

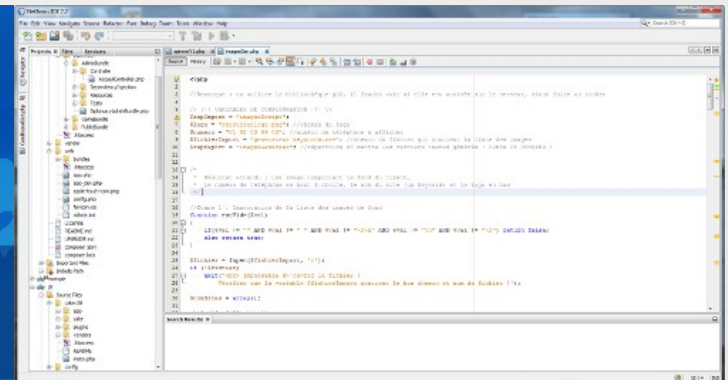
Formation web



Réseau Internet, IDE, W3C



HTML, CSS, PHP, MySQL





Utilité du cours

- Le web est :
- Omniprésent dans nos vies, ses facettes sont multiples et font appel à de la créativité.
- Un secteur qui recrute dans des domaines et supports variés (santé, finance, associations, applications mobiles...)
- Un secteur qui rapporte en indépendant (missions des junior entreprises majoritairement orientées web, possibilité d'être auto-entrepreneur, freelance...).





Le web, trop facile ?

Le web est parfois sous-estimé du fait que :

- les projets sont assez simples (peu de maths, pas de 2D ou 3D...)
- traumatisme de la bulle Internet ? Méfiance dans les années 2000
- Nouvelles technologies : web 3.0, metaverse...

Les projets web soulèvent en fait beaucoup de problématiques:

- Au moins 5 langages à maîtriser : HTML, CSS, Javascript, PHP, SQL (cf syllabus)
- Structuration optimale de la BDD
- Connaître certaines solutions (bibliothèques, framework, CMS comme Drupal, Joomla!...)
- Référencement
- Sécurité (ex: journaux en lignes, cours de bourse truqués), accessibilité, ergonomie...



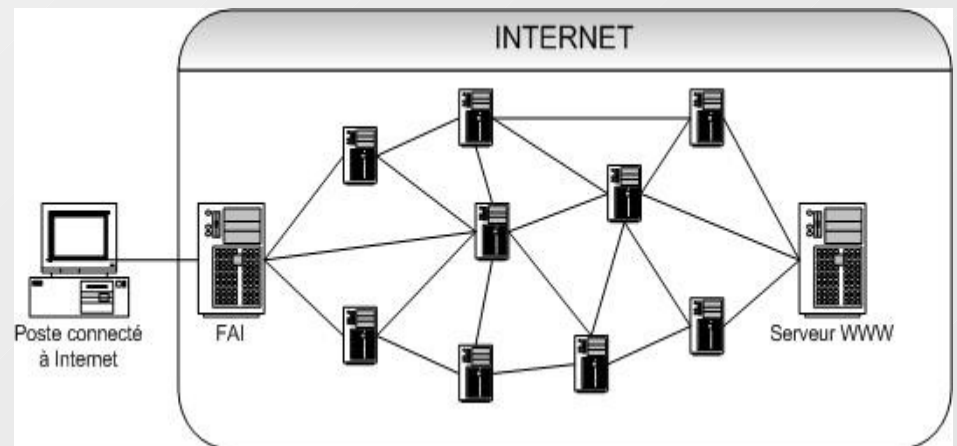
Internet : un réseau redondant

Le réseau internet



Parcours d'une requête :

Client => routeurs => serveur



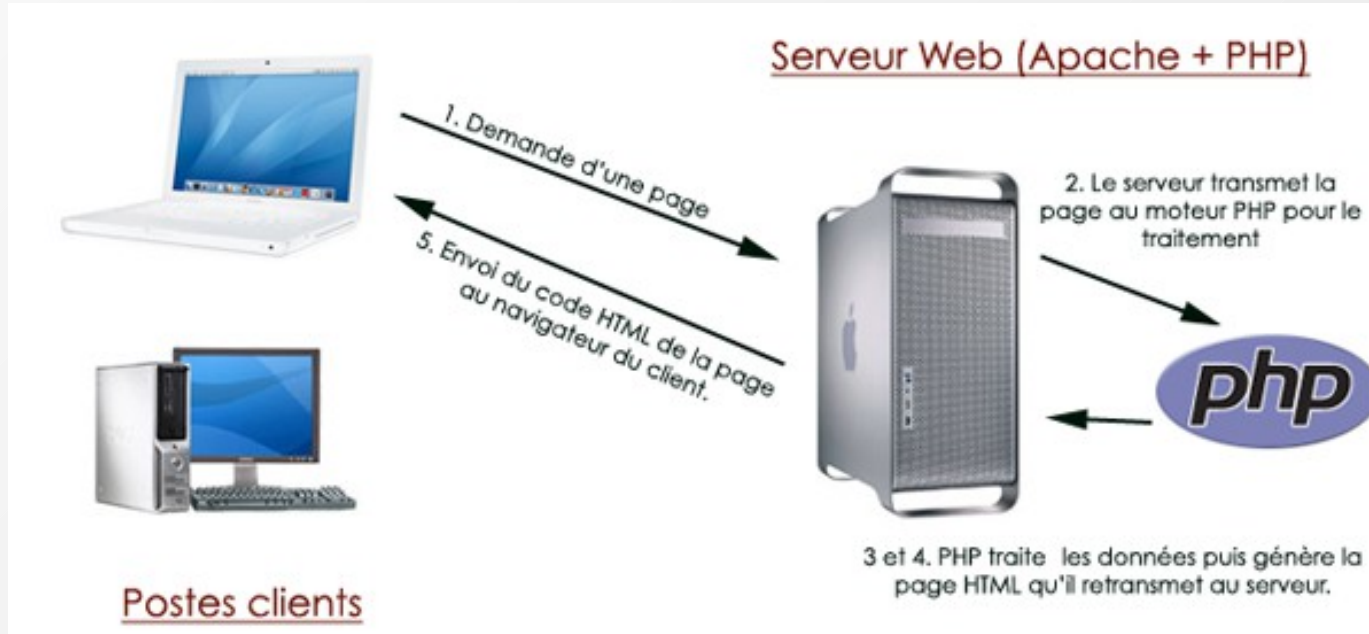


Serveur et client

- En pratique, on fait abstraction des routeurs (on suppose que l'information est correctement acheminée)
- On a donc un *client* (ordinateur qui demande à visualiser une page internet) et un *serveur* qui répond à cette *requête*.
- Client sous n'importe quel OS
- Serveur sous Linux (cas rares sous Windows)
- Nom de domaine transformé en l'IP du serveur (résolution DNS).



Serveur et client



2.5 Eventuellement, le PHP fait des requêtes dans la bases SQL

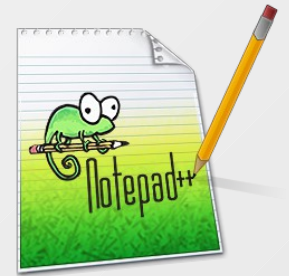
5.5 Eventuellement, le client exécute le code JS qui se trouve dans la page

Génération d'une page



Outils utiles

Editeurs de texte et IDE



- Editeurs de textes (suffisants pour HTML/CSS voire PHP) :
 - Windows : *notepad++* (plugin explorer pour avoir l'architecture des dossiers), *SublimeText* (plus puissant en restant léger)
 - Linux : *vim*, *emacs*
- IDE :
 - PHPStorm* (gratuit pour les étudiants)





HTML et les navigateurs

Problèmes d'interprétation des navigateurs web

- Le HTML et le CSS sont des langages qui viennent envelopper / enrichir le contenu d'une page web et qui *s'exécutent côté client*
- Un navigateur est un logiciel qui interprète ces langages pour fournir un rendu facilement compréhensible pour être humain
- Il existent *plusieurs navigateurs, qui interprètent différemment* la syntaxe HTML et CSS. Certains ont des outils de debuggage et de simulation de petit écran comme Chrome
- Le rôle de l'intégrateur web est de faire en sorte que pour tous les navigateurs récents, le site s'affiche correctement.



Le langage HTML

- Il appartient aux langages de *description* (définissent un contenu statique) qui diffèrent des langages de *programmation* (génèrent un contenu variable, dynamique).
- *Structure* l'information (aucune mise en forme), en suivant des conventions précises pour que les créateurs de browser n'aient pas à gérer différents nommages pour afficher la même chose (par exemple, il existe une balise `<p>` pour les paragraphes de texte, et `<paragraph>` n'existe pas).
- Inventé par Tim Berners-Lee en 1991, qui créa le Word Wide Web Consortium (W3C), organisme de normalisation du web
- Version actuelle : **HTML 5**



Première page HTML

Structure de base d'une page HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Formation web</title>
  </head>
  <body>
    <!-- Le contenu du site -->
  </body>
</html>
```

A recopier dans un fichier d'extension .html



Première page HTML



Ne pas oublier **l'indentation** : la lisibilité du code est utile pour :

- le débogage
- se remettre sur son code quelques semaines plus tard
- vital quand on travail en équipe.

Explications du code :

La ligne `<!DOCTYPE html>` indique au navigateur qu'on est en HTML5 et sous-tend la syntaxe qui va avec



Première page HTML

```
<head>  
  <meta charset="UTF-8">  
  <title>Formation web</title>  
</head>
```

Explications du code :

La balise <head> est l'en-tête de la page, elle contient le titre qui s'affiche dans l'onglet (le <title>) et des balises meta (ici, l'encodage). Ces données ne s'affichent pas dans la page elle-même.



Première page HTML

Syntaxe des commentaires : <!-- -->

```
<!-- Un commentaire sur une ligne -->  
<!-- Un commentaire sur  
plusieurs lignes -->
```

Non affichés dans la page web mais lisibles dans le code source par n'importe quel client.



Les balises HTML de base

Premières balises structurelles :

- Les paragraphes, qui permettent d'insérer du texte :

<p> texte </p>

- Rq : Les sauts de lignes présent dans l'éditeur de texte n'ont aucun effet lorsqu'on affiche la page, il faut utiliser la balise autofermante

**
** ou plutôt en HTML5 **
**

- Les titres que peut contenir la page, par ordre d'importance :

<h1>, <h2>, ... <h6>.

- Mettre en valeur quelques mots (dans un paragraphe) : **** ou plus puissant, ****. Le style généré par ces balises (italique et gras) pourra être changé lors de l'écriture du CSS.

- On peut également utiliser la balise **<mark>** pour faire ressortir une portion de texte pas très importante (les robots la prennent donc peu en compte pour le référencement).



Les balises HTML de base

Les listes à puces :

Code:

```
<!-- Liste non ordonnée -->  
<ul>  
  <li>Paris</li>  
  <li>Lyon</li>  
  <li>Marseille</li>  
</ul>
```

Rendu:

- Paris
- Lyon
- Marseille

Avec le CSS, on pourra modifier ou supprimer la puce. Les listes sont notamment utiliser pour les menus, qui sont des listes de liens. Pour faire une *liste ordonnée* (numérotée de 1 à X), il suffit de remplacer « ul » par « ol ».



Les balises HTML de base

Moins utilisées, les listes de définition :

Code:

```
<!-- Liste de définitions -->
<dl>
  <dt>HTML</dt><dd>Hypertext Markup Language</dd>
  <dt>CSS</dt><dd>Cascading Style Sheets</dd>
</dl>
```

Rendu:

HTML	
	Hypertext Markup Language
CSS	
	Cascading Style Sheets



Les balises HTML de base

Les liens

Il existe deux types de liens :

- **Externes** (ou absolus), ils mènent vers un autre site.

Exemple de chemin absolu : `http://monsite.fr/`

- **Internes** (ou relatifs), ils mènent vers une page du site actuel.

Exemple de chemin relatif : `dossier/autrepage.html`

Un lien s'écrit ainsi : `Texte du lien`

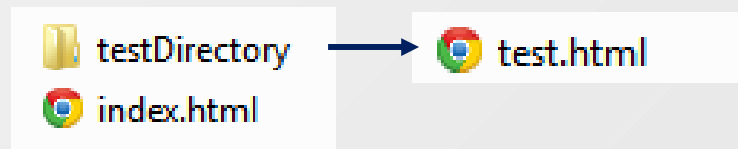
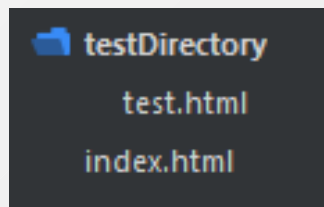


Les balises HTML de base

Les liens

Exemple : je suis sur la page index.html, je veux créer un lien vers la page test.html contenue dans le dossier testsDirectory (qui est au même niveau que index.html) :

```
<a href="testsDirectory/test.html">Cliquez ici pour tester !</a>
```



Remarque : si on veut remonter dans l'arborescence des dossier :

```
<a href="../test.html">Cliquez ici pour tester !</a>
```

(plutôt ../index.html)



Les balises HTML de base

Les ancres ou « id »

Elles permettent de créer des points de repères dans la page. Elles sont très utiles pour le CSS et le javascript (elles permettent d'identifier et donc de cibler des éléments), mais également pour faire des liens dans une sous-partie de page particulièrement longue. Pour créer une ancre, on utilise l'attribut **id** :

```
<h2 id="une_ancre">Titre quelconque</h2>
```



Les balises HTML de base

Les ancres

Pour créer un lien vers l'ancre (ici on suppose qu'on est sur la même page que l'ancre), il faut ajouter un # devant le nom de l'ancre :

```
<a href="#une_ancre">Accès à l'ancre</a>
```




Les balises HTML de base

Les ancres

Exemple d'utilisation des ancres : le sommaire d'une page wikipedia.

Cliquer sur « 3. Description de HTML » enverra directement à la partie de la page associée.

Sommaire [\[masquer\]](#)

- 1 Dénomination
- 2 Évolution du langage
 - 2.1 1989-1992 : Origine
 - 2.2 1993 : Apports de NCSA Mosaic
 - 2.3 1994 : Apports de Netscape Navigator
 - 2.4 1995-1996 : HTML 2.0
 - 2.5 1997 : HTML 3.2. et 4.0
 - 2.6 2000-2006 : XHTML
 - 2.7 2007 à nos jours : HTML5 et abandon du XHTML 2
 - 2.8 Le futur du HTML : sans numéro de version ?
- 3 Description de HTML
 - 3.1 Syntaxe de HTML
 - 3.2 Structure des documents HTML
 - 3.3 Éléments de HTML
 - 3.4 Attributs de HTML
 - 3.5 Jeu de caractères
- 4 Interopérabilité de HTML
- 5 Notes et références
- 6 Voir aussi
 - 6.1 Articles connexes
 - 6.2 Liens externes

Dénomination [\[modifier\]](#)

L'[anglais](#) *Hypertext Markup Language* est rarement traduit littéralement



Les balises HTML de base

Ajouts sur les liens

Ajouter une infobulle au survol d'un élément (lien ou autre type de balise) : l'attribut title.

```
<a href="page1.html" title="Voici une infobulle">En plein survol des Alpes</a>
```

Forcer l'ouverture d'un lien dans un nouvel onglet :

```
target="_blank"
```

Lien pour envoyer un e-mail : faire précéder l'adresse de mailto :

```
<a href="mailto:monadresse@gmail.com">Me contacter</a>
```

Un lien de téléchargement est un simple lien vers un fichier :

```
<a href="fichier.zip">Télécharger</a>
```



Les balises HTML de base

Les images

Problématique de la vitesse de téléchargement des éléments d'une page : éviter les BITMAP, qui pèsent plusieurs MO pour une seule image.

Les principaux types d'images :

- JPEG : 16 millions de couleurs, compressent très bien les photos
- PNG : 16 millions de couleurs également pour sa version 24bits, et gère la *transparence*.
- GIF : limité à 256 couleurs, utilisé pour les animations (un peu Old school aujourd'hui).



Les balises HTML de base

Les images

Une image s'insère via une balise autofermante **** qui a deux paramètres :

- src**, qui permet d'indiquer le chemin de l'image
- alt**, qui permet d'associer un texte alternatif à l'image, au cas où celle-ci ne serait ne pourrait être chargée (à cause d'une surcharge de serveur ou d'un problème de connexion...), mais également pour les robots (qui prennent aussi en compte le nom de l'image pour le référencement) et pour les malvoyants !

```

```



Les balises HTML de base

Quelques balises secondaires

Balise *abréviation* **<abbr>** :

```
The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.
```

Au passage de la souris :

The WHO was founded in 1948.

World Health Organization

Balises de *citations* :

- **<blockquote>** (citation longue)
- **<q>** citation courte



Les balises HTML de base

Quelques balises secondaires

Mise en exposant :

`3² = 9`

donne

$3^2 = 9$

La mise en indice se fait avec `<sub>`

Adresse de contact :

`<address>`

Texte supprimé : ``

Texte inséré : `<ins>`



Les balises HTML de base

TP 1 : Une première page web (à réaliser en local : aucun besoin d'envoyer les fichiers sur un serveur)

Créer une page avec un maximum des éléments précédents.

Elle comportera notamment :

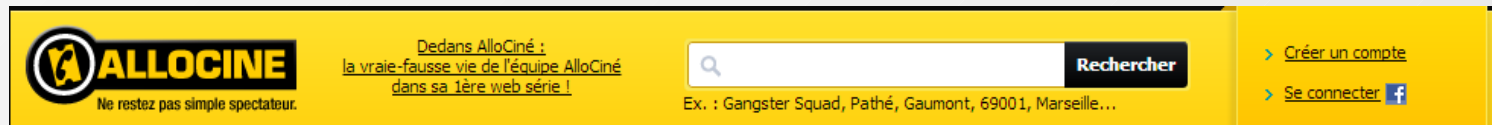
- un menu sous forme de liste de liens
- plusieurs paragraphes contenant une abréviation quelconque, la formule mathématique $E=mc^2$, la mise en valeur de certains mots. Chaque paragraphe doit être précédé d'un titre.
- une image quelconque
- un lien de téléchargement vers un fichier quelconque de votre disque dur (de petite taille)
- mettre des commentaires si nécessaire



Les nouveautés de HTML5

Les balises sémantiques et structurantes (mais invisibles)

<header> : Contiendra la bannière, le logo, le slogan du site



<footer> : Le pied de page (liens de contact, mentions légales...)





Les nouveautés de HTML5

Les nouvelles balises sémantiques et structurantes

<nav> : Les principaux liens de navigation du site => menu principal

<section> : Section de page servant à regrouper les contenus en fonction de leur thématique.

<article> : portion indépendante de la page, partie de page pouvant être reprise par un autre site. Exemple : article de blog.



Les nouveautés de HTML5

Les nouvelles balises sémantiques et structurantes

<aside> : informations complémentaires, souvent placée sur le côté

PHP

« PHP » redirige ici. Pour les autres significations, voir [PHP \(homonymie\)](#).

PHP: **H**ypertext **P**reprocessor³, plus connu sous son sigle **PHP** (**A**cronyme **r**écursif), est un **langage de scripts libre**⁴ principalement utilisé pour produire des **pages Web** dynamiques via un **serveur HTTP**³, mais pouvant également fonctionner comme n'importe quel **langage interprété** de façon locale, en exécutant les programmes en ligne de commande. PHP est un **langage impératif** disposant depuis la version 5 de fonctionnalités de **modèle objet** complètes⁵. En raison de la richesse de sa **bibliothèque**, on désigne parfois PHP comme une **plate-forme** plus qu'un simple langage.

Sommaire [masquer]

- 1 Historique
 - 1.1 Historique des versions
 - 1.1.1 PHP 6
- 2 Présentation
- 3 Fonctionnement
- 4 Syntaxe
 - 4.1 Hello world
 - 4.2 Balises
 - 4.3 Instructions
 - 4.4 Structures de contrôle
 - 4.5 Génération de code HTML
- 5 Programmation orientée objet (POO)
- 6 Mascotte
- 7 Quelques exemples d'application
- 8 Différentes plates-formes
- 9 Accélération
- 10 Notes et références
- 11 Voir aussi
 - 11.1 Bibliographie
 - 11.2 Articles connexes
 - 11.3 Liens externes

PHP



Apparu en	1994
Auteur	Rasmus Lerdorf
Développeur	The PHP Group
Dernière version	5.4.11 [d] (17 janvier 2013) [+] -]
Paradigmes	Programmation impérative orientée objet fonctionnelle
Typage	Typage dynamique faible
Influencé par	Perl C/C++ Java
Système d'exploitation	Multiplate-forme
Licence	Licence PHP ²
Site web	www.php.net [d]
modifier	[i]



Les nouveautés de HTML5

Les nouvelles balises sémantiques et structurantes

Les **figures** : elles viennent enrichir le texte, complètent, illustrent les informations de la page. Il peut s'agir de graphiques, de code source, de citations...

On peut adjoindre à **<figure>** la balise **<figcaption>** pour donner un titre à la figure.

```
<figure>
  
  <figcaption>La preuve, enfin !</figcaption>
</figure>
```



Les nouveautés de HTML5

Les nouvelles balises sémantiques et structurantes

Liste des autres balises et aide à leur utilisation :

<http://www.w3schools.com/tags/default.asp> (doc HTML du W3C)



HTML avancé

Les balises universelles

Les *balises universelles* permettent d'englober des portions de la page ne répondant à aucune sémantique particulière. Elles sont très utiles pour le design. Avant l'HTML5, on les utilisait aussi à la place de `<nav>`, `<header>`...

Il en existe 2 :

- **`<div>`** : englobe son contenu autour d'un « bloc »
- **``** : englobe un petit contenu textuel (quelques mots, une phrase)

On pourra identifier ces balises (et les autres) de manière spécifiques avec un attribut « id » ou l'attribut « class » qu'on verra plus tard (une classe est un id qu'on peut utiliser sur plusieurs éléments).



HTML avancé

Les tableaux

- On insère le contenu d'un tableau via les balises **<table>**
- On définit son contenu ligne par ligne via les balises **<tr>** qui indiquent le début et la fin d'une ligne.
- Les balises **<td>** délimitent les cellules.

```
<table>
  <tr>
    <td>1ère cellule 1ère ligne</td>
    <td>2ème cellule 1ère ligne</td>
  </tr>
  <tr>
    <td>1ère cellule 2ème ligne</td>
    <td>2ème cellule 2ème ligne</td>
  </tr>
</table>
```



1ère cellule 1ère ligne	2ème cellule 1ère ligne
1ère cellule 2ème ligne	2ème cellule 2ème ligne



HTML avancé

Les tableaux

On verra comment modifier le design des tableaux (notamment ajouter des bordures aux cellules) via le CSS.

On peut ajouter une ligne d'en-tête (pour les noms des colonnes) en remplaçant les **<td>** de la première ligne par des balises **<th>**.

On peut rajouter un titre au tableau avec **<caption>**

On peut fusionner les cellules avec les propriétés *colspan* et *rowspan*.



HTML avancé

Les formulaires

Code de base (formulaire vide) :

```
<form method="post" action="traitement.php">
  <p>
    <!-- Emplacement des futurs champs -->
  </p>
</form>
```

L'attribut **action** prend comme valeur le chemin du script PHP qui va traiter les données du formulaire (vous pouvez le laisser vide ou mettre un dièse # avec d'éviter d'être redirigé vers une 404 à l'envoi du formulaire).

L'attribut **method** peut prendre 2 valeurs : get (les données sont envoyées via l'URL, cette dernière étant limitée à 255 caractères) et post (aucune limite, les données saisies dans le formulaire transitent pas par la barre d'adresse – mais ce n'est pas plus sécurisé que get).

A présent, créons des champs dans le formulaire.



HTML avancé

Formulaire – Les zones de texte

Monoligne :

```
<input type="text" name="pseudo" />
```

On spécifie le type de champ avec l'attribut **type**.



L'attribut **name** permettra plus tard, lors du traitement PHP, de savoir à quel champ correspond chaque valeur.



HTML avancé

Formulaire – Les zones de texte

On peut rajouter un libellé au champ via la balise `<label>` dont l'attribut `for` doit valoir la même chose que l'`id` du champ (qui peut avoir la même valeur que le **name**, par soucis d'homogénéité) :

```
<form method="post" action="traitement.php">
  <p>
    <label for="pseudo">Entrez votre pseudo : </label>
    <input type="text" name="pseudo" id="pseudo" />
  </p>
</form>
```



Entrez votre pseudo :

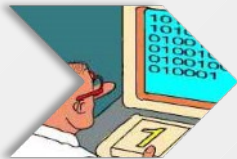


HTML avancé

Formulaire – Les zones de texte

Attributs supplémentaires :

- value** permet de pré-remplir le champ par une valeur par défaut
- placeholder** permet de donner une indication sur le contenu du champ. Cette indication disparaît au clic.
- size** permet d'agrandir le champ (mais on pourra aussi modifier sa taille en CSS). Ex : `size="40"`
- maxlength** permet de définir un nombre maximum de caractères



Application : créer un champ Prénom associé à un libellé, de taille 20, de taille maximale 30, et qui indique à l'intérieur du champ :
" écrivez ici votre prénom "



HTML avancé

Formulaire – Les zones de texte

Pour créer une zone de mot de passe, on change le type de l'input, qui passe de text à « password ».

Zone de texte multiligne : balise **<textarea>** (les attributs rows et cols permettent de l'agrandir, on la pré-remplie en mettant du texte entre les deux balises).

Les zones de saisies HTML5 sont encore peu gérées par les navigateurs.
Exemple : `<input type="email" />` : le navigateur est censé vérifier que le texte rentré est bien un e-mail.

Existe aussi pour les URL, les numéros de téléphone...

Certains comme `<input type="number" />` marchent bien.



HTML avancé

Formulaire – Les éléments d'option

Les cases à cocher :

```
Indiquez votre aliments préférés : <br>
<input type="checkbox" name="viande" id="viande"/>
<label for="viande">Viande</label><br>
<input type="checkbox" name="chocolat" id="chocolat"/>
<label for="chocolat">Chocolat</label><br>
<input type="checkbox" name="legumes" id="legumes"/>
<label for="legumes" checked ou checked="checked">Légumes</label><br>
```



Cochez si vous aimez les aliments suivants :

- ☐ Viande
- ☐ Chocolat
- ☐ Légumes



Pour cocher une case par défaut, on peut ajouter l'attribut `checked` ou `checked="checked"`.



HTML avancé

Formulaire – Les éléments d’option

Les zones d’options : l’utilisateur ne peut choisir qu’une option parmi un certain nombre (liées entre elles par le name) :

```
Indiquez votre régime alimentaire : <br>
<input type="radio" name="regime" id="vegetarien"/>
<label for="vegetarien">Végétarien</label><br>
<input type="radio" name="regime" id="carnivore"/>
<label for="carnivore">Carnivore</label><br>
<input type="radio" name="regime" id="omnivore"/>
<label for="omnivore">Omnivore</label><br>
```

Indiquez votre régime alimentaire :

- ☐ Végétarien
- ☒ Carnivore
- ☐ Omnivore

Pour savoir quelle bouton a été cliqué, il faudra ajouter un attribut **value** contenant une valeur spécifique au choix



HTML avancé

Formulaire – Les éléments d'option

Les listes déroulantes :

```
<label for="pays">Dans quel pays habitez-vous ?</label><br />
<select name="pays" id="pays">
  <option value="france">France</option>
  <option value="espagne">Espagne</option>
  <option value="italie">Italie</option>
  <option value="royaume-uni">Royaume-Uni</option>
  <option value="canada">Canada</option>
  <option value="etats-unis">États-Unis</option>
  <option value="chine">Chine</option>
  <option value="japon">Japon</option>
</select>
```

Sélection par défaut : attribut *selected*.

Dans quel pays habitez-vous ?

France	▼
France	
Espagne	
Italie	
Royaume-Uni	
Canada	
États-Unis	
Chine	
Japon	



HTML avancé

Formulaire – Les éléments d'option

Pour les longues listes, on peut regrouper les options en plusieurs catégories :

```
<optgroup label="Amérique">
  <option value="canada">Canada</option>
  <option value="etats-unis">Etats-Unis</option>
</optgroup>
```



HTML avancé

Formulaire – Structuration globale

On utilise les balises **<fieldset>** et **<legend>** pour regrouper les champs :

```
<form method="post" action="traitement.php">
  <p>
    <fieldset>
      <legend>Vos coordonnées</legend> <!-- Titre du fieldset -->

      <label for="nom">Quel est votre nom ?</label>
      <input type="text" name="nom" id="nom" />

      <label for="prenom">Quel est votre prénom ?</label>
      <input type="text" name="prenom" id="prenom" />

      <label for="email">Quel est votre e-mail ?</label>
      <input type="email" name="email" id="email" />

    </fieldset>
  </p>
</form>
```



HTML avancé

Formulaire – Structuration globale

Rendu :

Vos coordonnées

Quel est votre nom ?

Quel est votre prénom ?

Quel est votre e-mail ?



Astuce : pour sélectionner un champ par défaut (souvent le premier du formulaire), on lui rajoute l'attribut *autofocus*



HTML avancé

Formulaire – Bouton d'envoi

Le classique :

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

(le texte de value est affiché)

Le personnalisé :

```
<input type="image" src="monbouton.gif" name="envoi">
```

(le *name* permet de savoir sur quel bouton on a cliqué)



Les balises HTML de base

TP 2 : Affichage d'un formulaire

Affichez un formulaire permettant d'entrer le nom du futur président américain et des cases à cocher listant 5 états américains.



HTML bonus

Bonus – Les iframes

Les iframes servent à inclure une page (externe ou interne) dans une autre.

Exemple de code :

```
<iframe src="demo_iframe.htm" width="200" height="200"> </iframe>
```



HTML bonus

C'est le système qu'on peut utiliser pour inclure une vidéo youtube dans une page web :

Maria Maria
SantanaVEVO · 15 vidéos

15 451 133
55 840 803

J'aime À propos de **Partager** Ajouter à

Partager cette vidéo **Intégrer** E-mail Hangout

☒ Partager dans la playlist à partir de la vidéo en cours

```
<iframe width="420" height="315"
src="http://www.youtube.com/embed/nPLV7lGbmT4"
frameborder="0" allowfullscreen></iframe>
```

Taille de la vidéo : 420 × 315

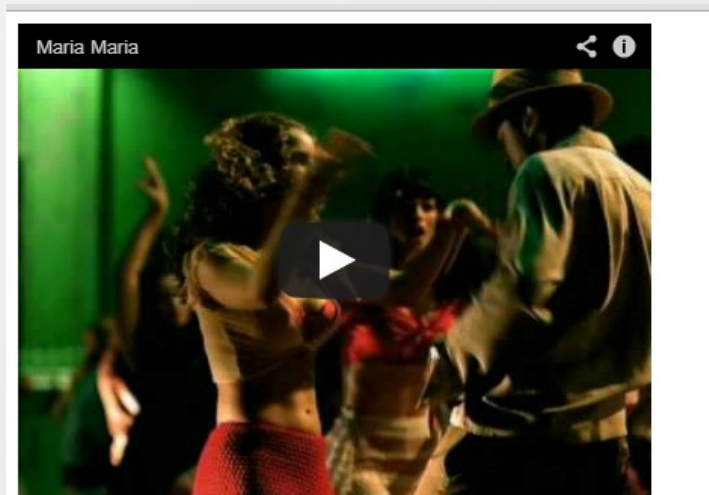


HTML bonus

Exemple :

```
<!DOCTYPE html>
<html status="Finished_Initial" POS="V" name="aim.v" ID="14111" frame="Aiming" frameID="1901" >
  <head>
    <meta charset="utf-8" />
    <title>Formation web EPSI</title>
  </head>

  <body>
    <iframe width="420" height="315" src="http://www.youtube.com/embed/nPLV7lGbmT4" frameborder="0" allowfullscreen></iframe>
  </body>
</html>
```



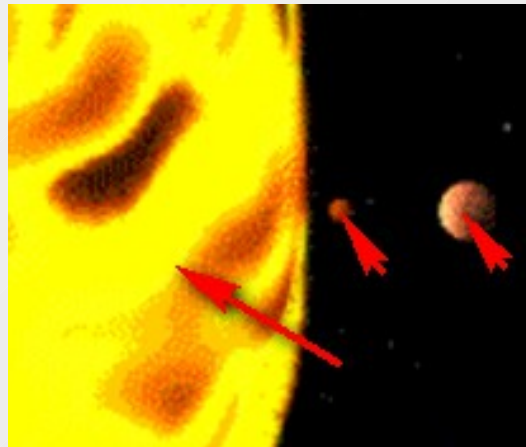


HTML bonus

Bonus – Les images multi-clicables

Il s'agit d'images clicables sur certaines zones (une image même contient plusieurs liens) :

http://www.w3schools.com/tags/tryit.asp?filename=tryhtml_areamap





HTML bonus

Culture – L'élément canvas

Il permet de faire des dessins par l'intermédiaire du langage Javascript. On peut y écrire du texte, y incorporer des formes simples (lignes, cercles...), y placer des images. Nécessite la connaissance de Javascript.

http://www.w3schools.com/tags/ref_canvas.asp



HTML bonus

Supplément : L'audio et la vidéo (HTML5)

Nous allons apprendre à nous servir de la documentation W3C, qui donne les formats compatibles avec les navigateurs et la syntaxe permettant d'incorporer sons et vidéos aux pages (seulement pour les navigateurs récents comme IE9 ! Il faut donc proposer une alternative en flash pour les autres...) :

Balise <audio> :

http://www.w3schools.com/tags/tag_audio.asp

Balise <video> :

http://www.w3schools.com/tags/tag_video.asp



HTML bonus

Balise <audio> : les formats

Dans « Definition and Usage », on voit que le format *mp3* est lisible pour Chrome, IE et Safari (et Firefox dans ses dernières versions). Il faut utiliser le format *ogg* pour être compatible avec Opera et toutes les versions de Firefox.

Exercice :

-télécharger un son instrumental en mp3 :

<http://s1download-universal-soundbank.com/mp3/sounds/22781.mp3>

(faire ctrl+s pour le sauvegarder)

-optionnel : avec un logiciel quelconque, convertir le fichier mp3 en ogg pour pouvoir proposer les deux formats :

<http://audio.online-convert.com/fr/convertir-en-ogg>



HTML bonus

Balise <audio> : le code de base

Exercice :

Adapter le code de la W3schools :

```
<audio controls autoplay>
  <source src="bansouri.mp3" type="audio/mpeg">
  <source src="bansouri.ogg" type="audio/ogg">
  La balise audio n'est pas supportée par le navigateur...
</audio>
```

Dans cet exemple, on a choisi d'afficher des éléments de contrôles (bouton Play, changement de volume...) et de lancer automatiquement le son (via l'attribut autoplay).

Un texte est affiché si le navigateur ne reconnaît pas <audio>.



HTML bonus

Balise <audio> : quelques attributs

Exercice :

On a d'autres attributs intéressants (cf la section Attributes) comme loop (faire tourner le son en boucle), muted (son désactivé par défaut), preload (si on ne veut pas pré-charger la musique afin d'économiser de la bande passante).

Testez par exemple l'attribut loop.



HTML bonus

Balise <audio> : un peu de JavaScript

Le Javascript (JS) est un langage de programmation (donc plus évolué que le HTML ou le CSS, qui sont des langages de description). Contrairement au PHP, qui s'exécute côté serveur pour générer le code des pages HTML, le JS est téléchargé par le client (l'ordinateur du visiteur) et s'exécute dans le navigateur.

Il est utilisé pour effectuer différents effets dynamiques (par exemple agrandir tout le texte de la page en cliquant sur une icône, ou faire défiler des images).



HTML bonus

Balise <audio> : un peu de JavaScript

Le but ici va être de ne pas afficher le lecteur audio et de lancer la lecture par simple clic sur la page. On va donc enlever les attributs de <audio>, et particulièrement controls :

```
<audio>
  <source src="bansouri.mp3" type="audio/mpeg">
  <source src="bansouri.ogg" type="audio/ogg">
  La balise audio n'est pas supportée par le navigateur...
</audio>
```

Ensuite, on va créer des balises <script> qui contiendront le code JS :

```
<script>
  //Le code javascript sera écrit ici
</script>
```



HTML bonus

Balise <audio> : un peu de JavaScript

Voici l'idée (l'algorithme) du script que l'on va créer :

- On sélectionne la balise <audio>
- Si on clique sur la page, on fait jouer le son.

Voilà le code correspondant :

```
<script>
    //On sélectionne la balise <audio>
    var player = document.querySelector('audio');
    //Si on clique sur la page
    document.onclick = function() {
        //on fait jouer le son
        player.play();
    }
</script>
```

La méthode
play() provient
d'une API JS

