

Framework et technologies de développement

Mme. Sabrina BOUSSEMA

Faculté de sciences Monastir



Framework ?

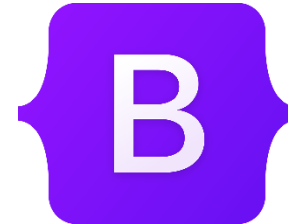
Front-End

Back-End



Plan

- Aborder les principaux frameworks et technologies utilisées dans le développement web.



Initiation au web et au HTML 5



Introduction

- Une brève histoire du WEB
- Le consortium W3C et le groupe WhatWG
- Le fonctionnement du WEB

Qu'est ce que HTML 5 ?

- HTML 5 : un ensemble de technologies (HTML 5, CSS 3, JavaScript)
- Son support dans les navigateurs

Une introduction au langage HTML

- Premier document HTML 5
- Le doctype, la validation W3C et le CSS
- Un document html avec le CSS et le JavaScript

Les grandes nouveautés HTML 5

- Les balises sémantiques
- Les nouveautés CSS 3
- Les balises multimédias (audio et vidéo)
- Les nouveaux éléments de formulaire

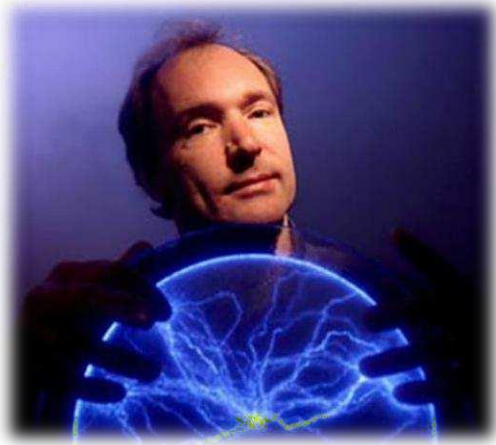
Conclusion

- Perspectives



A l'origine du WEB : Tim berners-Lee

- L'initiateur du premier site **internet** et du premier **navigateur** (WWW) en 1990 est **Tim berners-Lee**



- Il est à l'origine des 3 technologies principales du WEB : **Les adresses web (URL)**, le protocole **HTTP** et le **HTML**
- Il est le fondateur et le président du **W3C** depuis 1994.

W3C : **W**orld **W**ide **W**eb **C**onsortuim
HTTP : **H**yperText **T**ransfer **P**rotocol
URL : **U**nified **R**essource **L**ocator

Créations de nouvelles technologies

- 1993 : Le navigateur **Mosaic** développé pour **Sun** intègre la balise *img* pour l'intégration des images
- Une partie de l'équipe développe l'année suivante **Netscape Navigator**
- Il devient par la suite nécessaire de **standardiser** les langages du WEB -> Création du **W3C** par Tim
- A partir de 1995, tout s'accélère : création de **Javascript**, de l'**HTML 2.0**, et de la 1^{ère} version d'**internet explorer**.
- S'en suit en 1996, un nouveau standard nommé : **CSS**

- Le **W3C** est un organisme de normalisation **indépendant** qui émet des **standards libres** pour le WEB et notamment le **HTML, CSS, SVG,...**



- Ce dernier est administré par trois entités :

Le **MIT** (États-Unis)

L'université **Keio** (Japon)

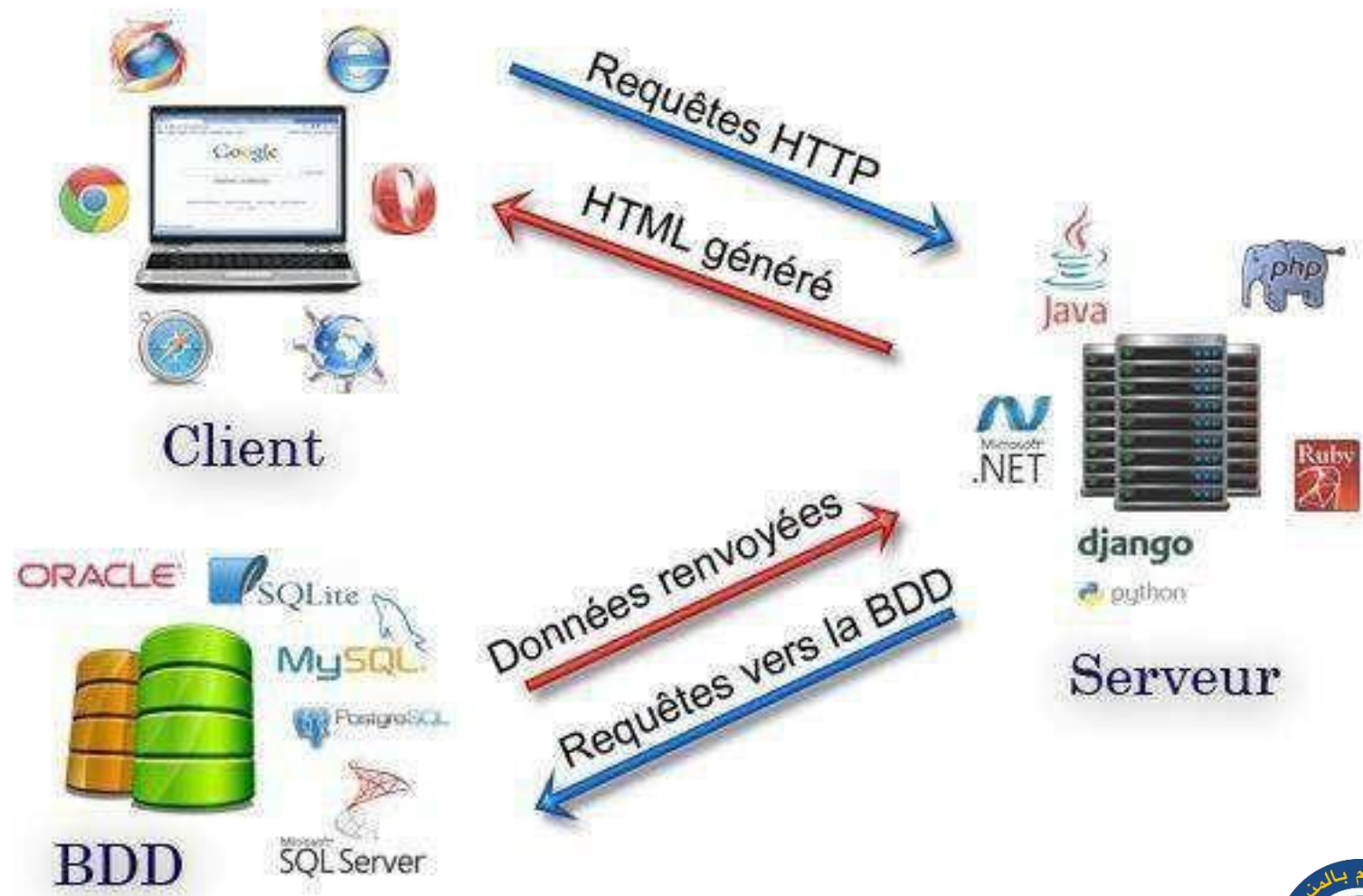
L'**Ercim** (Europe)

- Le **W3C** standardise également de nombreuses autres technologies comme **MathML, RDF, Soap, SVG, Smil, PNG** et autres variations autour de **XML**

- Le **W3C** tente de persévérer vers la voie du **XML** (création du XHTML)
- Le **WhatWG** souhaite améliorer HTML et débute son travail en 2004 avec **Ian Hickson** qui sera l'éditeur officiel du HTML 5
- Le **HTML5** est soutenu devant le **W3C** en 2007. Il était **retenu** et son premier brouillon a été publié l'année suivante.



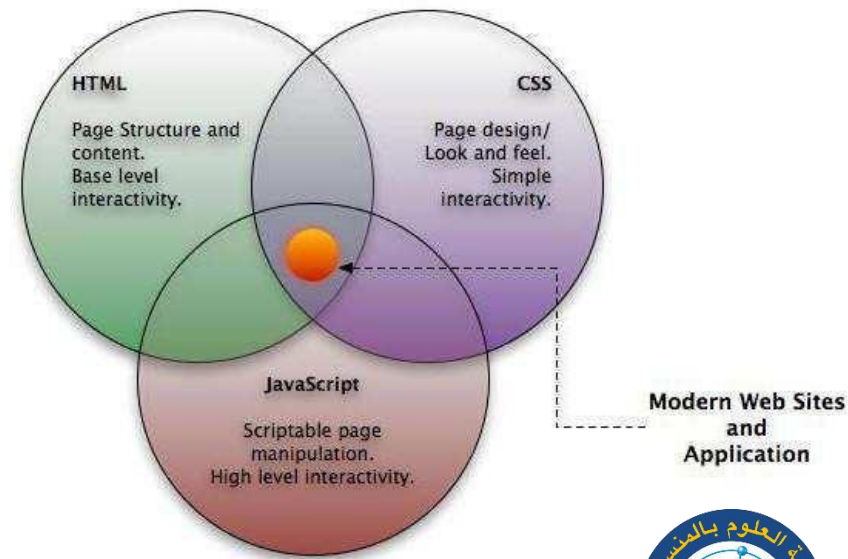
Le fonctionnement du WEB



- Le **HTML 5** est fondamentalement le langage HTML dans sa 5^{ème} version...
- ...Mais il regroupe, par un abus de langage justifié, le **HTML 5**, le **CSS 3** et le **JavaScript**. Le **HTML 5** est donc la pierre angulaire de cet édifice.

On va donc avoir :

- **HTML** -> fond du document
- **CSS** -> forme du document
- **JavaScript** -> dynamisme du document



Les navigateurs et le support de HTML 5

- Distinguons les **navigateurs** et les **moteurs de rendu**.
- Chaque navigateur intègre **un** moteur de rendu qui peut être commun à **plusieurs** autres navigateurs.
- Un petit tableau des correspondances :

Moteur	Navigateurs	Développement
Trident	Internet Explorer	Microsoft
Webkit	Google chrome, apple safari, android, epiphany	Apple, Google, KDE, Nokia, RIM, Palm et autres
Gecko	Mozilla firefox, Camino, K-Meleon, Galeon, Flock	Fondation Mozilla
Presto	Opera, opera mobile, opera Wii	Opera Software
KHTML	Konqueror	KDE

Introduction au langage HTML

- HTML (Hypertext Markup Language) est le format de données structurant une page web.
- HTML est initialement dérivé du SGML dont la première publication date de 1986.
- Les versions de HTML :
 - HTML 1 (1991), HTML2(1994)
 - HTML3(1996): la base des autres versions et définit par W3C.
 - HTML 4 (1998) : la version la plus répandue
 - XHTML 1.0 (2000) : reformulation de HTML en tant qu'une application XML. XHTML 2.0(2001-2009)
 - HTML 5.1 (2016) : évolution de HTML 4.0.1

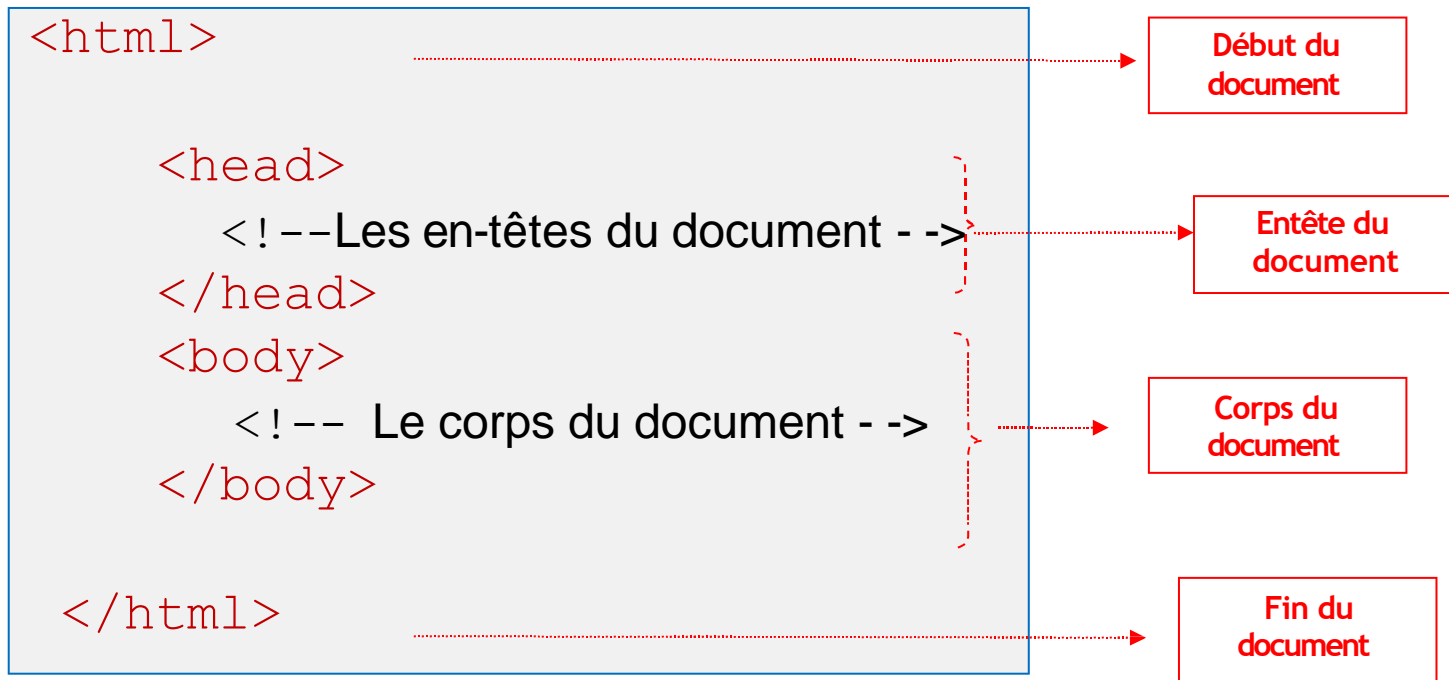
SGML : (Standard Generalized Markup Language)

XHTML : (eXtensible Hypertext Markup Language)



Structure de document HTML

- Exemple de document HTML avec les premières balises `<html>`, `<head>` et `<body>`



- C'est un langage de **balisage** : `<balise>Contenu</balise>`

- Le **doctype** est le préambule d'une page HTML, il spécifie le type de document.

- ## Doctype HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

- ## Doctype XHTML 1.1

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

- ## Doctype HTML 5

```
<!DOCTYPE html>
```



L'en-tête de document

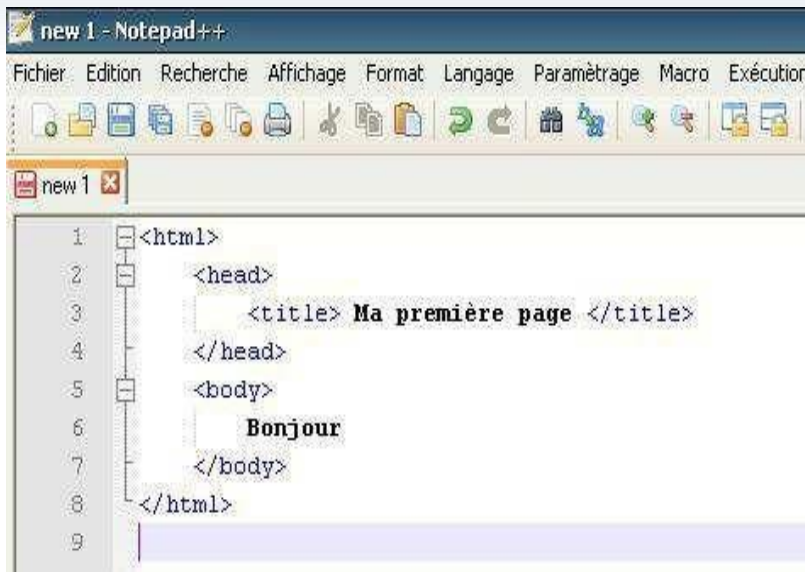
- Représente des informations générales sur la page
- `<title> </title>` : titre de la page
- `<meta >` : informations supplémentaires-mots clés, type de contenu, auteur, encodage utilisé- utilisées par les navigateurs et les moteurs de recherche
- `<link />` : lien vers une source utilisée par la page
- `<script></script>` : référence à des scripts en JS

Le corps de document

- Représente le contenu de la page web. qui sera affiché à l'écran.
- Contient des balises qui seront affichées dans le navigateur .
- Le style de la page doit être mis en un fichier extérieur

Environnement de travail

Processus de création d'une page

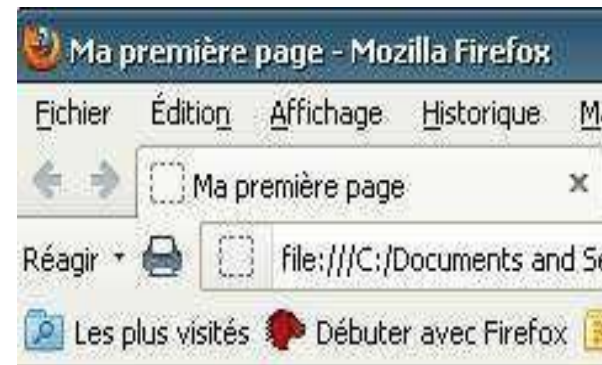


```
1 <html>
2   <head>
3     <title> Ma première page </title>
4   </head>
5   <body>
6     Bonjour
7   </body>
8 </html>
9
```

Editeur de texte
(pour écrire le code HTML)



Enregistrer le fichier
du code HTML



Navigateur
(pour visualiser la page web)

- Code HTML 5 valide W3C (à tester sur <http://validator.w3.org>)

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML5</title>
    <meta charset="UTF-8">
  </head>
  <body>
    Vive HTML 5 ! :D
  </body>
</html>
```

- Code CSS 3 valide W3C (à tester sur <http://jigsaw.w3.org/css-validator/>)

```
@CHARSET "UTF-8"
p {
  font-size : 10px;
  color : 'blue'
}
```

- Liens valide W3C (à tester sur <http://validator.w3.org/checklink/>)

- Le code **CSS** va mettre en forme notre document **HTML**

- **Syntaxe générale :**

```
sélecteur {  
    propriété : valeur;  
}
```

- **Exemple :**

```
p {  
    font-size : 10px;  
    color : 'blue'  
}
```

- Dans notre exemple, le contenu texte de toutes nos balises **<p>** auront une taille de 10 px et une couleur grise.

<p>Le CSS !!!**</p>**



Le CSS !!!

Mise en forme avec CSS

HTML
(pas de CSS)



HTML + CSS



- Association des 3 langages piliers du WEB :

Note HTML 5

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML5</title>
    <meta charset="UTF-8">
    <link rel="stylesheet" type="text/css" href="styles.css">
    <script src="javascript.js"></script>
  </head>
  <body>
    HTML5 !!!
  </body>
</html>
```

Inclusion de notre
fichier CSS

Inclusion de notre
fichier JavaScript



Les grandes nouveautés de HTML 5

- Les nouvelles balises sémantiques (**<header>**, **<footer>** ...)
- Les principales nouveautés **CSS 3**
- Les balises multimédias (**<audio>** et **<vidéo>**)
- La balise **<canvas>** pour un affichage dynamique
- Le stockage **hors ligne**
- La **géolocalisation**
- Une gestion des **formulaires** plus poussée

Les nouvelles balises sémantiques

`<header>`

`<nav>`

`<article>`

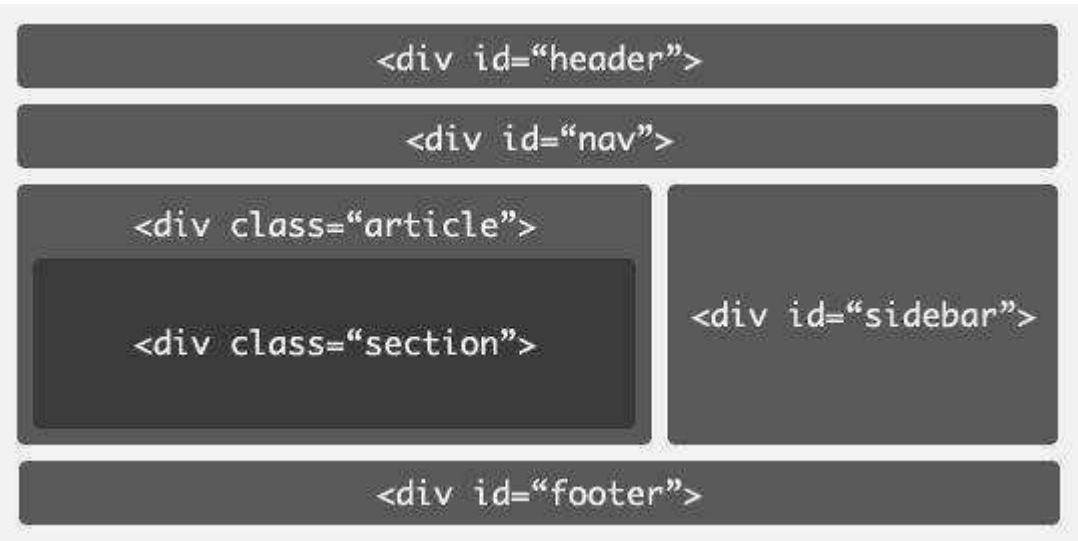
`<section>`

`<aside>`

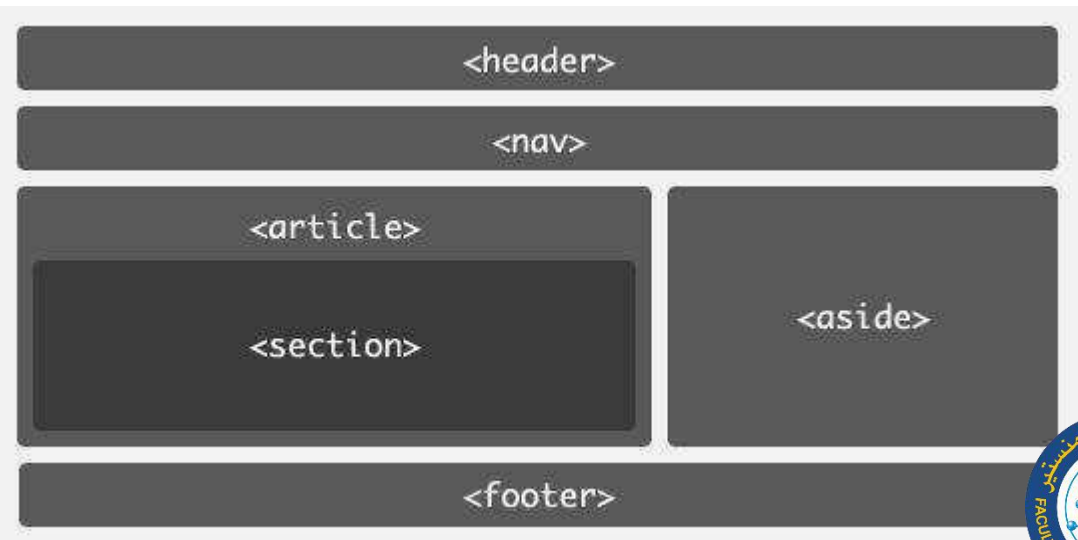
`<footer>`

Les nouvelles balises sémantiques

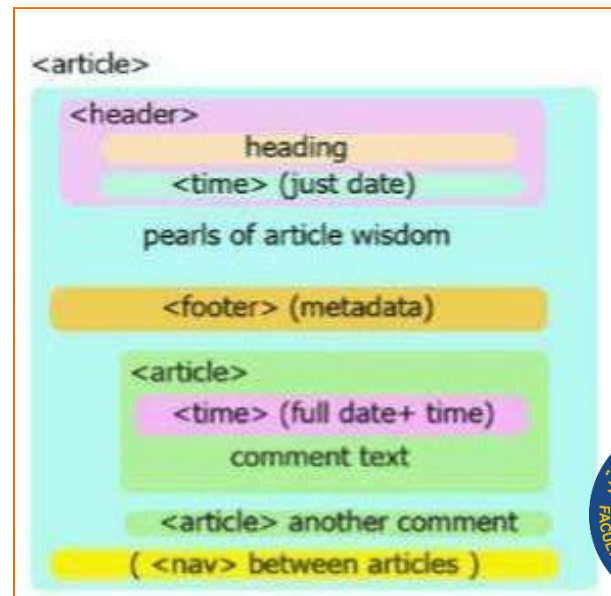
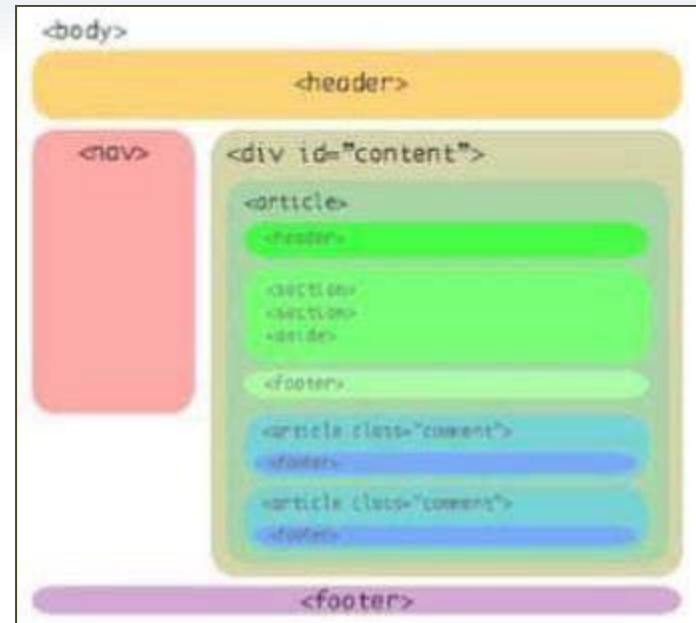
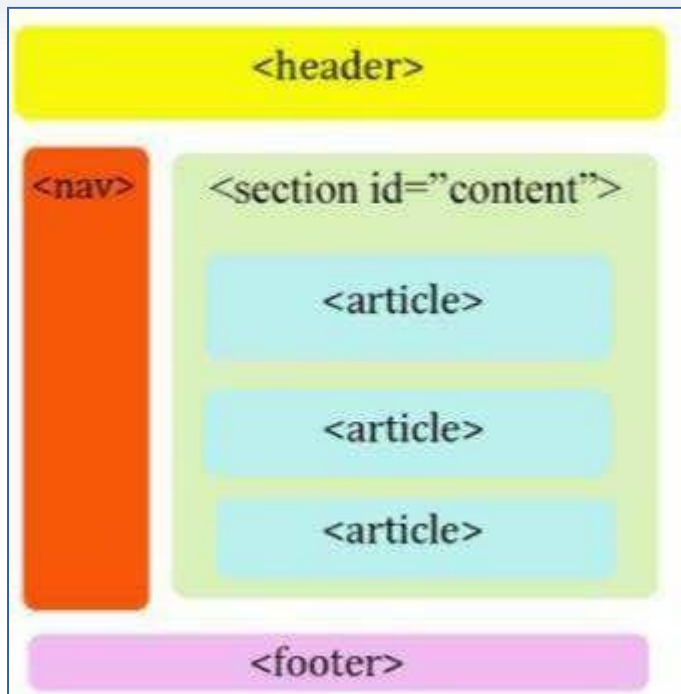
AVANT



APRÈS



Exemples de structures



➤ **<header>**

- Selon la spécification HTML5, l'élément header représente :
 - Un groupe d'aide de navigation ou d'introduction
- Il contient de façon générale les headings (les éléments h1 à h6 ou l'élément hgroup)
 - Il peut aussi contenir d'autres éléments, comme une table de matières, un formulaire de recherche, ou des logos.

➤ **<footer>**

- Représente le bas de la section à laquelle il s'applique.
- Contient typiquement une information sur sa section comme son auteur, des liens vers des documents liés, les données de copyright et autres données du même type.

➤ **<nav>**

- permet de regrouper une liste de liens d navigation
- Convient pour la navigation dans le site
- Le menu est réalisé sous forme de liste à puces à l'intérieur de la balise <nav>.

➤ **<aside>**

- Elle est conçue pour contenir des informations supplémentaires au document que l'on visualise (note, astuce, un extrait, etc).
- Ces informations sont placées sur le côté (bien que ce n'est pas une obligation).

➤ **<section>**

- Une section regroupe des contenus en fonction de leur thématique.
- Elle distingue un bloc logique de contenu.
- Elle peut être elle-même se subdiviser en plusieurs sections et être constitué d'un en-tête, d'un pied et d'une navigation.

➤ **<article>**

- Représente une section indépendante d'un document, d'une page ou d'un site.
- Convient pour du contenu comme : des nouvelles, des articles de blog, des messages de forum ou des commentaires individuels.
- C'est une partie de la page qui pourrait ainsi être reprise sur un autre site.

Les nouvelles balises sémantiques



```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

```
<title>Le Site Web</title>
```

```
</head>
```

```
<body>
```

```
<header>
```

```
<h1>Thème 1</h1>
```

```
<h2>Carnets de voyage</h2>
```

```
</header>
```

```
<nav>
```

```
<ul>
```

```
<li><a href="#">Accueil</a></li>
```

```
<li><a href="#">Blog</a></li>
```

```
<li><a href="#">CV</a></li>
```

```
</ul>
```

```
</nav>
```



```
<section>
```

```
<aside>
```

```
<h1>News</h1>
```

```
<p>Dernières Actualités</p>
```

```
</aside>
```

```
<article>
```

```
<h1>un grand voyage</h1>
```

```
<p>(texte de l'article)</p>
```

```
</article>
```

```
</section>
```

```
<footer>
```

```
<p>Copyright Tous droits réservés
```

```
<br />
```

```
<a href="#">contact !</a>
```

```
</p>
```

```
</footer>
```

```
</body>
```

```
</html>
```

Syntaxe :

`<balise> Contenu </balise>`

- une balise représente un élément de la page,
-sont lues séquentiellement,
-sont affichées au fur et à mesure par le navigateur.
- 2 types de balises :
 - **Balises en paire** `<balise>.....</balise>`
 - **Balises auto-fermante** `</balise>`
- Une balise admet des attributs ayant des valeurs.
- Ils viennent les compléter pour donner des informations supplémentaires.

```
<balise attribut1='valeur1' attribut2='valeur2' ...>  
    contenu  
</balise>
```

```
<balise attribut1='valeur1' attribut2='valeur2' ..... />
```

Élément bloc

- Il est précédé et suivi d'un saut de ligne.
- Il forme une boîte et inclut d'autres éléments bloc (<p>, <table>, , <div>, <tfooter>, ...) et des éléments en ligne (<q>, <sup>, <sub>, <kbd> <small>,).

Élément en ligne

- appelé aussi élément *de niveau de texte*
- Il s'insère dans le fil de texte.
- ne peut contenir que du texte ou d'autres éléments en ligne.
 - **éléments remplacées** : les dimensions peuvent être définies (image, zone de saisie d'un formulaire,...).
 - **éléments non remplacés** : la taille est fonction de leur contenu (, , , <a>, ...)

Principaux éléments HTML bloc:

- `<div>`: Sert de boîte « conteneur »
- `<H1>` à `</H6>`: Titres et sous titres
- `<p>`: Créer des Paragraphes
- ``, ``, `<dl>` et ``, `<dt>`, `<dd>`: Créer des listes à puces, numérotée, de définition et ses éléments
- `<blockquote>`: Extrait de citation apparaît en retrait
- `<tfooter>`, `<thead>`, `<nav>`, `<section>`, `<article>`: Les nouvelles balises sémantiques introduite par HTML 5.
- `<table>`: Créer un tableau
- `<adress>`: Créer un paragraphe contenant une adresse

Principaux éléments HTML en ligne :

- ``, ``, Mise en exergue ou en évidence ou
- `<pre>` : préformaté
- `<abbr>`: Extrait ou référence à une autre source
- `<dfn>` : Terme encadré a une définition
- `<code>`: Portion de code ou de texte à entrer
- `` : Délimiter une partie de texte
- ``, Insérer une image
- `<object>` : Insérer un objet multimédia
- `<a>`: Insérer un lien
- `<q>`, `<cite>` : Extrait de citation
- `<ins>`, `` : Texte remplacé et texte supprimé
- `<var>` : Nom d'une variable
- `<mark>` : Marquer des mots, des expressions ou des phrases

Liste à puces

```
<ul>
  <li> élément 1 </li>
  <li> élément 1 </li>
  <li> élément 1 </li>
</ul>
```

Liste numérotée

```
<ol>
  <li> élément 1 </li>
  <li> élément 1 </li>
  <li> élément 1 </li>
</ol>
```

Liste de définition

```
<dl>
  <dt> terme </dt>
  <dd> définition associée </dd>
  <dt> terme </dt>
  <dd> définition associée </dd>
</dl>
```

Les tableaux simples

- La balise `<table>` `</table>` indiquent le début et la fin d'un tableau.
- La balise `<tr>` `</tr>` déclare une ligne.
- La balise `<td>` `</td>` déclare une cellule normale.
- La balise `<th>` `</th>` déclare une cellule titre.
- La balise `<caption>` `</caption>` ajoute un titre au tableau.
- Les attributs `rowspan` et `colspan` fusionnent les cellules.

En HTML, un tableau se construit ligne par ligne. Dans chaque ligne (`<tr>`), on indique le contenu des différentes cellules (`<td>` ou `<th>`).

Les tableaux simples

La liste des étudiants

td
(cellules)

th
(cellules titres)

tr
(ligne)

Nom	Age	Ville
Hajer	21 ans	Tunis
Ahmad	22 ans	Sfax
Malek	20 ans	Gabes
Amina	19 ans	Djerba

Les tableaux structurés

– Le tableau est divisé en 3 parties :

- **En-tête** du tableau
- **Corps** du tableau
- **Pied** du tableau

```
<table>
  <caption>
    titre du tableau
  </caption>
  <thead>
    en-tête de table
  </thead>
  <tfoot>
    pied de table
  </tfoot>
  <tbody>
    corps de table
  </tbody>
</table>
```

Les tableaux structurés

thead

(entête du tableau)

La liste des étudiants

tbody
(corps du
tableau)

Nom	Age	Ville
Hajer	21 ans	Tunis
Ahmad	22 ans	Sfax
Malek	20 ans	Gabes
Amina	19 ans	Djerba
Nom	Age	Ville

tfoot

(pied du tableau)

- Pour définir une mise en forme spécifique aux différentes colonnes d'un tableau, il suffit d'utiliser autant de balises `<col...../>` réuni dans une balise `<colgroup>`.

```
<table>
  <colgroup>
    <col class='colonne 1' />
    <col class='colonne 2' />
    <col class='colonne 3' />
  </colgroup>
  .....contenu de tableau.....
</table>
```

Remarque : la mise en forme de chaque colonne (largeur, couleur de fond, ...) est alors effectuée dans une feuille de style, à partir de classes associées à chacune des colonnes.

- La balise `` permet d'insérer une image.
- Les attributs suivants sont obligatoires :
 - **src** : l'URI où se trouve l'image
 - **alt** : courte description de l'image

Exemple :

```
<img src = "logo.jpg" alt = "le logo de site" />
```

- Autres attributs sont optionnels :
 - **width** : largeur de l'image
 - **height** : hauteur de l'image

Remarque : En spécifiant la taille de l'image (**width** et **height**), on accélère le chargement.

Les liens

- La balise `..... ` permet d'insérer un lien.
- Le contenu de l'élément est celui qui sera affiché en tant que lien
- L'attribut **href** contient l'URL vers laquelle le lien pointe.

- URL : `http://www.google.com`
- URL (mail) : `mailto:alexandre.pauchet@insa-rouen.fr`
- Fichier local avec chemin relatif : `./dossier/autre_page.html`
- Fichier local avec chemin absolu : `/www/dossier/autre_page.html`

- L'attribut **title** affiche une bulle d'aide lorsqu'on pointe sur le lien.

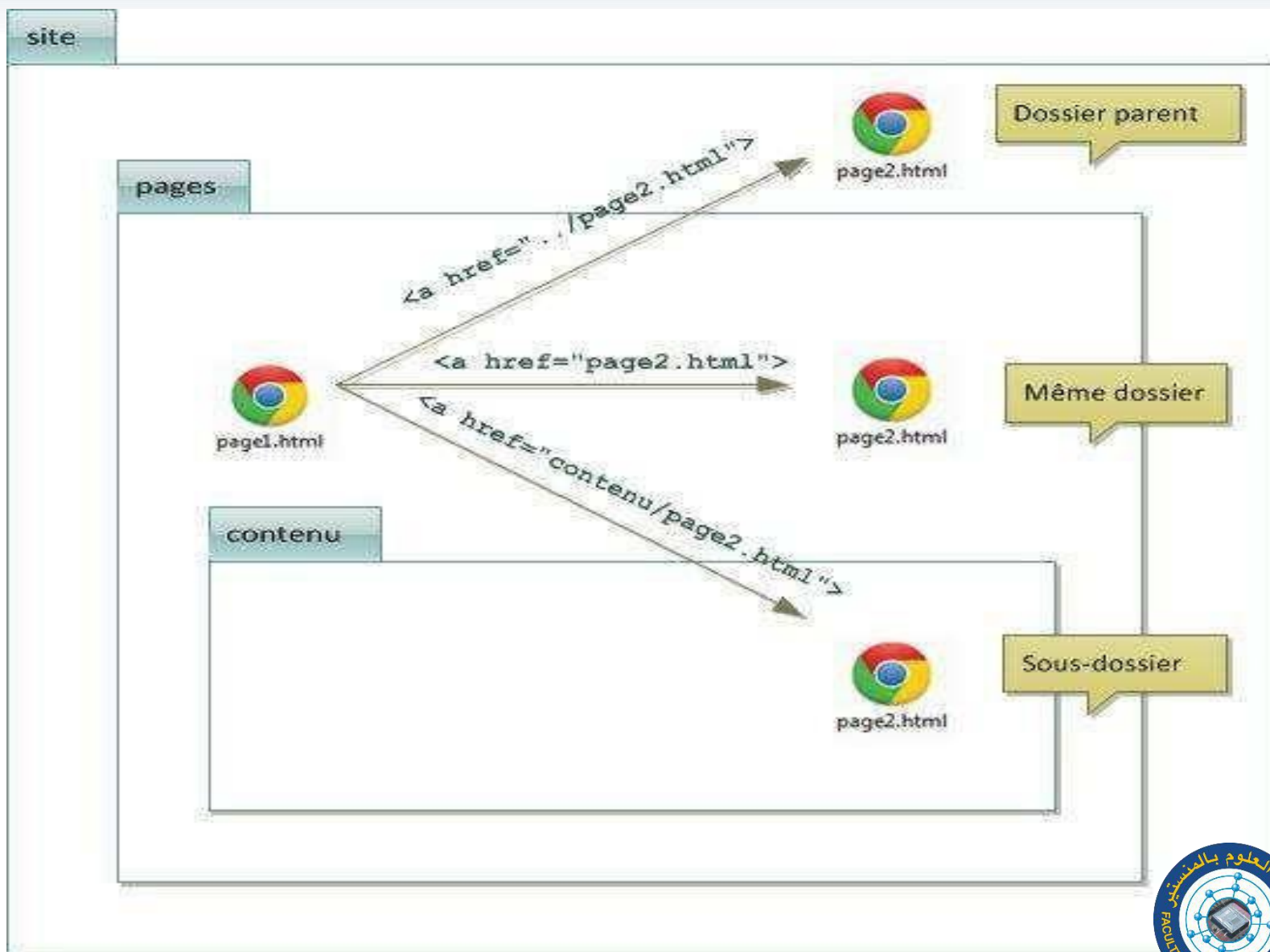
```
<a href = "page.html" title= "Cliquez ici pour aller à la page suivante" > Suivant </a>
```

Suivant



Cliquez ici pour aller à la page suivante

Liens vers une autre page du même site



- Il s'agit d'un lien **interne**.
- Ce lien permet de se déplacer à l'intérieur du fichier Web (exp: aller de bas de page vers le haut,).
- Pour créer un ancre, 2 étapes nécessaires :
 - définir la cible d'un lien (l'ancre).
 - associer ce même lien à cette ancre.

```
<a id="haut"></a>
```

ancre

```
<a href="#haut">Haut de la page</a>
```

lien

- Lien vers une ancre situé dans une autre page :

```
<a href="page2.html#haut">Haut de la page2</a>
```

Les nouveaux éléments HTML 5

`<figure >` et `<figcaption >`

`<progress >`

`<details >`

`<meter >`

`<time >`

L'élément `<figure>`

- La balise `<figure></figure>` permet d'insérer une image ou une figure.
- La balise `<figcaption></figcaption>` permet d'ajouter une légende à la figure.

```
<figure >  
  <img src = " logo.jpg " alt = " le logo de site " />  
  <figcaption> Le logo de site </figcaption>  
</figure>
```

L'élément `<time>`

- La balise `<time>` `</time>` permet d'insérer une date par exemple dans un article.
- Les attributs possibles :
 - **datetime** est formaté selon l'année, le mois et le jour.
 - **pubdate** publie le contenu.

```
<p>
```

Par David, posté le

```
<time datetime = " 2012-09-05 " pubdate>
```

05 Septembre 2012

```
</time>
```

```
</p>
```

Nouveaux Éléments HTML 5

L'élément `<progress>`

- La balise `<progress></progress>` permet d'afficher l'état d'avancement d'une tâche.
- Les attributs sont :
 - **value** : indique la valeur courante
 - **max** : indique la valeur maximale.

```
<p> <- -! Progression inconnue -->
```

```
<progress id = " progressBar1 " max= " 100 " >
```

```
<span> 0 </span>% </progress>
```

```
</p>
```

```
<p> <- -! Progression connue -->
```

```
<progress id = " progressBar2 " max= " 100 " value= " 66.6 " >
```

```
<span> 66.6 </span>% </progress>
```

```
</p>
```



L'élément `<meter>`

- La balise `<meter></meter>` permet de représenter une distance, un pourcentage ou toute autre mesure à condition de connaître les bornes.

```
<p> Pourcentage des votants :  
  <meter value= " 0.75 " > 75% </meter>  
</p>  
<p> Nombre de tickets vendus :  
  <meter min= " 0 " max= " 100 " value= " 75 " > 75% </meter>  
</p>
```

- Autres attributs :
 - **optimum** : valeur optimale, moyenne de low e high par défaut.
 - **low** : limite basse (min par défaut), n'est pas < **min**.
 - **high** : limite haute (max par défaut), n'est pas > **max**.
 - **title** : titre de l'élément pour indiquer l'unité.

L'élément `<details>`

- La balise `<details></details>` donne la possibilité d'afficher sur demande des précisions sur un texte.
- Ces précisions sont masquées par défaut. Le clic sur le titre fait apparaître les détails.
- L'attribut `open` permet d'afficher directement les détails en question.
 - `open = " true "` : détails affichées
 - `open = " false "` : détails masquées

`<details>` Pourcentage des votants :

`<summary>` **Le W3C** `</summary>`

`<p>` W3C signifie World Wide Web Consortium. `</p>`

`<p>` Il a été créé par Tim Berners-Lee en Octobre 1994. `</p>`

`</details>`



Les éléments de formulaire

Tarif HT €

TVA 19.6 %

Prix TTC : €

Formulaire.html

Exemple de formulaire

Nom : Prénom :

☐ Use checkbox ☐ Homme ☐ Femme

Fichier : Choisir un fichier... Aucun fichier choisi

toto

les autres...

Raconte-moi une histoire...

- La balise `<form>..... </form>` déclare un formulaire.
- Les attributs :
 - **action** : URL spécifiant le traitement des données (mail, script,...).
 - **method** : méthode d'envoi des données (GET ou POST).
 - **id** : identifiant pour un traitement (js ou autre) ou associer un sty
- Autres nouveaux attributs :
 - **autocomplete** : lancer la saisie semi-automatique
 - **novalidate** : si présent le formulaire n'est pas validé lorsqu'il est soumis.
- La méthode d'envoi des données :
 - **method = 'POST'** : envoi masqué des données
 - **method = 'GET'** : envoi visible des données dans la barre adresse de navigateur.

Exemple de formulaire

Formulaire.html

Exemple de formulaire

Nom : Prénom :

☐ Une checkbox ☐ Homme ☐ Femme

Fichier : Aucun fichier choisi

toto
titi
les autres ▾

Raconte-moi une histoire...



Regroupement de parties de formulaire

- lier certains éléments de formulaire dans un regroupement permet d'améliorer la présentation.
- Balise utilisée : `<fieldset>..... </fieldset>`
- Ajouter une légende : `<legend>..... </legend>`

```
<fieldset>
  <legend> Informations personnelles </legend>
  <label for = "nom ">Nom : </label>
  <input type= "text " name= "nom " id= "nom " /> <br/>
  <label for= "prenom ">Prénom : </label>
  <input type= "text " name = "prenom " id = "prenom " />
</fieldset>
```

Zone de texte simple

- La balise `<input />` correspond à saisie d'informations
- Il permet de définir plusieurs éléments selon le type (nom, prénom, mot de passe, numéro de téléphone, etc.)
- ...contient les attributs suivants :
 - **type** : spécifie le type de l'élément
 - **name** : donne un nom à cet élément
 - **value** : donne une valeur

```
<input type="text" name="champ" size="10" value="texte" />
```

Les nouveaux types `<input />`

```
type = "tel"
```

```
<label for = "champ1 ">Tel : </label>
<input type="tel" id= " champ1" required />
```

- Le contrôle sur la saisie est grâce à l'attribut **pattern** ainsi qu'un contrôle côté serveur.

```
<label for = "champ2 ">Tel : </label>
<input type="tel" id= " champ2" pattern = " ^((\+\d{1,3}(-|
)?\(?\d\)?(-| )?\d{1,5})|(\(?\d{2,6}\)?))(-| )?\d{3,4})(-| )?\d{4})(( x|
ext)\d{1,5}){0,1}$ " required />
```


Les nouveaux types `<input />`

```
type = " URL "
```

Tous les types d'URL sont admis : (ftp:// , mailto: , http:// ,).

```
<label for = "champ1 ">Votre site web : </label>  
<input type= " URL " id= " champ1" value= " http://" required />
```

```
type = " email "
```

```
<label for = "champ1 ">Votre e-mail: </label>  
<input type= " email " id= " champ1" required />
```

- Ce champ attend au minimum un caractère (non accentué comprenant les séparateurs tirets ou underscore) suivi d'un @ et à son tour d'un caractère.

Les nouveaux types `<input />`

`type = "date"`

Le contenu est une date du calendrier sans précision de la localisation (timezone).



```
<label for = "champ1 ">Votre date: </label>
```

```
<input type= "date" id= "champ1" required />
```

`type = "datetime"`

Ce type permet de renseigner la date, l'heure et le fuseau horaire.



```
<label for = "champ1 ">Prochain RDV: </label>
```

```
<input type= "datetime" id= "champ1" name= "datetime" /> <br/>
```

```
<input type= "submit" value= "Valider" />
```

Les nouveaux types `<input />`

`type = " datetime-local "`

Le même rôle que **Datetime**, la seule différence est l'absence de précision du fuseau horaire.

```
<label for = "champ1 ">Votre date: </label>
```

```
<input type= " datetime-local " id= " champ1 " />
```

`type = " time"`

Ce type permet de renseigner une heure, avec plus ou moins de précision. Le format attendu est le même que pour le champ de type date, sans localisation (timezone).

```
<label for = "champ1 ">Heure de r&eacute;veil : </label>
```

```
<input type= " time" id= " champ1 " />
```

Les nouveaux types `<input />`

```
type = " week "
```

Le contenu permet de renseigner une semaine dans une année.

```
<label for = "champ1 ">Semaine : </label>
<input type= " week " id= " champ1 " name= " semaine " />
```

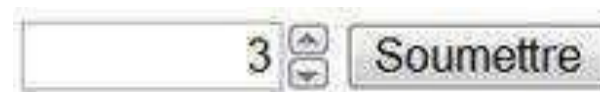
Select a week:

Semair	Lun	Mar	Mer	Jeu	Ven	Sam	Dim
5	30	31	1	2	3	4	5
6	6	7	8	9	10	11	12
7	13	14	15	16	17	18	19
8	20	21	22	23	24	25	26
9	27	28	29	1	2	3	4
10	5	6	7	8	9	10	11

Aujourd'hui

Les nouveaux types `<input />`

`type = " number "`

A screenshot of a web form. It features a numeric input field with the value '3' and two small arrows for incrementing and decrementing. To the right of the input field is a button labeled 'Soumettre'.

- Ce champ renseigne une valeur numérique.
- Il est alors transformé en une sorte de boîte permettant l'incréméntation et la décrémentation d'une valeur numérique initiale (0 par défaut), lorsque la prise en charge par le navigateur est complète.

```
<label for = "champ1 ">Code postale : </label>  
<input type= " number" id= " champ1 " name= " cp " />
```

- Attributs possibles : **step** (spécifier un pas d'incréméntation), **min** (minimum), **max** (maximum) et **value** (valeur lors de chargement de page).

Les nouveaux types `<input />`

`type = " range "`

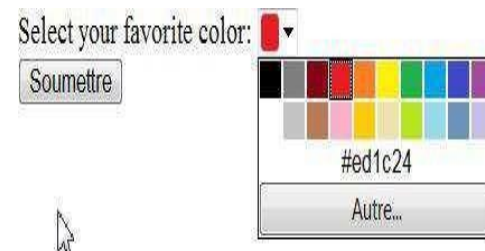


Ce champ propose un contenu évalué approximatif.

```
<label for = "champ1 ">Niveau : </label>
```

```
<input type= " range" id= " champ1 " name= " range " value= " 15 " max= " 50 " min= " 0 " step= " 5 " />
```

`type = " color "`



Ce type permet de transformer le champ en une palette de couleurs.


```
<label for = "champ1 ">Couleur : </label>
```

```
<input type= " color" id= " champ1 " name= " couleur" value= " #fad345 " />
```

La valeur attendue : couleur au format hexadécimal (un dièse suivi de 6 caractères alpha-numériques compris entre A et F, et 0 et 9).

Les nouveaux types `<input />`

`type = " search "`



Ce champ propose un champ pour saisir des mots clé dans la zone de recherche.

```
<label for = "marecherche ">Entrer le texte de recherche : </label>
<input type= " search" id= " recherche" placeholder= " rechercher" />
```

Google propose une autre fonctionnalité pour la recherche en se basant sur l'API Speech de HTML 5 : la saisie vocale!

```
<input type= " search" id= " recherche" placeholder= " rechercher"
x-webkit-speech = " x-webkit-speech " />
```

Les boutons radio

Les boutons radio sont utilisés lorsqu'il s'agit d'effectuer un choix exclusif dans une liste d'options.

```
<form name = "monform" >
  <label for = "h ">Home : </label>
  <input type= " radio" id= " h " name= " genre " value= " homme"
checked />
  <br/>
  <label for = "f ">Femme : </label>
  <input type= " radio" id= " f " name= " genre " value= " femme" />
</form />
```


Les boutons checkbox

Contrairement aux boutons radio, les cases à caucher permettent plusieurs choix ou aucun.

```
<form for = "monform" >
  <label for = "ch1 ">Football </label>
  <input type= " checkbox" id= " ch1 " name= " choix1 " />
  <br/>
  <label for = « ch2 ">Tennis </label>
  <input type= " checkbox" id= " ch2 " name= " choix2 " />
  <br/>
  <label for = "ch3 ">Natation </label>
  <input type= " checkbox" id= " ch3 " name= " choix3 " />
</form />
```

Les boutons

- Un élément `<input />` de type :
 - **Submit** : affiche un bouton et permet l'envoi des données du formulaire.
 - **image** : affiche une image et permet l'envoi des données du formulaire au serveur.
 - **reset** : affiche un bouton et permet de restaurer les valeurs par défaut des éléments du formulaire.
 - **file** : affiche un bouton permettant d'ouvrir une boîte de recherche de fichier.

```
<input type= "submit" name= "action" value= "Insérer " />
```

```
<input type= "file" name= "fichier" id= "lefichier " />
```

```
<input type= "image" src= "images/logo.jpg" value= "
```

```
Supprimer " name= "action" />
```

```
<input type= "reset" value= "Annuler " name= "supprimer" />
```

Zone de texte multilignes

- Un élément `<textarea>` permet de créer un champ texte sur plusieurs lignes.

```
<textarea rows= " 50" cols= " 50" >
```

Ce texte est éditable et sera envoyé lors du submit

```
</ textarea >
```

- Les attributs :
 - **cols** : détermine le nombre de colonnes
 - **rows** : détermine le nombre des lignes

Liste de sélection

- La balise `<select>` permet de définir une liste des valeurs.
- Les attributs :
 - **multiple** : sélectionner plusieurs éléments de la liste
 - **code** : si `>2` affiche un tableau, sinon un menu déroulant

```
<select name=" maliste" size=" 3" multiple=" multiple " >  
  <option size=" 1" > toto </option >  
  <option selected=" selected" value=" 2" > titi </option>  
  <optgroup label = " les autres " >  
    <option value=" 3" > tata </option >  
    <option value=" 3" > tutu </option >  
    <option value=" 3" > tete </option >  
  </optgroup >  
< /select>
```

Les étiquettes

- La balise `<label>` sert à nommer des champs.
- L'attribut : `for` indique champ décrit (attribut `id`).
- Utile sur les radios et checkbox : augmente la surface d'activation.

```
<label for = "champ1 ">Code postale : </label>  
<input type= " number" id= " champ1 " name= " cp " />
```

Nouveaux éléments de formulaire

- HTML 5 introduit de nouveaux éléments utiles aux formulaires : `<output>`, `<keygen/>`, `<datalist>`
- Ce qui permet plus de souplesse pour la création de l'interface utilisateur.
 - `<output>` : représente le résultat d'un calcul.
 - `<datalist>` : créer une liste de suggestions associée à un champ.
 - `<keygen>` : génère une clé sécurisé.

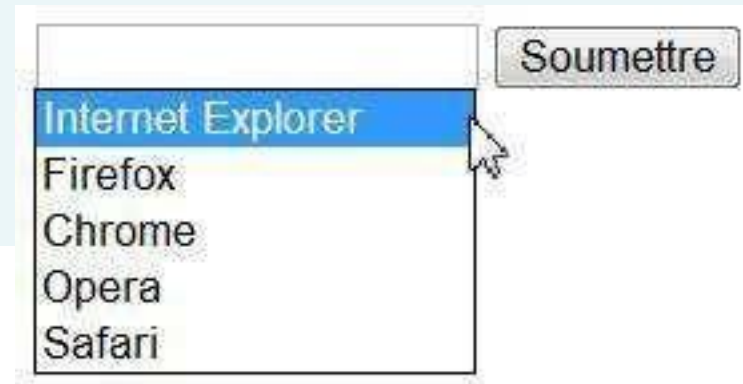
Elément **<output>**

```
<input type="number" value=" 2 " name=" nbre1 " >
*
<input type="number" value=" 3 " name=" nbre2" >
=
<output name=" opt " onforminput = " value =
    nbre1.value*nbre2.value">
</output>
```

2 x 3 = 6

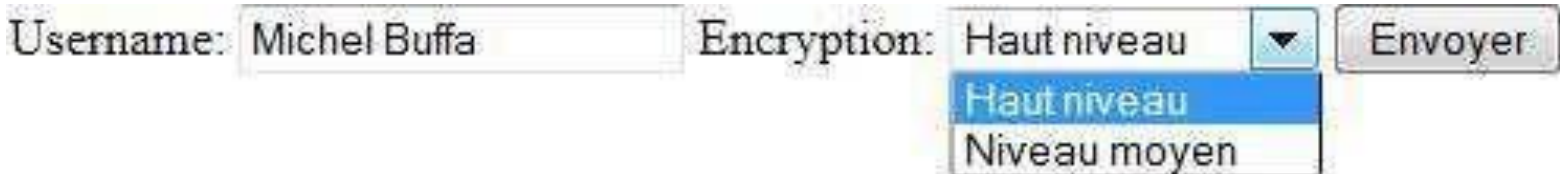
Elément **<datalist>**

```
<form action = " controleForm.php " method = " Get " >
  <input list= " listesuggt" name= " zonesaisie " />
  <datalist id="      listesuggt      "      >
    <option value="Internet Explorer">
    <option value=" Firefox">
    <option value="Chrome">
    <option value="ISafari">
  </datalist>
  <input type= " submit " />
</form >
```



Remarque : L'id du tag **<datalist>** doit être identique à la valeur de l'attribut **list** du champ **<input />** associé.


```
<form action = " traitKeygen.asp " method = " Get " >  
  Username : <input type= " text" name= " nomuser " />  
  Encryption : <keygen name= " clésecurité" />  
    <input type= " submit "/>  
</form >
```

A screenshot of a web form. On the left, there is a text input field labeled 'Username:' containing the text 'Michel Buffa'. To its right is a dropdown menu labeled 'Encryption:' with 'Haut niveau' selected. Below the dropdown, a list is visible showing 'Haut niveau' and 'Niveau moyen'. To the right of the dropdown is a button labeled 'Envoyer'.

- Le navigateur génère une clé, il affiche un dialogue « génération de la clé ».
- Cette opération peut prendre plusieurs minutes », puis il soumet le formulaire avec la clé (il s'agit du champ nom utilisateur crypté).

Nouveaux attributs de formulaire

- HTML 5 introduit de nouveaux attributs qui s'appliquent à tous les champs, et permet aussi de « sortir » des champs du formulaire lui-même tout en y étant rattachés
- Ce qui permet plus de souplesse pour la création de l'interface utilisateur.
- Les principaux attributs sont les suivants :

Placeholder	required	Formnovalidate
pattern	autofocus	multiple
formaction	formmethod	

Formaction

- Attribut qui remplace l'attribut « action » du formulaire.
- Il indique l'URL à laquelle envoyer les données du formulaire.

Formmethod

- Attribut qui remplace l'attribut « method » du formulaire.
- Il indique la méthode HTTP d'envoi des données à l'URL.

```
<form action = " controleForm.php " method = " POST " >  
<input type= " submit " formaction= " post.php"  
        formethod= " get" value= " visualiser " />  
<input type= " submit " value= " envoyer " />  
</form />
```

Placeholder

- Attribut qui permet de renseigner un texte indicatif par défaut dans un champ de formulaire.
- L'attribut **placeholder** peut être placé sur les éléments :
 - `<input />` : de type **text, search, password, url, tel, email**
 - `<textarea />`

Required

- Attribut qui permet de rendre obligatoire le remplissage d'un champ et bloquer la validation du formulaire si l'un des champs (concernés par cet attribut) n'a pas été renseigné
- L'attribut **required** peut être placé sur les éléments :
 - `<input />` : de type **text, search, password, url, tel, email, date, datetime, datetime-local, months, week, time, number, checkbox, radio, file**
 - `<textarea />`

Pattern

- Attribut qui permet de renseigner une certaine règle de syntaxe.
- La valeur est une expression régulière.

```
<input type="text" pattern="[A-F][0-9]{5}" />
```

- Certains patterns par défaut existant pour certains types de champ, comme les champs de type email ou url par exemple

- L'attribut pattern peut être placé sur les éléments :

<input /> : de type **text, search, password, url, tel, email**

Multiple

- Attribut qui permet à l'utilisateur d'entrer plusieurs valeurs.
- Cet attribut est utilisé avec :

```
<input type="email" />
```

```
<input type="file" />
```

```
<form action="controleForm.php" method="Get" >
```

Sélectionner un fichier :<input type="file"

```
name="image" multiple="multiple" />
```

```
<input type="submit" />
```

```
</form >
```

Autofocus

- Attribut qui définit le focus sur un champ particulier lors du chargement de la page.

```
<form action = " controleForm.php " method = " Get " >  
  Nom :<input type= " text "  
        name= " nom" autofocus= " autofocus " />  
  Prénom :<input type= " text " name= " nom" />  
  <input type= " submit " />  
</form >
```

Formnovalidate

- Attribut qui permet de soumettre un formulaire même s'il contient des champs qui doivent être validés et qui sont incorrects.

```
<form >  
  <input type= " submit " formnovalidate= "  
    formnovalidate" value= " envoyer sans valider " />  
</form >
```


La lecture audio et vidéo avec `<audio>` et `<video>`



Les nouvelles balises `<audio>` et `<video>`

- Une des **grandes nouveautés** de l'HTML 5 est la prise en charge **sans plugins**, de la lecture des flux **audio** et **vidéo**
- Fondamentalement, les nouvelles balises `<audio>` et `<video>` se comporte de la même manière et s'utilisent quasiment de la **même façon** que l'inclusion d'une simple image !

- syntaxe minimale pour la lecture d'un fichier **vidéo** :

```
<video src="video.webm"></video>
```

- syntaxe minimale pour la lecture d'un fichier **audio**:

```
<audio src="audio.mp3" controls></audio>
```

- syntaxe minimale pour l'inclusion d'une **image** :

```

```

La nouvelle balise `<video>`

- La simple syntaxe `<video src="video.webm"></video>`

nous offre donc la possibilité de lire une vidéo directement dans notre navigateur :



Aperçus dans Google Chrome

- Cependant, l'absence de l'attribut **controls** implique que la lecture, la pause et le stop doivent s'effectuer avec le clic droit ce qui n'est pas très pratique....

- Code source d'un élément <video> plus complet avec le contrôle, la taille, l'image d'intro, les différentes sources et le texte alternatif.

```
<video width="360" height="640" poster="image.jpg" controls>  
  <source src="video.mp4" type="video/mp4" />  
  <source src="video.webm" type="video/webm" />  
  <source src="video.ogv" type="video/ogg" />  
  l'alternative à la vidéo : un lien de téléchargement, un message, etc.  
</video>
```























Aperçus dans Mozilla Firefox

- Exactement la même vidéo que précédemment mais avec l'image d'intro et le contrôleur visible par le client.

La balise <video> possède également l'attribut **autoplay**, **preload** et **loop**

- La prise en charge par les navigateurs des différents formats vidéos est la suivante :

Navigateur	H.264/MP4/AAC	Ogg/Theora/Vorbis	WebM
3.5+ 		3.5+ 	4.0+ 
10.5 		10.5+ 	10.6+ 
9.0+ 	9.0+ 		9+ avec installation d'un plugin  
3.0+ 	4.1+ 	3.0+ 	6.0+ 
4.0+ 	3.1+ 		9+ avec installation d'un plugin

- Pour une prise en charge simplifiée par les navigateurs, il existe de nombreux scripts JavaScript comme : **Popcorn.js**, **videoJS**, **html5media**, **SublimeVideo** ect...
- Outil de conversion vidéo simple (Micro Video Converter)
<http://www.mirovideoconverter.com/>

La nouvelle balise <audio>

- La syntaxe `<audio src="audio.mp3" controls></audio>` nous offre la possibilité de lire un fichier audio directement dans notre navigateur tout comme pour la vidéo.
- Aperçus dans les principaux navigateurs :



La nouvelle balise `<audio>`

- De même que pour l'élément `<video>`, nous pouvons avec l'élément `<audio>` gérer plusieurs sources pour répondre à l'incompatibilité des navigateurs.

```
<audio controls>
```

```
  <source src="audio.oga" type="audio/ogg" />
```

```
  <source src="audio.mp3" type="audio/mpeg" />
```

```
  <source src="audio.acc" type="audio/mp4" />
```

l'alternative à l'audio: un lien de téléchargement, un message, etc.

```
</audio>
```

La balise `<audio>` possède également l'attribut `autoplay`, `preload` et `loop`

La nouvelle balise <audio>

- La prise en charge par les navigateurs des différents formats audio est la suivante :

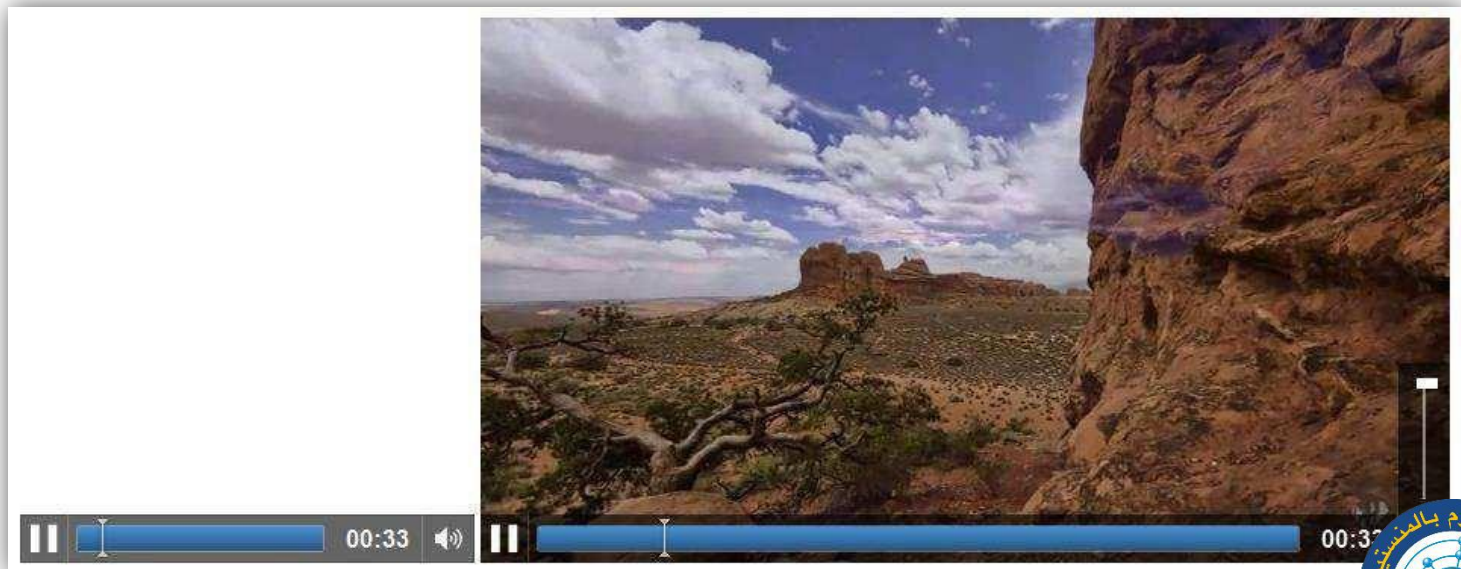
Navigateur	MP3	Ogg Vorbis	ACC	WAV
3.5+ 	✗	✓	✓	✓
10.5 	✗	✓		✓
9.0+ 	✓	✗	✗	✗
3.0+ 	✓	✓	✓	✓
4.0+ 	✓	✗	✓	✓

- De même que pour la balise vidéo, vous pouvez utiliser les scripts **JavaScript** vus précédemment pour une manipulation simplifiée de ces balises.

Activité sur les balises <audio> et <video>

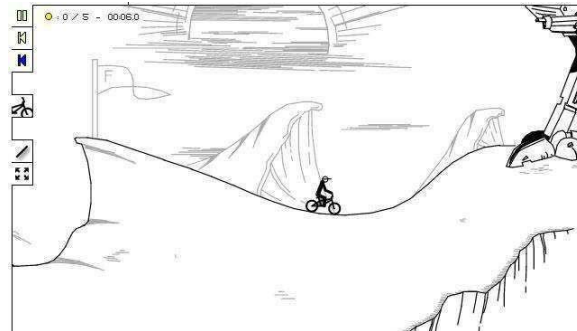
- En utilisant les codes sources de cette présentation, essayez de faire marcher la lecture d'un fichier audio et vidéo sur la même page dès son chargement avec l'intégration des contrôles.

On souhaite obtenir ce résultat :



Le dessin avec

<canvas>



Canvas rider

Le dessin avec Canvas

- HTML 5 offre aujourd'hui la possibilité de **dessiner**, de traiter des **images** en allant jusqu'à la possibilité de créer un **jeu complet** grâce au nouvel élément **<canvas>**
- Ainsi, cette fonctionnalité devient une alternative puissante face au **flash** ou encore au **Java** dans la réalisation des ces différentes tâches
- La syntaxe HTML pour la création d'un élément **canvas** est très simple :

```
<canvas id="dessin" width="640" height="480">
```

Votre navigateur ne support pas canvas ! Bouuuu ...

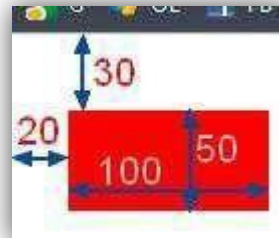
```
</canvas>
```

Dessiner un rectangle rouge

- Pour créer un dessin dans cet élément, on va recourir à l'API de dessin 2D avec le langage JavaScript
- Exemple de création d'un simple rectangle rouge :

```
<canvas id="dessin" width="640" height="480">
  Votre navigateur ne support pas canvas ! Bouuuu ...
</canvas>
<script>
  var dessin = document.getElementById('dessin');
  var forme = dessin.getContext('2d');
  forme.fillStyle = "red";
  forme.fillRect(20, 30, 100, 50);
</script>
```

- Aperçus du résultat :



- Les formes primitives en Canvas ne comprennent que des fonctions de la famille des rectangles :

Fonction	Rôle
<code>fillRect(x, y, w, h)</code>	Rectangle plein
<code>strokeRect(x, y, w, h)</code>	Rectangle surligné
<code>clearRect(x, y, w, h)</code>	Rectangle vide (efface)

- Code source pour un drapeau suisse :

```
var dessin = document.getElementById('dessin');
var forme = dessin.getContext('2d');
forme.lineWidth = 2; // Largeur du trait
forme.strokeStyle = "#ccc"; // Couleur du trait
forme.fillStyle = "red"; // Rectangle plein rouge
forme.fillRect(10, 10, 200, 100); // coord. + taille rectangle
forme.clearRect(100, 20, 20, 80); // rectangle qui efface
forme.clearRect(70, 50, 80, 20); // 2ème rectangle qui efface
forme.strokeRect(1, 1, 220, 120); // cadre gris
```

- Résultat :



Tracer un chemin

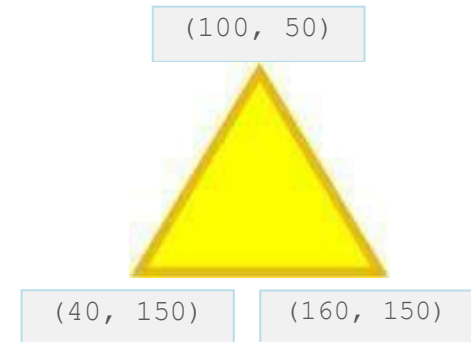
- A défaut d'avoir simplement le rectangle comme tracé de primitive, **Canvas** possède l'avantage d'une grande liberté avec le tracé de **chemins**
- Tracé d'un triangle avec les **chemins** :

```
<script>
  var dessin = document.getElementById('dessin');
  var forme = dessin.getContext('2d');
  forme.fillStyle = "#FFFF00"; // triangle jaune
  forme.strokeStyle = "#EABB00"; // contour jaune foncé
  forme.lineWidth = 5; // Largeur du contour

  forme.beginPath(); // Commence le chemin
  forme.moveTo(100, 50); // Se positionner
  forme.lineTo(160, 150); // Se rendre à ...
  forme.lineTo(40, 150); // Se rendre à ...
  forme.closePath(); // Termine le chemin

  forme.fill(); // color le fond
  forme.stroke(); // trace le contour
</script>
```

➤ Résultat :



Activité : tracer la triforce de Zelda



En utilisant le code source suivant comme base de l'exercice, essayez de tracer la triforce de Zelda.

```
<script>
    var dessin = document.getElementById('dessin');
    var forme = dessin.getContext('2d');
    forme.fillStyle = "#FFFF00";
    forme.strokeStyle = "#EABB00";
    forme.lineWidth = 5;
    var x = 200;
    var y = 200;
    forme.beginPath();
    forme.moveTo(x, y);
    forme.lineTo(x+60, y+100);
    forme.lineTo(x-60, y+100);
    forme.closePath();
    forme.fill();
    forme.stroke();
</script>
```



- Il existe de nombreuses possibilités à Canvas que vous pouvez retrouver notamment sur la documentation de Mozilla : <https://developer.mozilla.org/fr/HTML/Canvas>
- Ces possibilités nous permettent de gérer la **transparence**, les **dégradés**, de faire des **transformations** (échelle, rotation, translation ...), de **traiter des images**, de modifier des **pixels**, de gérer les **sprites**, d'écrire du **texte** ect...
- Applications marquantes utilisant les canvas :
 - Sketchpad <http://mugtug.com/sketchpad/>
 - The Wilderness Downtown : <http://www.thewildernessdowntown.com/>

- Le socle canvas a permis le saut vers **WebGL**, une API **JavaScript** spécialement conçus pour la **création 3D**.
- Cette technologie est en cours d'expérimentation et est principalement compatible avec **Google Chrome** et **Mozilla Firefox** pour le moment.








- Quelques expérimentations :
 - Google : <http://www.chromeexperiments.com/webgl>
 - Hakim El Hattab : <http://hakim.se/experiments/>
 - Mr doob : <http://mrdoob.com/>
 - Jeu de rally : <http://triggerrally.com/>



La balise canvas et les navigateurs

- La prise en charge par les navigateurs de la balise canvas est la suivante :

Navigateur	Version
	3+
	9+
	9+
	1+
	3.1+

- Il existe une solution alternative pour internet explorer < 9 avec une librairie **JavaScript** :
<http://code.google.com/p/explorercanvas>