
Et voici un <a href="http://www.google.com" target="_blank">moteur de recherche</a> intéressant  
</body>
```

Et voici un [moteur de recherche](http://www.google.com) intéressant

Liens vers une partie de la page

- Un lien peut pointer sur un élément particulier dans la page
- Un clic sur son lien provoque la visualisation de cet élément
- Très pratique pour les longues pages

```
<ol>
<li><a href="#par1">Paragraphe 1</a></li>
<li><a href="#par2">Paragraphe 2</a></li>
<li><a href="#par3">Paragraphe 3</a></li>
</ol>
<p id="par1">
```

HTML5 improves interoperability and reduces development costs by making precise rules on how to handle all HTML elements, and how to recover from errors.

```
</p>
<p id="par2">
```

Some of the new features in HTML5 are functions for embedding audio, video, graphics, client-side data storage, and interactive documents. HTML5 also contains new elements like nav, header, footer, and figure.

```
</p>
<p id="par3">
```

The HTML5 working group includes AOL, Apple, Google, IBM, Microsoft, Mozilla, Nokia, Opera, and many hundreds of other vendors.</p>

- ```
1. Paragraphe 1
2. Paragraphe 2
3. Paragraphe 3
```

HTML5 improves interoperability and reduces development costs by making precise rules on how to handle all HTML elements, and how to recover from errors.

Some of the new features in HTML5 are functions for embedding audio, video, graphics, client-side data storage, and interactive documents. HTML5 also contains new elements like nav, header, footer, and figure.

# Liens vers une partie d'une autre page

- Le même principe sauf que avant de mettre l'id de l'élément, on met l'adresse de la page

```
Yahoo pied de page
```

# Exercice 3

- Dans le CV, dans la section de l'expérience professionnelle et des diplômes, mettre les liens vers les institutions en question

# Section 5 : Images

# Introduction aux images

- Peut-on imaginer le web sans les images ?
- Une image est plus parlante qu'un millier de mots
- Les images sont très importantes à la fois en terme de contenu et aussi en terme de design
- Choisir les bonnes images est un facteur crucial pour la bonne conception d'un site

# Où trouver les images ?

- Les prendre soi-même
- Les concevoir soi-même
- Utiliser les moteurs de recherche d'images (Google, Bing,...)
- Faire attention aux copyrights
- Les acheter à des organismes tiers ou des sites spécialisés (istockphoto, fotolia,...)

# Organiser les images

- Les sites en mode production contiennent souvent beaucoup d'images
- Une bonne organisation consiste à mettre ces images dans un dossier séparé (appelé Images ou Img par exemple) voire même de structurer ce dossier en sous-dossiers selon la thématique

# La balise img

- La balise img est la balise permettant d'insérer une image dans une page
- L'attribut **src** indique où se trouve l'image (ça peut être en local ou sur un autre site)
- L'attribut **alt** est un texte alternatif qui s'affiche si l'image ne se charge pas
- L'attribut **title** donne un intitulé à l'image

```

```

# La balise img / Exemple

```
<p>Le bourak est un des éléments les plus importants de la table du ramadhan</p>
<div>
</div>
```

Le bourak est un des éléments les plus importants de la table du ramadhan



# Taille des images

- Les images prennent plus de temps à télécharger que le code du HTML
- Plus l'image est grande, plus son volume est grand plus elle impacte la disposition de la page où elle se trouve
- Pour imposer une taille d'une image il y a deux solutions : la modifier à la source par un éditeur d'images ou utiliser les attributs **width** et **height**
- Utiliser les attributs width et height règle le problème de disposition mais pas celui du téléchargement
- Lorsqu'on utilise width et height, si on ne fait pas le bon rapport, on risque de déformer l'image à l'affichage

# Taille des images - Exemple

```
<p>
Le bourak est un des éléments les plus importants de la table du ramadhan
</p>
<div>

</div>
```

Le bourak est un des éléments les plus importants de la table du ramadhan



# Emplacement des images

- Avant un paragraphe
- Au début d'un paragraphe
- Au milieu d'un paragraphe



Le bourak est un des éléments les plus importants de la table du ramadhan



Le bourak est un des éléments les plus importants de la table du ramadhan



Le bourak est un des éléments les plus importants de la table du ramadhan

# Alignement des images

- L'attribut **align** permet d'aligner une image par rapport au texte où elle se trouve
- Cet attribut est obsolète, il faut désormais utiliser le CSS

```
<p>Le bourak est un des éléments les plus importants
 de la table du ramadhan</p>
```

Le bourak est un des éléments les plus importants de la table du ramadhan



# Règles à respecter

- Choisir le bon **format**
- Choisir la bonne **taille**
- Choisir la bonne **résolution**

# Editeurs

- Il y a une pléthore d'éditeurs d'images dont le plus connu est **Adobe Photoshop**
- **Gimp** est une alternative open source de Photoshop

# Formats

- Les formats les plus utilisés du web sont JPEG, GIF et PNG
- **JPG (ou JPEG)** : Format idéal pour les photos et les illustrations compliquées. Contient des millions de couleurs. La Compression de l'image se fait avec une perte de qualité qui va de 0% à 99% (à choisir pour chaque image en fonction de son poids final).
- **GIF** : Très pratique pour les dessins et images simples, avec peu de nuances, ainsi que pour les logos. Peut contenir 256 couleurs ou moins. Ce qui veut dire qu'une image de plus de 256 couleurs enregistrée au format GIF ne va en retenir que 256. Gère la transparence et les animations.
- **PNG** : Format libre de droits. Convient pour tout type d'image, la transparence mais pas les animations. Il existe en 8 bits (type GIF, 256 couleurs) et en 24 bits (type JPEG, millions de couleurs, mais pas de compression).

# Résolution d'images

- La résolution est le nombre de **points (pixels)** par pouce (inch)
- Les équipements disposent de **résolutions différentes** (par exemple, la résolution d'une imprimante spécialisée (généralement 300dpi) est largement supérieure à celle d'un écran)
- Plus la résolution est grande, plus l'image est volumineuse
- La plupart des écrans ont une résolution de 72 pixels
- Inutile d'utiliser des images dont la résolution dépasse les médias ciblés

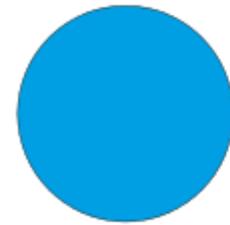
# Images par points et images vectorielles

67

- Les images par points représentent un carré de pixels
- Une image vectorielle est dessinée en suivant une **certaine procédure** de dessin (placer des points, les relier, remplir avec une couleur,...)
- Lorsqu'on agrandit une image vectorielle, il n'y a **aucune perte de qualité**
- Le format vectoriel utilisé pour le web est le format **SVG**
- Les images vectorielles sont créées avec des outils comme **Adobe Illustrator**
- Le support pour le moment est restreint aux navigateurs récents
- Côté performance, l'affichage des images par point est beaucoup plus rapide

# Images vectorielles - Exemple

```
<p>
 Les cercles sont plus jolis que les carrés
</p>
```

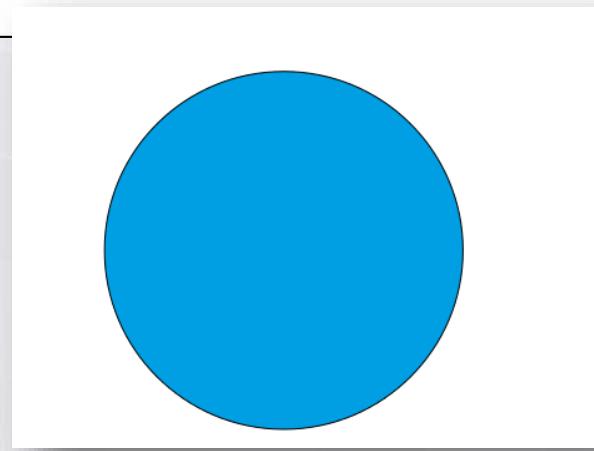


Les cercles sont plus jolis que les carrés

# Insérer directement du code SVG avec le HTML

- On peut directement mettre du code SVG dans la page web
- Ceci n'est supporté que par les navigateurs récents

```
<p><svg version="1.1" id="Layer 1" xmlns="http://www.w3.org/2000/svg"
 xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"
 width="400px" height="400px" viewBox="0 0 400 400" enable-background="new 0 0 400 400"
 xml:space="preserve">
 <circle fill="#009FE3" stroke="#000000" stroke-miterlimit="10" cx="209.783" cy="200.368"
 r="143.632"/>
</svg></p>
```



# GIF animées

- Les GIF animées représentent des petites animations d'images ne dépassant pas 256 couleurs
- Elles sont idéales pour illustrer un chargement ajax
- Elles ne sont pas convenables pour de vraies vidéos

```
<p></p>
```

```
<p>Chargement en cours...</p>
```



Chargement en cours...

# Transparence

- La transparence permet de donner des effets extraordinaires aux images
- Les seuls formats supportant la transparence sont GIF et PNG
- PNG n'est pas supporté dans IE6

```
<div style="background-color: green"></div>
```

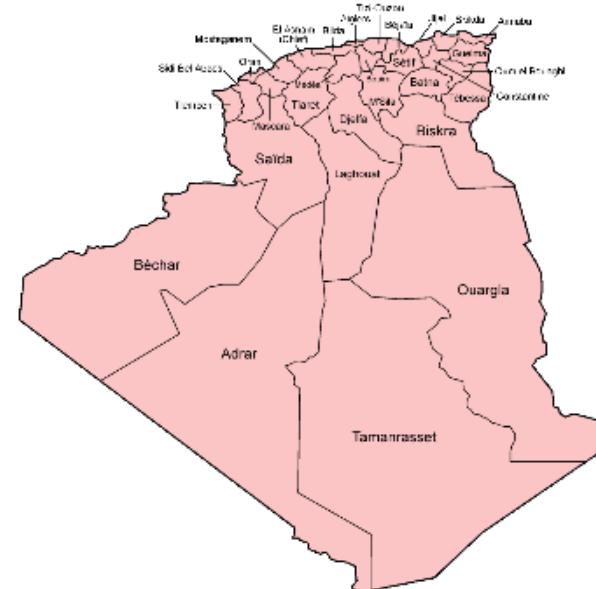


# Les figures (HTML 5)

- La balise figure indique une figure et figcaption l'intitulé ou la description de cette figure

```
<figure>

<figcaption>Les 48 wilayas de l'Algérie</figcaption></figure>
```



Les 48 wilayas de l'Algérie

# Exercice

- Ajouter votre photo à votre CV
- Ajouter une photo à votre mini-biographie

# Section 6 : Les Tableaux

# Introduction

- La balise **table** permet de présenter des données sous forme tabulaire
- Idéale pour les rapports, les factures, les listes,...
- Dans HTML, un tableau est un ensemble de **lignes** et une ligne est un ensemble de **cellules**

CLIO affiner par génération : <a href="#">TOUTES GÉNÉRATIONS</a>					
<a href="#">Liste</a> <a href="#">Galerie</a>		<a href="#">20 par page</a>	<a href="#">Trier par</a>	<a href="#">1 2 3 4 5 ▶</a>	
Marque / Modèle / Finition	Prix	Km	Année	Dpt.	Vend.
 RENAULT CLIO III (2) 1.5 DCI 90 DYNAMIQUE TOMTOM 5P EURO5	11 500 €	14 500	2011	78	Coll.
 RENAULT CLIO III 1.6 16V 110 DYNAMIQUE 5P XX	6 950 €	73 000	2006	01	Part.
 RENAULT CLIO II (3) CAMPUS 1.2 16V 75 .COM 5P EURO5 XX	8 480 €	10 390	2011	77	Pro.
 RENAULT CLIO III 1.5 DCI 85 CONFORT PACK CLIM EXPRESSION 5P	6 400 €	126 750	2006	75	Pro.
 RENAULT CLIO III (2) 1.5 DCI 70 EXPRESSION CLIM 5P	10 990 €	19 530	2010	92	Pro.
 RENAULT CLIO III (2) 2.0 16V 203 RS 20TH XX	16 990 €	17 788	2010	04	Pro.
 RENAULT CLIO III (2) 1.5 DCI 70 AUTHENTIQUE 5P	7 470 €	40 120	2010	92	Pro.

# Structure

- Un tableau est un ensemble de lignes (**tr**) composé de cellules (**td**)

```
<table>
<tr>
<td>15</td><td>Tizi Ouzou</td></tr>
<tr>
<td>16</td><td>Alger</td></tr>
<tr>
<td>17</td><td>Djelfa</td></tr>
<tr>
<td>18</td><td>Jijel</td></tr></table>
```

15 Tizi Ouzou  
16 Alger  
17 Djelfa  
18 Jijel

# Organisation des tableaux

- La balise **th** indique l'entête d'une colonne ou d'une ligne (définie par l'attribut scope)
- Les balises **thead**, **tbody** et **tfoot** définissent l'entête, le corps et le pied d'un tableau

```
<table>
<thead>
<tr>
<th scope="col">Région</th>
<th scope="col">1ère année</th>
<th scope="col">2ème année</th></tr></thead>
<tbody><tr>
<th scope="row">Centre</th>
<td>1 000 000</td>
<td>2 000 000</td></tr><tr>
<th scope="row">Est</th>
<td>5 000 000</td>
<td>3 000 000</td></tr></tbody>
<tfoot><th>Total</th>
<td>6 000 000</td>
<td>5 000 000</td></tfoot></table>
```

	Région	1ère année	2ème année
	Centre	1 000 000	2 000 000
	Est	5 000 000	3 000 000
Total	6 000 000	5 000 000	

# Fusionner des colonnes

- L'attribut **colspan** permet de fusionner plusieurs colonnes

```
<table style="background-color: lightGray">
<thead>
<tr>
<th scope="col">Semaine</th>
<th scope="col">Dimanche</th>
<th scope="col">Lundi</th>
<th scope="col">Mardi</th>
</tr>
</thead>
<tbody>
<tr>
<th scope="row" >1ère semaine</th>
<td colspan="2" class="green">Marketing</td>
<td class="red">Technique</td>
</tr>
<tr>
<th scope="row">2ème semaine</th>
<td colspan="3" class="yellow">Finances</td>
</tr>
</tbody>
</table>
```

Semaine	Dimanche	Lundi	Mardi
1ère semaine	Marketing		Technique
2ème semaine	Finances		

# Fusionner des lignes

- L'attribut **rowspan** permet de fusionner plusieurs lignes

```
<table style="background-color: lightGray">
<thead>
<tr>
<th scope="col">Semaine</th>
<th scope="col">Dimanche</th>
<th scope="col">Lundi</th>
<th scope="col">Mardi</th>
</tr>
</thead>
<tbody>
<tr>
<th scope="row" >1ère semaine</th>
<td rowspan="2" class="green">Marketing</td>
<td colspan="2" class="red">Technique</td>
</tr>
<tr>
<th scope="row">2ème semaine</th>
<td colspan="2" class="yellow">Finances</td>
</tr>
</tbody>
</table>
```

Semaine	Dimanche	Lundi	Mardi
1ère semaine	Marketing	Technique	
2ème semaine			Finances

# Fusionner des lignes

- L'attribut **rowspan** permet de fusionner plusieurs lignes

```
<table style="background-color: lightGray">
<thead>
<tr>
<th scope="col">Semaine</th>
<th scope="col">Dimanche</th>
<th scope="col">Lundi</th>
<th scope="col">Mardi</th>
</tr>
</thead>
<tbody>
<tr>
<th scope="row" >1ère semaine</th>
<td rowspan="2" class="green">Marketing</td>
<td colspan="2" class="red">Technique</td>
</tr>
<tr>
<th scope="row">2ème semaine</th>
<td colspan="2" class="yellow">Finances</td>
</tr>
</tbody>
</table>
```

Semaine	Dimanche	Lundi	Mardi
1ère semaine	Marketing	Technique	
2ème semaine		Finances	

# Espacement et remplissage

- **Cellspacing** et **cellpadding** définissent l'espacement et le remplissage des cellules
- Ces attributs sont obsolètes, il faut passer par CSS pour réaliser leur effet

```
<table cellpadding="5" cellspacing="10">
<tr>
<td>15</td><td>Tizi Ouzou</td>
</tr>
<tr>
<td>16</td><td>Alger</td>
</tr>
<tr>
<td>17</td><td>Djelfa</td>
</tr>
<tr>
<td>18</td><td>Jijel</td>
</tr>
</table>
```

15	Tizi Ouzou
16	Alger
17	Djelfa
18	Jijel

# Exercice

- Mettre en forme l'expérience professionnelle sous forme de tableau

# Section 7 : Les Formulaires

# Introduction

- Pour le moment, l'adresse et les URLs étaient les seuls moyens d'envoyer des données vers les serveurs
- Le web a très évolué de simple sites web vers de véritables applications web où les données sont le noyau de ces applications
- Pour qu'une application soit utile, il ne suffit pas qu'elle affiche des données, il faut qu'elle donne à l'utilisateur de créer ou de mettre à jour ces données
- Les formulaires sont le moyen de collecte de données depuis les utilisateurs
- Exemple : connexion, inscription, création d'un produit, ajout d'un commentaire,...

# Fonctionnement

- Un formulaire est composé de plusieurs **contrôles** et d'un **bouton d'envoi**
- Chaque contrôle a un **nom unique** qui le distingue défini par l'attribut **name**
- Une fois le bouton envoi appuyé, le navigateur formate les données du formulaire sous forme de paires **nom / valeur** puis les envoie au serveur. Cette opération est appelée **sérialisation**.
- Plusieurs méthodes de sérialisation existent : POST, JSON,... la plus populaire et simple est la sérialisation POST
- Une fois que le serveur reçoit les données, il les traite par exemple il enregistre une entité dans la base de donnée
- Une fois l'opération s'est bien (ou mal) déroulé, le serveur renvoie une page de redirection.

# Envoi de données - Exemple

- Soit le formulaire suivant :

```
<form method="post" >
<p>Utilisateur : <input type="text" name="login" /></p>
<p>MDP : <input type="text" name="mdp" /></p>
<input type="submit" value="Envoyer"/>
</form>
```

- Lorsque l'utilisateur appuie sur envoyer, les données sont envoyées comme suit :

login=Amine&mdp=mot+de+passe

The screenshot shows a simple HTML form with two text input fields and one submit button. The first field is labeled "Utilisateur" and contains the value "Amine". The second field is labeled "MDP" and contains the value "mot de passe". Below the fields is a blue "Envoyer" button.

# La balise form

- La balise **form** définit un formulaire
- Sa syntaxe est comme suit : <**form action="url" method="post"**>
- L'attribut **action** donne l'url de la page qui va recevoir les informations
- L'attribut **method** définit la méthode d'envoi (GET ou POST)

# Méthodes GET et POST

- Avec la méthode GET, les données sont ajoutée à la fin de l'URL définie dans l'attribut action
- Avec la méthode POST, les données sont transmises dans le corps de la requête HTTP
- La méthode GET est idéale pour les très petits formulaires contenant des données pas sensibles
- La méthode POST permet de : uploader un fichier, encapsuler des données sensibles, gérer les connexions cryptées

# Démonstration GET

- Soit le formulaire suivant :

```
<form action="url" method="get" >
<p>Utilisateur : <input type="text" name="login" /></p>
<p>MDP : <input type="text" name="mdp" /></p>
<input type="submit" value="Envoyer"/>
</form>
```

- L'appui sur envoyer redirige vers l'URL suivante :

<http://127.0.0.1:8020/Formation/url?login=Amine&mdp=mot+de+passe>

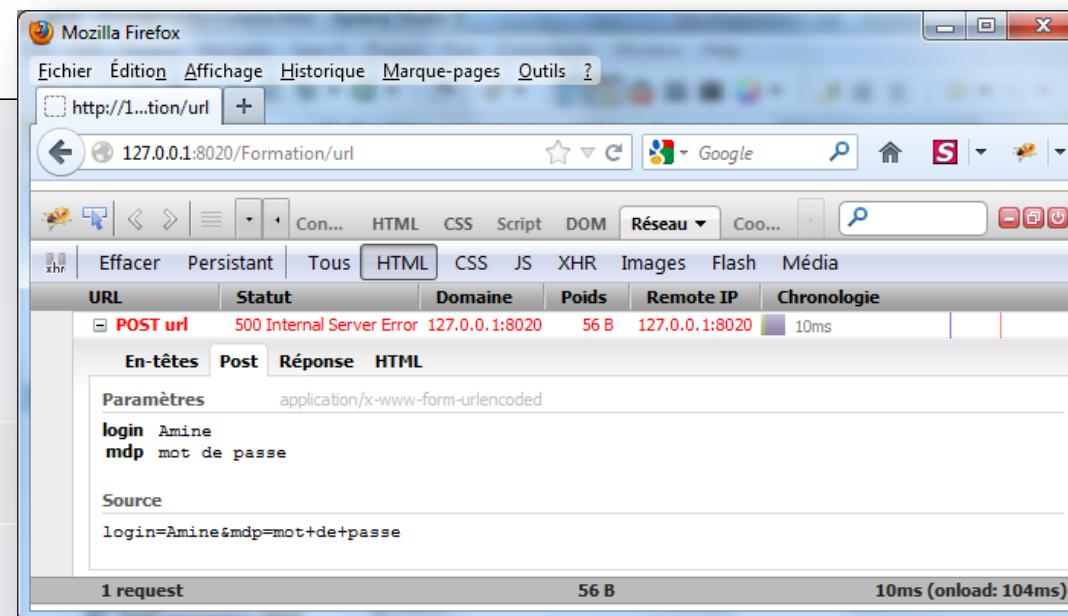
The screenshot shows a simple HTML form for user authentication. It consists of two text input fields and a submit button. The first field is labeled "Utilisateur" and contains the value "Amine". The second field is labeled "MDP" and contains the value "mot de passe". Below the inputs is a blue rectangular button labeled "Envoyer".

# Démonstration POST

- Soit le formulaire suivant :

```
<form action="url" method="post" >
<p>Utilisateur : <input type="text" name="login" /></p>
<p>MDP : <input type="text" name="mdp" /></p>
<input type="submit" value="Envoyer"/>
</form>
```

- Ouvrir la fenêtre firebug dans firefox puis cliquez sur envoyer
- Examiner l'onglet « réseau » dans firebug



# Encodage

- L'attribut **enctype** définit comment les données sont encodées lorsqu'elles sont envoyées au serveur. Les valeurs suivantes sont possibles :

Encodage	Description
application/x-www-form-urlencoded	Valeur par défaut. Les caractères sont encodés en ASCII et les espaces remplacés par des « + »
multipart/form-data	Les caractères ne sont pas encodés. Utilisé lorsque le formulaire inclut un upload.
text/plain	Les caractères ne sont pas encodés mais les espaces sont remplacés par des « + »

# Attributs HTML5 de la balise form

- L'attribut **autocomplete** définit si les contrôles du formulaires s'auto-remplissent. Les valeurs possibles **sont** *on* ou **off**.
- L'attribut **novalidate** lorsqu'il est présent, indique que la validation est désactivée pour le formulaire.

# La saisie de texte

- La saisie de texte est effectuée avec la balise **input** avec l'attribut **type** ayant la valeur « **text** »
- L'attribut **maxlength** définit le nombre maximum de caractères qu'un utilisateur peut entrer dans la zone de texte

```
<p>Nom : <input name="nomText" maxlength="30" type="text" /></p>
```

Nom :

# La saisie de mots de passe

- La saisie de mots de passe est effectuée avec la balise **input** avec l'attribut **type** ayant la valeur « **password** »
- L'attribut **maxlength** définit le nombre maximum de caractères qu'un utilisateur peut entrer dans la zone de mdp

```
<p>Login : <input name="loginText" maxlength="30" type="text" /></p>
<p>MDP : <input name="passText" maxlength="8" type="password" /></p>
```

The screenshot shows a simple web form with two text input fields. The first field is labeled "Login :" and contains the text "Amine". The second field is labeled "MDP :" and contains four black dots, representing a password. This visualizes how the "password" type attribute hides the input value.

# La saisie de long texte

- La saisie de mots de passe est effectuée avec la balise **textarea**
- L'attribut **rows** définit le nombre de lignes visibles et l'attribut **cols**, le nombre de colonnes

```
<p>Vos commentaires :</p>
<textarea name="comments" rows="5"></textarea>
```

Vos commentaires :

Entrez les commentaires ici

# Boutons radio

- Le bouton radio est effectué avec la balise **input** avec l'attribut **type** ayant la valeur « **radio** »
- L'attribut **value** indique la valeur qui va être envoyée au serveur
- L'attribut **checked** indique si le bouton est coché ou pas

<p>

Votre langage de programmation préféré :<br />

```
<input name="langage" value="c#" type="radio" />C#
<input name="langage" value="java" type="radio"/>Java
<input name="langage" value="php" type="radio"/>PHP
</p>
```

Vos langage de programmation préféré :

C#  Java  PHP

# Les cases à cocher

- La case à cocher est effectuée avec la balise **input** avec l'attribut **type** ayant la valeur « **checkbox** »
- L'attribut **value** indique la valeur qui va être envoyée au serveur
- L'attribut **checked** indique si le bouton est coché ou pas

```
<p>
Vos marques préférées :

<input name="marque" value="microsoft" type="checkbox" />Microsoft
<input name="marque" value="apple" type="checkbox"/>Apple
<input name="marque" value="google" type="checkbox"/>Google
</p>
```

Vos marques préférées :  
 Microsoft  Apple  Google

# Les listes déroulantes

- La liste déroulante est définie par la balise **select**, la balise **option** quant-à-elle définit les éléments de cette liste
- L'attribut **value** indique la valeur qui va être envoyée au serveur
- L'attribut **selected** indique l'élément sélectionné après le chargement de la page

```
<p>
Votre wilaya d'origine:

<select name="wilaya">
<option value="16">Alger</option>
<option value="17">Djelfa</option>
<option value="18" selected="selected">Jijel</option>
</select>
</p>
```

Votre wilaya d'origine:

# Les listes à sélection multiples

- La liste à sélection multiple est définie par la balise **select**, la balise **option** quant-à-elle définit les éléments de cette liste
- L'attribut **size** indique le nombre d'éléments visibles simultanément
- L'attribut **multiple** indique que l'utilisateur peut sélectionner plusieurs éléments

<p>

Les langues que vous parlez:<br />

```
<select name="langue" multiple="multiple">
<option value="ar" selected="selected">Arabe</option>
<option value="fr" selected="selected">Français</option>
<option value="en" >Anglais</option>
</select>
</p>
```

Les langues que vous parlez:



# Uploader des fichiers

100

- L'upload de fichiers est effectué avec la balise **input** avec l'attribut **type** ayant la valeur « **file** »
- Le navigateur ajoute automatiquement le bouton « parcourir » au contrôle
- Pour pouvoir faire des uploads, la méthode du formulaire doit être obligatoirement **POST** et enctype à **multipart/form-data**

```
<form method="post" enctype="multipart/form-data">
<p>
Veuillez inclure votre CV svp

<input type="file" name="cv" />
</p>
<input type="submit" value="ok" />
</form>
```

Veuillez inclure votre CV svp  
F:\Docs\ToSend\watch... Parcourir...

ok

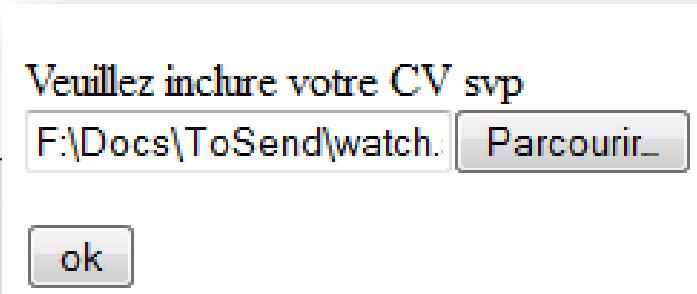
# Envoyer les données du formulaire

101

- L'envoi de données est effectué par un bouton défini par la balise **input** avec l'attribut **type** ayant la valeur « **submit** »
- L'attribut **value** indique le titre du bouton

```
<form method="post" enctype="multipart/form-data">
<p>
Veuillez inclure votre CV svp

<input type="file" name="cv" />
</p>
<input type="submit" value="ok" />
</form>
```



# Boutons de type image

102

- Le bouton de type image est défini par la balise **input** avec l'attribut **type** ayant la valeur « **image** »

```
<form method="post" >
<p>
Nom :
<input type="text" name="nom" />
</p>
<input type="image" src="Images/bouton.gif" />
</form>
```

Nom :



# Les boutons

- Le boutons est défini par la balise **button**
- Un bouton transforme son contenu en bouton

```
<form method="post">
<p>
Nom :
<input type="text" name="nom" />
</p>
<button value="send" >Envoyer</button>
</form>
```

Nom :

**Envoyer**

# Les champs cachés

- Un champ caché est défini par la balise **input** avec l'attribut **type** ayant la valeur **hidden**
- Les champs cachés sont très importants car ils intègrent des informations qui ne doivent pas être visibles à l'utilisateur mais qui doivent être remontées au serveur
- L'attribut **value** donne la valeur du champ caché. La seule façon de la voir est de voir le code source

```
<form method="post">
<p>
Nom :
<input type="text" name="nom" />
<input type="hidden" name="id" value="55"/>
</p>
<button value="send" >Envoyer</button>
</form>
```

The screenshot shows a simple HTML form. It contains a text input field with the placeholder "Nom :" and a blue "Envoyer" button below it. The entire form is enclosed in a light gray border.

# Libeller les contrôles

- Les contrôles sont libellés en utilisant la balise label
- Il y a deux façons de libeller un contrôle : l'intégrer dans une balise label ou utiliser l'attribut for
- L'attribut for indique quel contrôle le libellé pointe, il doit être égal de l'attribut id du contrôle en action

```
<p>
<label>Nom : <input type="text" name="nom" /></label>

Vos marques préférées :

<input name="marque" value="microsoft" type="checkbox" id="cbMicrosoft" />
<label for="cbMicrosoft">Microsoft</label>
<input name="marque" value="apple" type="checkbox" id="cbApple"/>
<label for="cbApple">Apple</label>
<input name="marque" value="google" type="checkbox" id="cbGoogle"/>
<label for="cbGoogle">Google</label></p>
```

Nom :

Vos marques préférées :

Microsoft  Apple  Google

# Regrouper des éléments

- La balise **fieldset** groupe les éléments ayant un certain lien dans un formulaire
- La balise **legend** donne un intitulé au groupe (fieldset)

```
<form action="url" method="post" enctype="text/plain" >
<fieldset>
<legend>Identité</legend>
<label>Nom :

<input type="text" name="nom" />
</label>
</fieldset>
<fieldset>
<legend>Marques</legend>
Vos marques préférées :
<input name="marque" value="microsoft" type="checkbox" id="cbMicrosoft" />
<label for="cbMicrosoft">Microsoft</label>
<input name="marque" value="apple" type="checkbox" id="cbApple"/>
<label for="cbApple">Apple</label>
<input name="marque" value="google" type="checkbox" id="cbGoogle"/>
<label for="cbGoogle">Google</label>
</fieldset>
</form>
```

The screenshot shows a web page with a form containing two fieldsets. The firstfieldset is titled "Identité" and contains a label "Nom :" followed by a text input field. The secondfieldset is titled "Marques" and contains the text "Vos marques préférées :" followed by three checkboxes labeled "Microsoft", "Apple", and "Google".

# Effacer un formulaire

- On efface un formulaire en ajoutant un contrôle **input** avec le type égal à **reset**

```
<form method="post" >
<p>
Nom :
<input type="text" name="login" required="required" placeholder="nom"/></p>
<p>
Spécialité :
<input type="text" pattern="SIT|SIQ|SIL" required="true" placeholder="spécialité" />
</p>
<input type="submit" value="Envoyer" />
<input type="reset" value="effacer" />
</form>
```

The screenshot shows a simple HTML form. At the top, there is a text input field labeled "Nom" with the placeholder "nom". Below it is another text input field labeled "Spécialité" with the placeholder "spécialité". At the bottom of the form are two buttons: a grey "Envoyer" button and a blue "effacer" button.

# Validation des formulaires

108

- Les données envoyées par les formulaires doivent être validées pour veiller à une certaine cohérence et une intégrité des données
- Exemples : un champ est obligatoire (vide non accepté), un champ doit être entre une valeur minimale et maximale, un mail doit obéir à un certain format (expression régulière)
- Il existe trois types de validation : **côté client**, **côté serveur** et **validation HTML5**
- La validation côté serveur n'effectue la validation qu'une fois les données sont postées au serveur
- La validation côté client est effectuée par Javascript sans passer par le serveur
- La validation HTML5 enlève le besoin d'utiliser Javascript pour valider

# Validation des formulaires - Comparaison

109

Type de validation	Avantages	Inconvénients
Serveur	Garantit qu'aucune donnée incohérente n'est intégrée au système	Surcharge le serveur
Client	Décharge le serveur des calculs	<ul style="list-style-type: none"><li>- Programmation supplémentaire</li><li>- Si elle est contournée, des données incohérentes peuvent être envoyées</li></ul>
HTML5	Pas de programmation supplémentaire Décharge les serveur des calculs	<ul style="list-style-type: none"><li>- Nécessite le support HTML 5</li></ul>

# Validation HTML5 des champs texte

110

- L'attribut **required** indique que le champ est obligatoire
- L'attribut **pattern** définit une expression régulière à laquelle doit obéir la valeur du champ
- L'attribut **placeholder** donne une indication de la valeur à entrer

<Nom :

<p>

Nom :

<input type="text" name="login" required="required" placeholder="nom"/></p>

<p>

Spécialité :

<input type="text" pattern="SIT|SIQ|SIL" required="true" placeholder="spécialité" /></p>

The screenshot shows a web form with two input fields. The first field, labeled "Nom :" with placeholder "nom", contains "sffg". The second field, labeled "Spécialité :" with placeholder "spécialité", contains "sd" and is highlighted with a red border, indicating it is invalid. Below the fields is a blue "Envoyer" button.

# Entrée des champs numériques (HTML5)

111

- Utiliser la balise **input** avec l'attribut type à **number**
- Les attributs **min** et **max** définissent l'intervalle de la valeur

```
<p>
```

Nom :

```
<input type="text" name="login" required="required"/>
```

```
</p>
```

```
<p>
```

Age :

```
<input type="number" name="age" required="true" min="18" max="35" />
```

```
</p>
```

Nom : sfgf

Age : 99

Envoyer

# Entrée des champs date (HTML5)

- Utiliser la balise **input** avec l'attribut type à **date**
- Les attributs **min** et **max** définissent l'intervalle de la valeur

<p>

Nom :

```
<input type="text" name="login" required="required"/>
```

</p>

<p>

D.Naissance :

```
<input type="date" name="dnaiss" required="true" />
```

The screenshot shows a web page with a light gray background. At the top, there is some faint text that reads "Technologies et Développement Web" and "112". Below this, the main content area has a white background. It contains two form fields. The first field is labeled "Nom :" followed by a text input box. The second field is labeled "D.Naissance :" followed by a date input box, which is currently set to "Jour/Mois/Année". Below these fields is a single button labeled "Envoyer".

# Entrée de champs mail et URL (HTML5)

- Utiliser la balise **input** avec l'attribut type à **date**
- Les attributs **min** et **max** définissent l'intervalle de la valeur

<p>

Nom :

```
<input type="text" name="login" required="required"/>
```

</p>

<p>

Mail :

```
<input type="email" name="mail" required="true" /></p>
```

<p>

Site perso :

```
<input type="url" name="url" required="true" />
```

</p>

Nom : amine

Mail : amine

Site perso : mo

Envoyer

# Autres types de saisie

114

- HTML 5 ajoute plusieurs nouvelles options pour l'attribut type de la balise input
- La valeur **search** indique un champ de recherche
- La valeur **color** indique la saisie de couleur
- La valeur **time** indique la saisie d'une heure

# Exercice

115

- Dans votre site personnel, ajoutez une page de contact où l'intéressé fournirait le maximum d'informations personnelles le concernant

# Section 8 : Organisation

# Commentaires

117

- Pour un développeur web, il est important d'ajouter des commentaires dans son code pour plus de lisibilité
- Les commentaires ne sont pas visibles à l'affichage, mais sont visibles lorsque l'utilisateur affiche le source
- Un commentaire est ajouté avec la balise `<!-- -->`

```
<!-- Le premier paragraphe est une introduction-->
<p>1er par</p>
<!-- Le deuxième raconte les détails-->
<p>2ème par</p>
```

1er par

2ème par

# Identification et regroupement des éléments

118

- L'attribut ***id*** est utilisé pour identifier un élément unique
- L'attribut ***class*** permet de désigner des éléments ayant des points en commun
- Les attributs ***id*** et ***class*** sont très utiles pour les utiliser dans CSS et Javascript
- Un élément peut appartenir à plusieurs classes (séparer les noms de classe par des espaces)

```
<p id="p1" class="parAvecIndent">
1er par
</p>
<p id="p2">
2ème par
</p>
<p id="p3" class="parAvecIndent parEnGras">
3ème par</p>
```

# Affichage en mode bloc

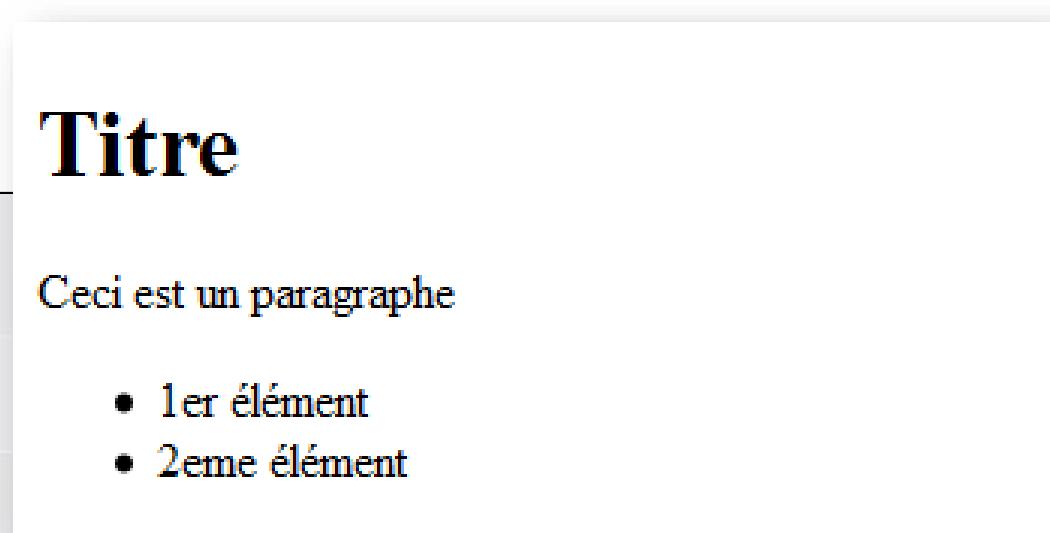
119

- Un élément affiché en mode **bloc** provoque toujours un retour à la nouvelle ligne dans la fenêtre du navigateur
- h1, p, ul, table et li sont quelques exemples d'éléments affichés en mode bloc

```
<h1>Titre</h1>
<p>Ceci est un paragraphe</p>

1er élément
2eme élément

```



Titre

Ceci est un paragraphe

- 1er élément
- 2eme élément

# Affichage en mode ligne

120

- L'affichage en mode ***ligne*** ressemble à l'affichage du texte naturel, chaque élément est affiché à côté des éléments adjacents et le retour à la nouvelle ligne n'est provoqué que si il n'y a plus d'espace
- em, img, a et b sont des exemples de quelques balises en mode ligne
- Les éléments en bloc ***ne peuvent pas être insérés*** dans des éléments en mode en ligne
- Les éléments en ligne ignorent certains attributs (css) comme la largeur ou la largeur max : elle est définie par leur contenu
- En utilisant le **CSS**, on peut modifier l'affichage par défaut d'un élément

< p >

L'affichage en mode **ligne** ressemble à l'affichage du **texte**

# Grouper du texte et des éléments en mode bloc

121

- La balise **div** est utilisée pour grouper du texte ou d'autres éléments en mode bloc
- En ajouter les attributs **id** et/ou **class**, le CSS permet grâce à la balise **div** de définir la disposition de la page

```
<div id="entete">
Lien1|Lien2
</div>
<div id="contenu">
Contenu de la page
</div>
```

Lien1|Lien2  
**Contenu de la page**

# Grouper du texte et des éléments en mode ligne

122

- La balise **span** est utilisée pour grouper du texte ou d'autres éléments en mode ligne
- En ajouter les attributs **id** et/ou **class**, le CSS permet grâce à la balise **span** de définir un comportement ou un style particulier à certains éléments

< p >

Une des erreurs les plus **communes** est d'affecter une largeur à un élément en ligne

< / p >

Une des erreurs les plus communes est d'affecter une largeur à un élément en ligne

# Intégrer une page provenant d'un autre site

123

- La balise **iframe** permet d'intégrer une page d'une autre site
- L'attribut **src** indique l'url de la page, **width** et **height** indiquent la largeur et la hauteur de l'intégration
- **iframe** est déconseillé à utiliser et n'est plus supporté dans HTML5

< p > Ceci est un paragraphe </ p >

< iframe src = "http://www.esi.dz" width = "800" height = "600" > </ iframe >



# Les métadonnées

124

- Une métadonnée est une donnée sur la donnée. En HTML les métadonnées sont définies par la balise **meta**
- L'attribut **name** définit le nom de la métadonnée
- L'attribut **content** définit la valeur de la métadonnée
- L'attribut **http-equiv** fournit une entête http traitant des informations spéciales fournies dans **content**
- Une métadonnée est définie par **name OU http-equiv**
- Les métadonnées sont ajoutée à l'entête (head) et ne sont pas affichée. Elles sont accédées par les robots ou en affichant le source
- L'utilisateur peut définir ses propres métadonnées

# Quelques métadonnées

125

Métadonnée	Description	Exemple
author	Donne l'auteur de la page	<code>&lt;meta name="author" content="Mostefai Mohammed Amine" /&gt;</code>
description	Donne un résumé sur la page	<code>&lt;meta name="description" content="Un article sur quelques nouveautés sur HTML5" /&gt;</code>
keywords	Mots clé relatifs à la page	<code>&lt;meta name="keywords" content="HTML5, développement web, formation, nouveauté" /&gt;</code>
refresh	Rafraîchissement automatique	<code>&lt;meta http-equiv="refresh" content="30" /&gt;</code>
refresh	Redirection après N secondes	<code>&lt;meta http-equiv="refresh" content="30" url="http://www.esi.dz" /&gt;</code> <code>&lt;/head&gt;</code>
pragma	Empêche le navigateur de mettre la page en cache	<code>&lt;meta http-equiv="pragma" content="no-cache" /&gt;</code>
expires	Indique l'expiration et force le navigateur à recharger la page	<code>&lt;meta http-equiv="Expires" content="03 Jun 2013 14:30:00"/&gt;</code>

# Exercice

126

- Sur la page du CV personnel, ajouter les métadonnées que vous jugez nécessaires
- Ajoutez des éléments de groupement en mode ligne et mode bloc

# Section 9 : Audio et Vidéo

# Ajouter une vidéo à son site web

128

- Il existe trois moyens d'insérer une vidéo : utiliser un **service vidéo** dédié (par exemple youtube), utiliser des **plugins** (Flash ou Silverlight) ou utiliser la balise HTML 5 video
- L'avantage d'utiliser un service vidéo est que l'on ne soucie pas des détails techniques tels que l'hébergement et l'encodage
- L'inconvénient est que la vidéo est accessible en dehors du site et qu'on n'est pas à l'abri de publicité ou d'annonces intrusives

# La vidéo en HTML5

129

- HTML 5 a introduit le support de la vidéo avec la nouvelle balise **video**
- **video** n'est supportée que par les navigateurs récents, pour les anciens il faut passer par d'autres techniques comme Flash ou Silverlight
- **video** offre une solution native (sans passer par un plugin ou un composant externe) permettant d'intégrer de la vidéo dans une page web
- **video** est l'une des plus grandes balises à succès dans HTML5 vu que des dispositifs populaires comme iPhone et iPad ne supportent pas le Flash

# Formats supportés

130

- La balise video supporte plusieurs formats dans trois tiennent le devant de la scène : **H.264/MP4, OGG/Theora** et **WebM**
- Quand on inclut une vidéo dans une page, il faut faire attention aux droits du format (gratuit ou non) et aussi à ceux du contenu
- **H.264** est supporté par le Moving Picture Experts Group. C'est un format non-libre (soumis à brevets) et non-gratuit. Toutefois, il est gratuit dans certaines utilisations (la diffusion gratuite de vidéos par des sites Web par exemple).
- **Theora** est un format de compression vidéo open-source, sans brevets. Ceci donne le droit à tous d'utiliser Theora (à des fins non commerciales tout comme à des fins commerciales) sans devoir payer de redevance au consortium MPEG.
- **WebM** est un format multimédia ouvert qui a été lancé par Google (après rachat de la société On2 Technologies). L'utilisation est en libre et gratuite.
- Il existe plusieurs outils de conversion dont **Miro converter** qui est open source et gratuit
- Le plus grand problème est que chaque navigateur à des formats qu'il ne supporte pas

# La balise video

131

- L'ajout d'une vidéo se fait en ajoutant la balise **video**
- L'attribut **src** indique la source (url) de la vidéo
- L'attribut **preload** indique le pré-chargement. La valeur **none** indique que la vidéo n'est pas chargée quand la page est chargée. La valeur **metadata** indique que seulement les métadonnées sont chargées. La valeur **auto** indique que la vidéo est chargée avec la page.
- L'attribut **controls** indique que les contrôles de lecture sont affichés
- L'attribut **autoplay** indique que la vidéo se lance dès qu'elle est chargée
- L'attribut **poster** indique une image affichée lorsque la vidéo se charge
- Les attributs **height** et **width** indiquent la taille de la vidéo sur la page
- L'attribut **loop** indique que la vidéo tourne en boucle
- L'attribut **muted** indique que la vidéo se lit sans le son

# La balise video - Exemple

132

- Ci-dessous un exemple du code HTML d'une vidéo

```
<video src="video/movie.theora.ogv" width="400" height="400" preload="auto" controls
autoplay>
<p>Elle est belle la nature</p>
</video>
```



# Une vidéo avec plusieurs formats

133

- La balise `video` peut supporter plusieurs sources pour permettre au maximum de navigateurs de visualiser une vidéo. Pour ce, la balise **source** est utilisée
- La balise **source** indique une source de vidéo, L'attribut **src** indique l'url de la source. L'attribut **type** indique le type.

```
<video width="400" height="400" preload="auto" controls autoplay>
<source src="video/movie.theora.ogv" type="video/ogg"/>
<source src="video/movie.webmvp8.webm" type="video/webM"/>
<source src="video/movie.mp4" type="video/mp4"/>
</video>
```

# Ajouter un fichier audio à un site web

134

- Il existe trois moyens d'insérer une vidéo : utiliser un **service audio** dédié (par exemple soundcloud), utiliser des **plugins** (Flash ou Silverlight) ou utiliser la balise HTML 5 audio
- Le format le plus populaire est le format MP3

# La balise audio

135

- L'ajout d'un fichier audio se fait en ajoutant la balise **audio**
- L'attribut **src** indique la source (url) du fichier audio
- L'attribut **preload** indique le pré-chargement. La valeur **none** indique que l'audio n'est pas chargée quand la page est chargée. La valeur **metadata** indique que seulement les métadonnées sont chargées. La valeur **auto** indique que l'audio est chargée avec la page.
- L'attribut **controls** indique que les contrôles de lecture sont affichés
- L'attribut **autoplay** indique que l'audio se lance dès qu'elle est chargée
- L'attribut **loop** indique que l'audio tourne en boucle

# La balise audio - Exemple

136

- Ci-dessous un exemple de la balise **audio**

```
<audio src="audio/yann.mp3" controls autoplay><p>Lecteur audio</p></audio>
```



# Utiliser plusieurs formats audio

137

Comme pour la balise **video**, en utilisant source, on peut mettre plusieurs sources audio pour le même lecteur en utilisant la même balise source

```
<audio controls autoplay="">
<source src="audio/yann.theora.ogv" />
<source src="audio/yann.mp3" />
<p>
Lecteur audio</p></audio>
```

# Exercice

138

- Dans sa page personnelle, ajouter la vidéo d'un évènement ou d'un moment qui a son importance (utiliser HTML 5)
- Prendre en charge plusieurs formats différents

# Section 10 : Nouveautés HTML 5

# Introduction

140

- HTML 5 introduit un certain nombre de balises et d'attributs
- HTML 5 met l'accent sur l'aspect sémantique des éléments
- Au moment d'écrire ces diapos, les navigateurs implémentent certains éléments HTML 5 différemment (parfois pas implémentés)

# Les articles (HTML5)

141

- La balise **article** représente un contenu indépendant
- Ça peut être un post de blog, un article, une contribution,...

```
<article><h1>Les langages de programmation</h1>
<p>Il y a dans la vie de nombreux langages de programmation tels que C#, PHP ou C++</p>
</article>
```

## Les langages de programmation

Il y a dans la vie de nombreux langages de programmation tels que C#, PHP ou C++

# Les sections(HTML5)

- La balise **section** représente une section (un chapitre, entête, pied,...)
- Une section peut subdiviser un article

```
<article>
<section>
<h1>Comment devenir programmeur</h1>
<p>Il faut aimer l'informatique et surtout le développement</p>
</section>
<section>
<h1>Quel langage choisir ?</h1>
<p>En fonction de ses moyens, son feeling et son contexte !</p>
</section>
</article>
```

## Comment devenir programmeur

Il faut aimer l'informatique et surtout le développement

## Quel langage choisir ?

En fonction de ses moyens, son feeling et son contexte !

# Les éléments de gestion de disposition (HTML5)

143

- Avant HTML 5 , on utilisait uniquement les div pour gérer la disposition
- HTML 5 introduit de nouveaux éléments à connotation sémantique définissant la disposition d'un document, d'un article ou d'une page. Il s'agit de **aside**, **nav**, **header** et **footer**
- La balise **header** représente l'entête et **footer** le pied
- La balise **nav** représente une zone de navigation qui contient des liens menant vers des endroits du site
- La balise **aside** représente un contenu à côté du contenu principal mais qui a une certaine relation avec

# Les éléments de gestion de disposition (HTML5)

## - Exemple

- Ci-dessous un exemple :

```
<header>
<h1>Mon fichier</h1>
</header>
<nav>
<p>
Accueil
</p>
<p>
Contact
</p>
</nav>
<div>
</div>
<footer>
<p>
© Copyright by Amine
</p>
</footer>
```

## Mon fichier

[Accueil](#)

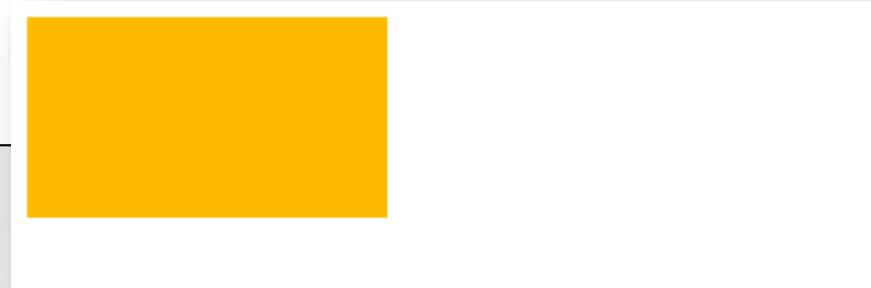
[Contact](#)

© Copyright by Amine

# Dessiner des graphes (HTML5)

- La balise **canvas** est utilisée pour faire des dessins en utilisant un langage de script (Javascript)

```
<canvas id="myCanvas"></canvas>
<script type="text/javascript">
var canvas = document.getElementById('myCanvas');
var ctx = canvas.getContext('2d');
ctx.fillStyle = '#FFBB00';
ctx.fillRect(0, 0, 80, 100);
</script>
```



# Incorporer des éléments (HTML5)

146

- La balise **embed** est utilisée pour incorporer des éléments externes (par exemple, des animations flash)
- L'attribut **src** indique l'url de l'élément externe

```
<body>
<embed src="swf/speakers.swf"></embed>
</body>
```



# Montrer la progression (HTML5)

147

- La balise **meter** utilisée pour montrer la progression des actions
- L'attribut **value** indique la progression en cours, **min** et **max** indique l'intervalle de progression

```
<meter value="2" min="0" max="10">
```

Vient juste de commencer

```
</meter>
```

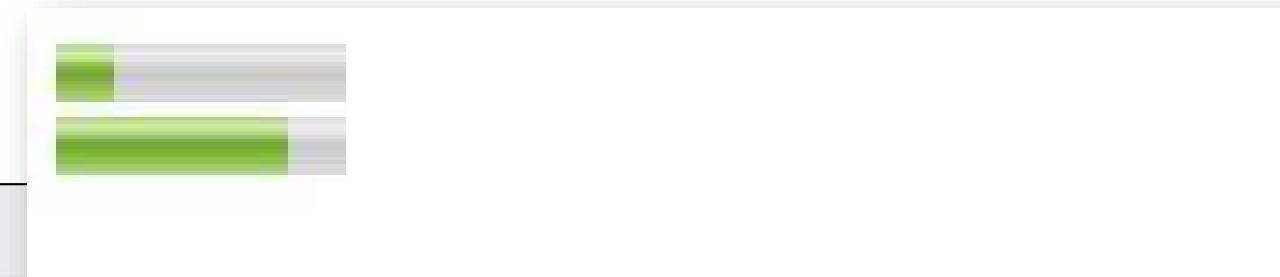
```


```

```
<meter value="0.8">
```

Bientôt fini

```
</meter>
```



# Exercice

148

- Refaire la disposition de la page personnelle en utilisant les balises HTML 5

# Bibliographie

149

- « JPEG, GIF, PNG... quel format d'image choisir pour le web ? »,  
<http://www.infocom21.net/spip.php?article138>
- « Introduction à la balise video de HTML5 »  
<http://www.alsacreations.com/article/lire/1125-introduction-balise-video-html5-mp4-h264-webm-ogg-theora.html>
- « Caractères spéciaux HTML », Comment ça marche,  
<http://www.commentcamarche.net/contents/html/htmlcarac.php3>