BDW - Programmation web - HTML

Fabien Duchateau

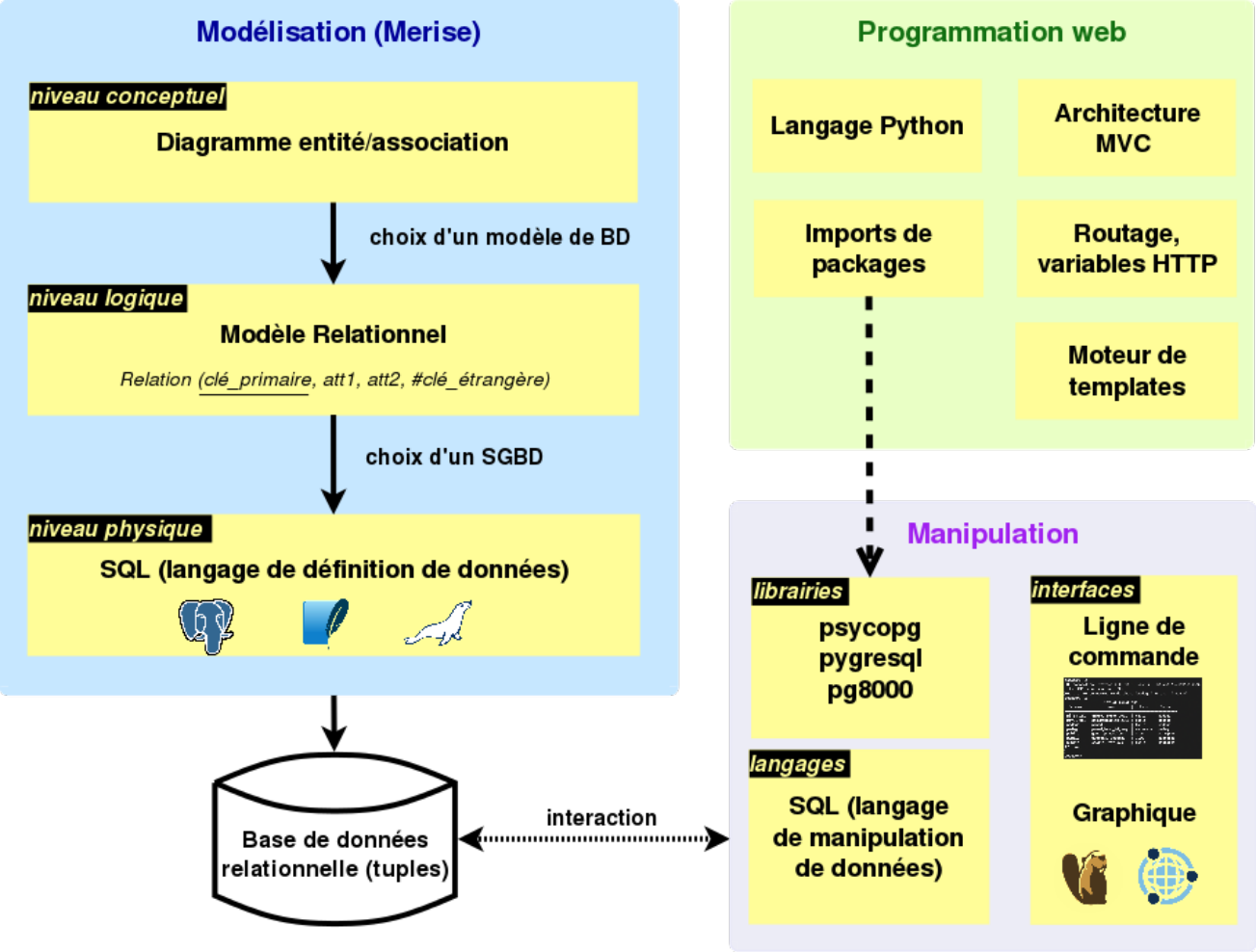
fabien.duchateau [at] univ-lyon1.fr

Université Claude Bernard Lyon 1

2024 - 2025

https://perso.liris.cnrs.fr/fabien.duchateau/BDW/

Positionnement dans BDW



Ces diapositives utilisent le genre féminin (e.g., chercheuse, développeuses) plutôt que l'écriture inclusive (moins accessible, moins concise, et pas totalement inclusive)

Introduction

Les BD sont rarement manipulées directement, mais plutôt :

- Via une application bureau (e.g., Firefox avec SQLite)
- Via un site / application web (e.g., un blog)
- Intégrées à un ERP (e.g., SAP)
- ...

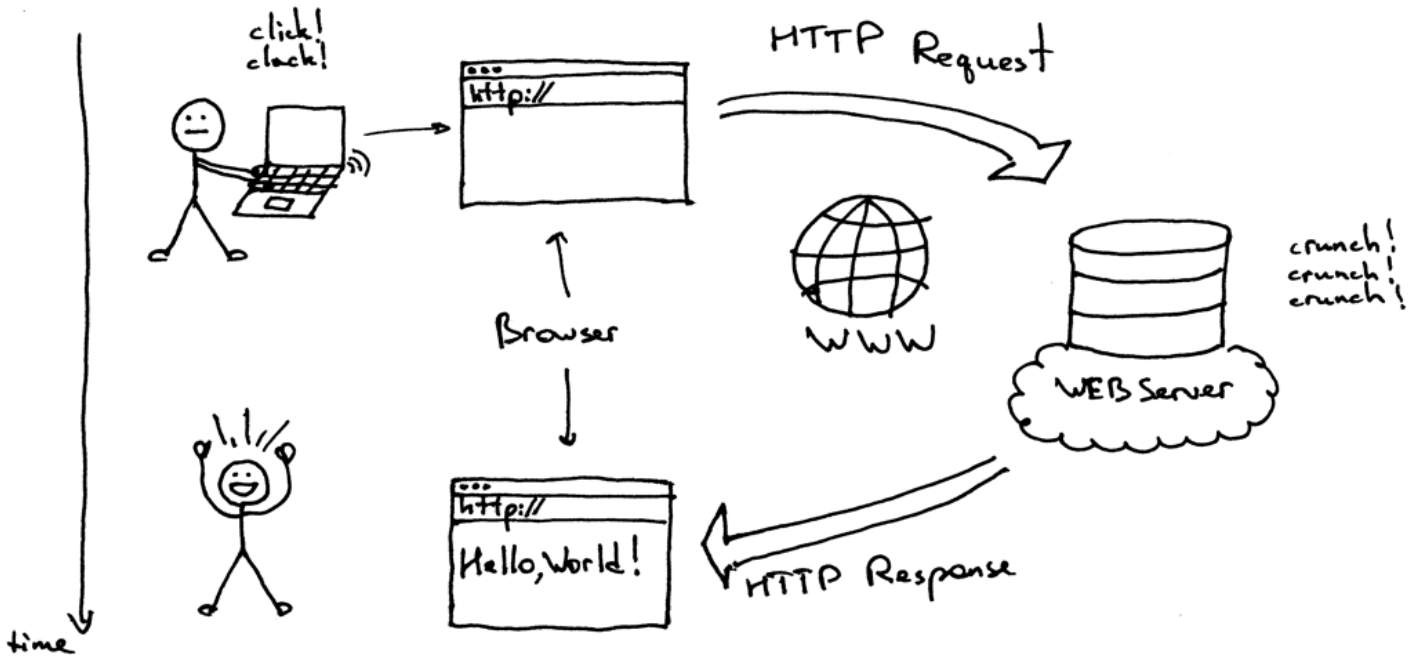
Dans ce cours, utilisation d'une BD à travers un site web :

- Popularité croissante
- Facilité de développement
- En prévision du stage de L3

---

http://sqlite.org/
http://fr.wikipedia.org/wiki/Progiciel_de_gestion_int%C3%A9gr%C3%A9

Fonctionnement d'un serveur web



Un serveur web travaille :

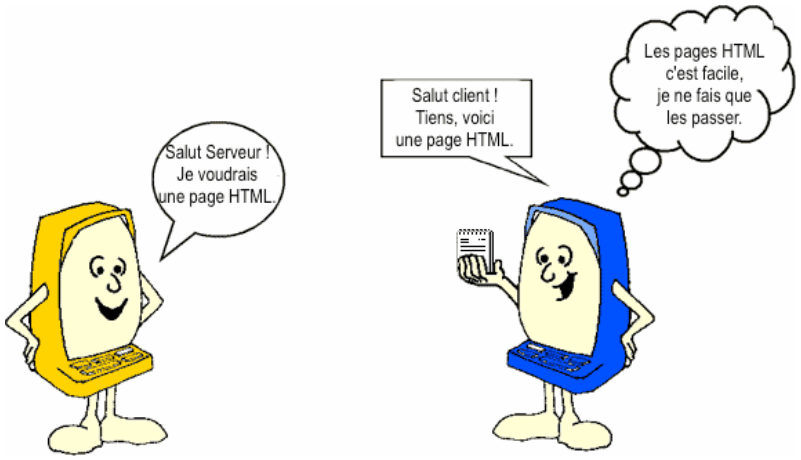
- Soit en relayant l'information (page statique)
- Soit en générant l'information (page dynamique)

---

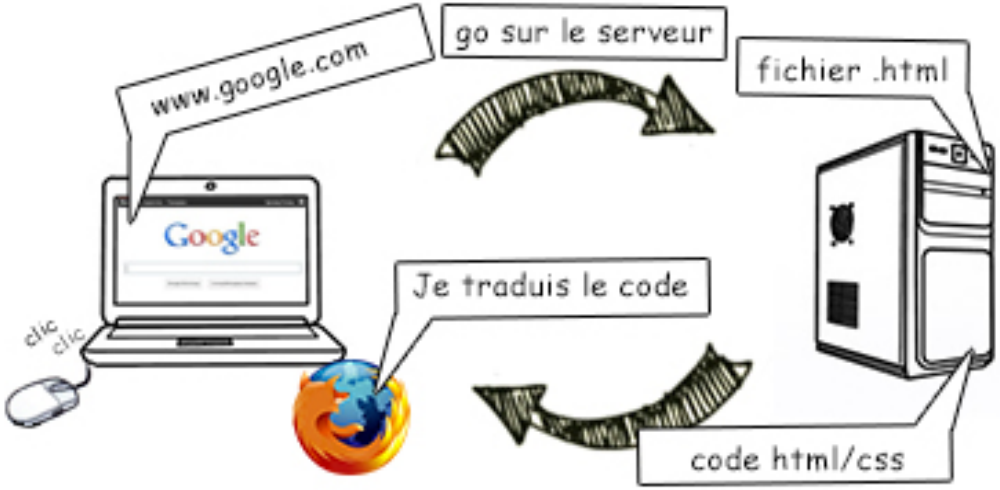
http://fr.wikipedia.org/wiki/Programmation_web
http://www.onelib.org/article/lets-build-a-web-server-part-1

Pages web statiques

- Demande d'accès à une page web depuis un navigateur (URL saisie, clic sur un lien, utilisation d'un signet, etc.)
- Le navigateur contacte le serveur web
- Le serveur web lit le fichier demandé sur le disque dur
- Le serveur web envoie le contenu HTML du fichier au navigateur
- Le navigateur affiche le contenu de la page (et demande éventuellement d'autres fichiers au serveur comme des images)

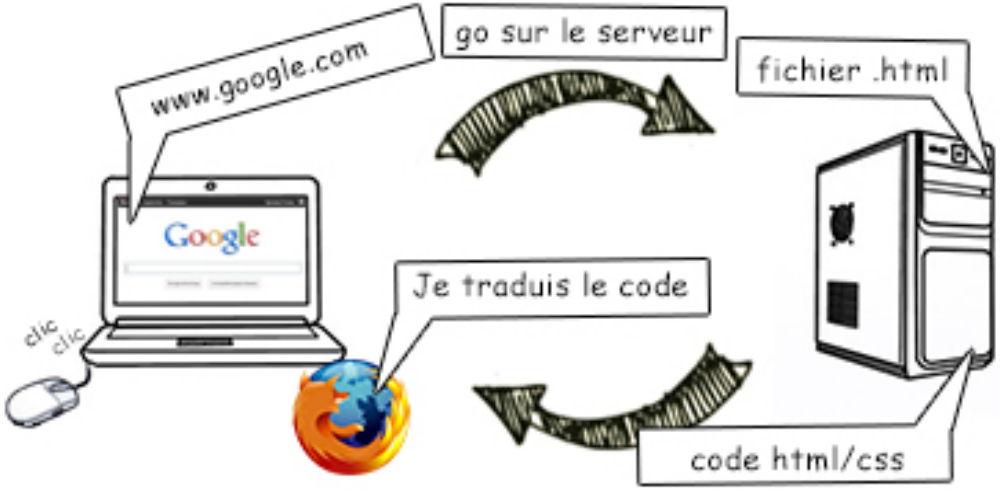


Pages web statiques (2)



http://wordpress.laurentdumoulin.com/
http://en.wikipedia.org/wiki/Static_web_page

Pages web statiques (2)



- Pas besoin de programmation côté serveur
- Pour changer le contenu, il faut éditer des fichiers
- Pas de participation des internautes (e.g., commentaires)

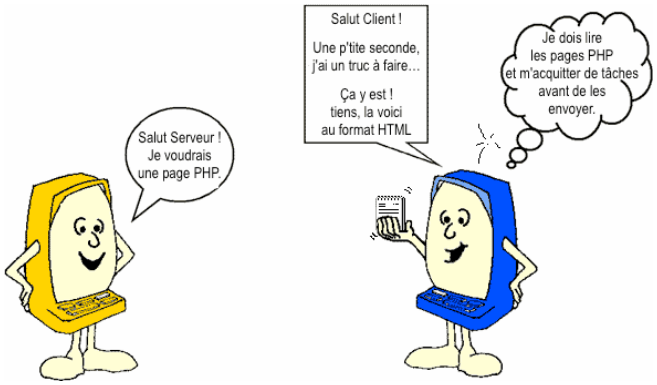
---

http://wordpress.laurentdumoulin.com/
http://en.wikipedia.org/wiki/Static_web_page

Pages web dynamiques

Une page dynamique génère son contenu au moment de la demande, en fonction :

- De paramètres fournis avec la demande (e.g., formulaires)
- Du contenu d'une base de données



Par dynamique, on parle ici de contenu (variable dans le temps). Ne pas confondre avec dynamique au sens interactif (e.g., avec du JavaScript)

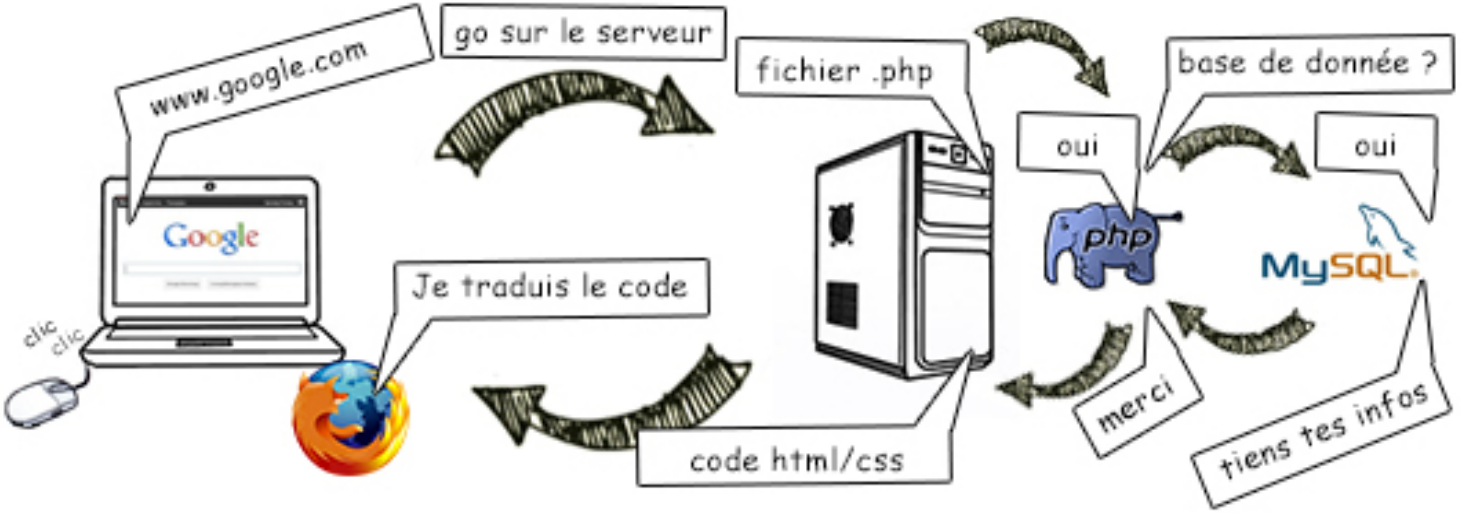
---

http://en.wikipedia.org/wiki/Dynamic_web_page

Pages web dynamiques (2)

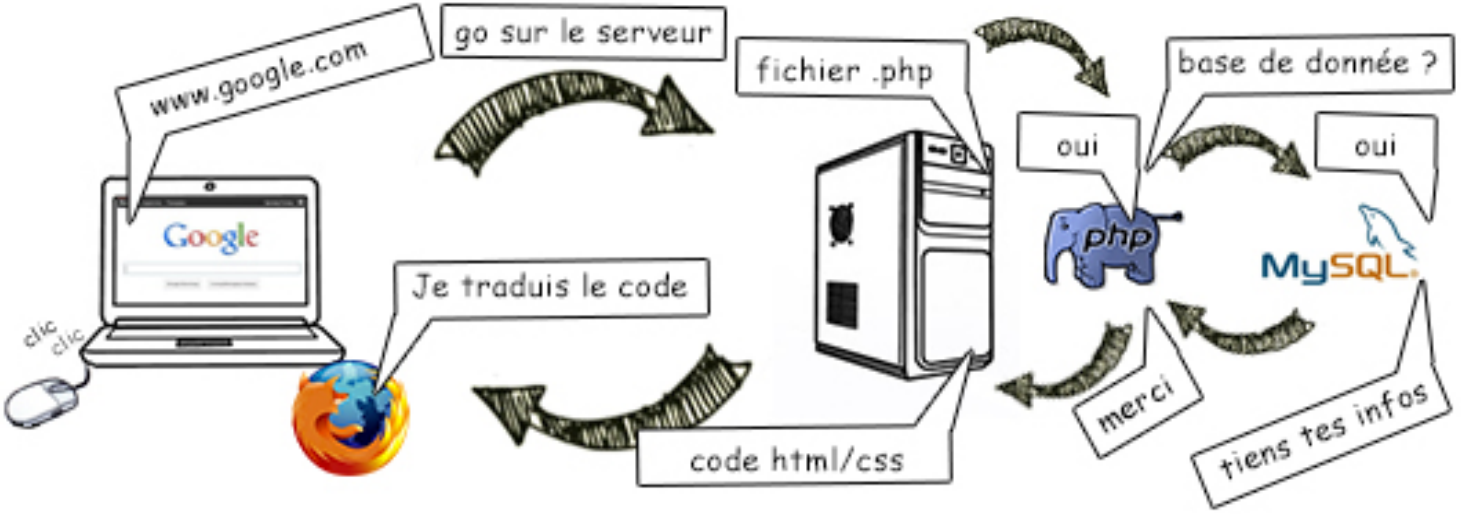
- Demande d'accès à une page web depuis un navigateur
- Le navigateur envoie une demande au serveur web comprenant :
  - le nom de la page, qui correspond en fait à un programme
  - éventuellement un ensemble de paramètres
- Le serveur web exécute le code source du programme :
  - en utilisant les paramètres transmis avec la demande du navigateur
  - en récupérant si besoin des données dans une BD ou sur le disque
  - le programme génère un contenu HTML
- Le contenu HTML est envoyé au navigateur
- Le navigateur affiche le résultat

Pages web dynamiques (3)



http://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_contenu

Pages web dynamiques (3)



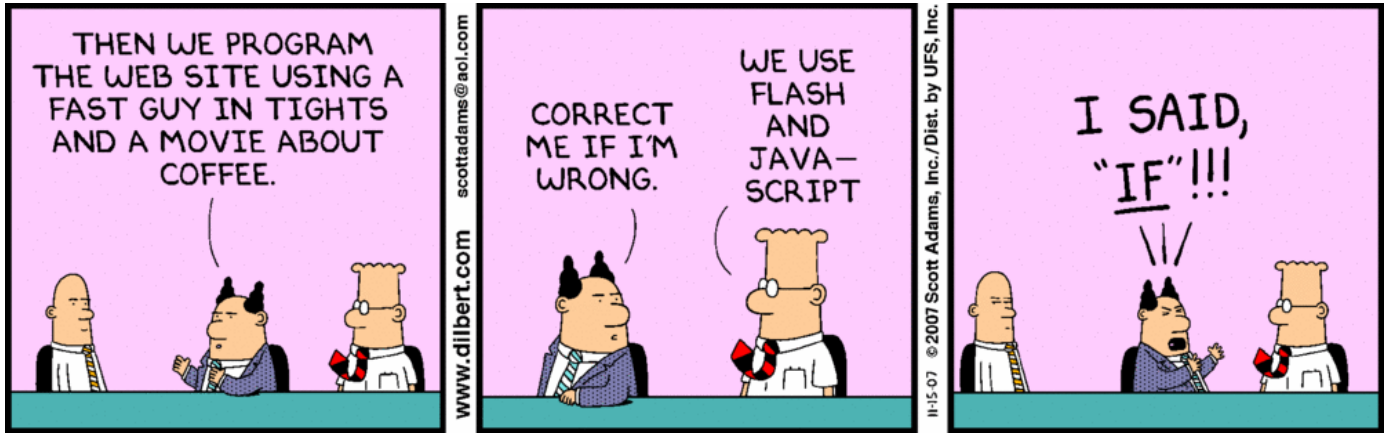
- Majorité de sites web dynamiques depuis les années 2000 (web 2.0, CMS, etc.)
- Théoriquement plus long pour afficher une page
- Besoin d'un langage de programmation côté serveur

http://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_contenu

Langages pour la programmation web

Quelques langages populaires :

- Pour la structure et le contenu des pages : HTML, XML
- Pour la présentation (mise en page / forme) : CSS
- Pour le comportement (interactions) : JavaScript, AJAX



Langages pour la programmation web (2)

- Pour des animations : Flash (à éviter)
- Pour générer le contenu (de pages dynamiques) : PHP, Python, Java, Perl, JavaScript, ...
- Pour interroger une source de données : SQL, XPath, XQuery, SPARQL, ...

Dans cet enseignement : HTML, CSS et Python (et SQL !)

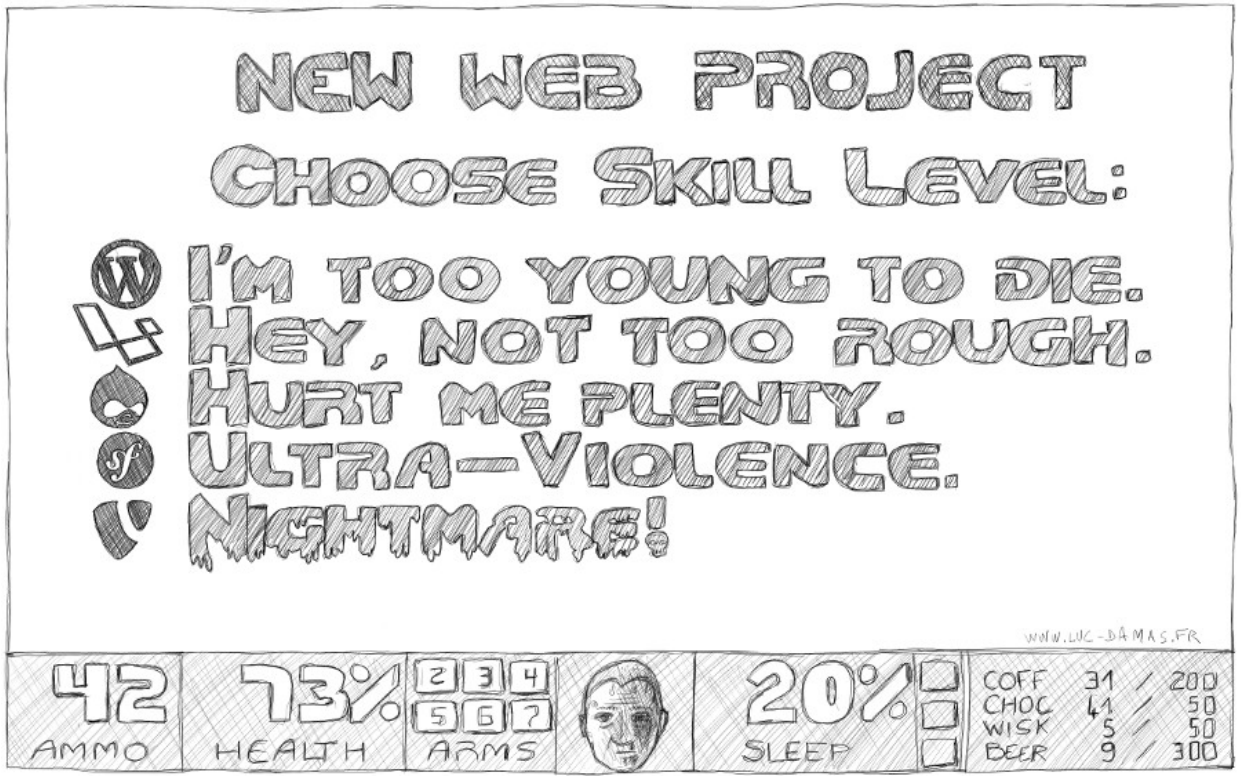
Outils pour la programmation web

- Éditeurs textuels (Emacs, BlueFish, VSCode, Brackets, Atom, ...) ou WYSIWYG (Dreamweaver, NVU, ...)
- CMS pour Content Management System (Drupal, Joomla, Spip, Wordpress, ...)
- Framework (Symfony, Laravel, Hibernate, ...)

---

http://fr.wikipedia.org/wiki/%C3%89diteur_HTML
http://fr.wikipedia.org/wiki/WYSIWYG
http://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_contenu

Outils pour la programmation web (2)



Dans cet enseignement : un éditeur textuel

http://www.luc-damas.fr/humeurs/

Plan

Syntaxe de base du HTML

Balises structurantes

Balises de contenu

Formulaires

Généralités

HTML pour HyperText Markup Language :

- Langage de balisage (structure + contenu)
- Origine : 1989-1990
- Toujours en développement (HTML5)
- Extension de fichiers : **.html**
- Standard W3C depuis 1996

Fichier HTML :

- Fichier texte contenant des informations de structuration
- C'est le navigateur qui lit le fichier et interprète les balises pour faire l'affichage

---

http://fr.wikipedia.org/wiki/Hypertext_Markup_Language
http://fr.wikipedia.org/wiki/Langage_de_balisage
http://www.w3.org/TR/html5/

Principe des balises

La structure est indiquée à l'aide de balises :

- Le nom de la balise indique le type d'élément que l'on ajoute (e.g., une image, un titre, un menu)

Il existe des balises en paires (une ouvrante et une fermante avec un slash)

```
<balise> du texte ou autre </balise>
```

Il existe des balises orphelines (une ouvrante seulement, avec recommandation de mettre le slash fermant)

```
<balise />
```

Principe des balises (2)

Une balise ouvrante peut contenir des *attributs* de la forme `nom="valeur"` :

- Les attributs permettent de préciser des informations concernant la mise en forme, propriétés de la balise, etc.

```
<balise nom-att1="val1" nom-att2="val2"> ...</balise>
```

Les balises peuvent être imbriquées (sans s'entremêler) :

```
<balise1> <balise2 /> <balise3> </balise3> </balise1>
```

Structure d'une page HTML

```
<!DOCTYPE html>
<html lang="fr">
 <head>
  <!-- métadonnées, scripts, etc. -->
  ...
 </head>
 <body>
  Partie affichée dans le navigateur
  ...
 </body>
</html>
```

Définition du type document

Balise **doctype** spécifie le type de document :

- Plusieurs versions (et variantes) de la norme HTML
- Il faut donc préciser laquelle on utilise (et s'y tenir)
- Obligatoire si l'on veut faire valider son document par un validateur
- Balise `doctype` simplifiée depuis HTML5 :

```
<!DOCTYPE html>
```

Balise **html** pour le début du document :

- Attribut `lang` pour la langue (synthèse vocale)

En-tête (partie <head>)

Partie **head** du fichier HTML pour :

- Des métadonnées sur le document, utilisables par les navigateurs ou moteurs de recherche :
  - Syntaxe d'une métadonnée :

```
<meta name="propriété" content="attributs">
```

  - Exemples de métadonnées :
    - la balise `title`, **obligatoire** (dans la norme)
    - des mots-clés
    - l'encodage du document (important pour afficher correctement les caractères accentués ou spéciaux)
    - le comportement des robots
- La définition de scripts (qui seront utilisés dans la partie `body`)
- Un lien vers une feuille de style ou des scripts externes

Exemple de fichier en-tête

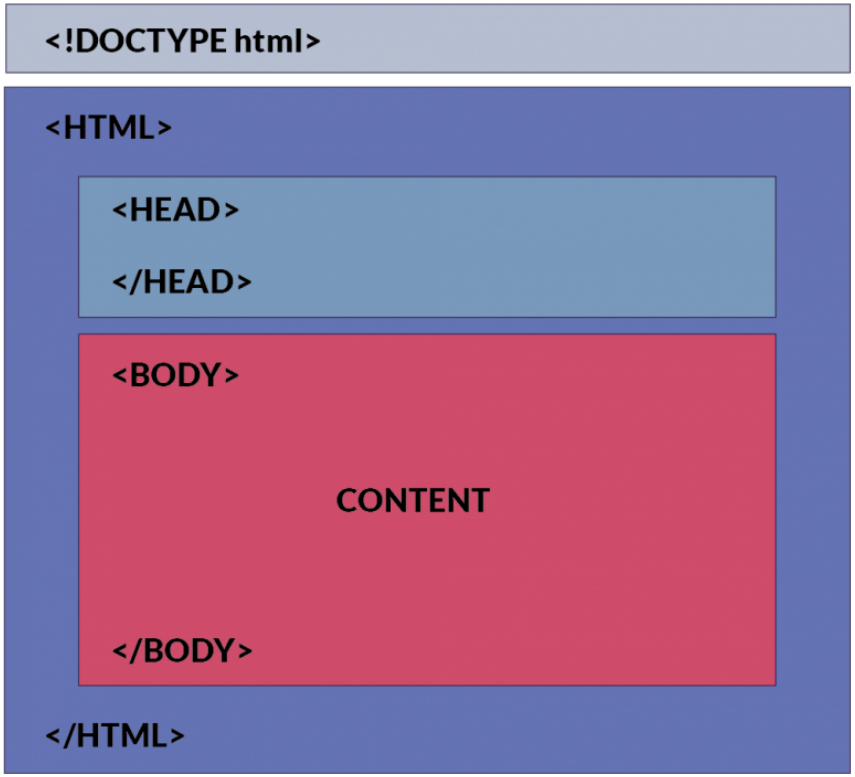
```
<!DOCTYPE html>
<html>
 <head>
  <meta charset="utf-8" />
  <title>Titre de la page</title>
  <link rel="stylesheet" type="text/css" href="includes/styles.css" />
  <meta name="keywords" content="cours HTML, HTML5, site web" />
 </head>
 <body>
  Partie affichée dans le navigateur
 </body>
</html>
```

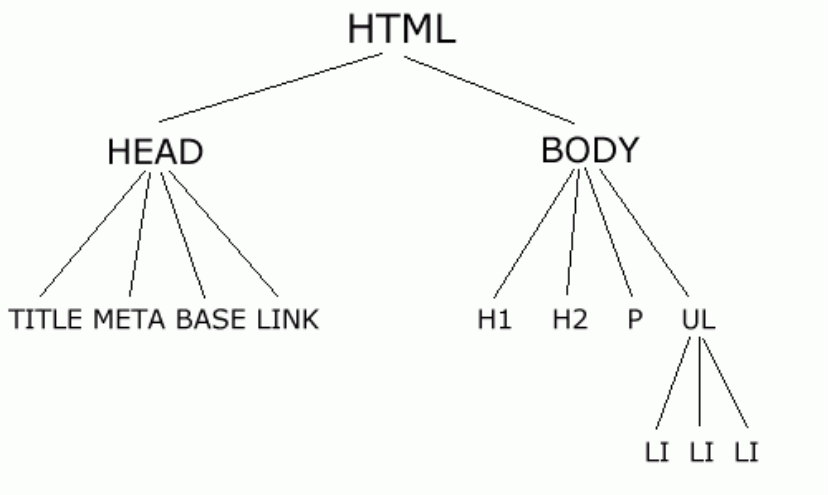
Quatre métadonnées dans la partie <head> : une spécification du codage, le titre de la page, un lien vers un fichier de style CSS et des mots-clés décrivant le document

Contenu (partie <body>)

Partie **body** du fichier HTML pour le contenu

- Organisation réalisée avec des balises structurantes, puis avec des balises de contenu





Bonnes pratiques

- Les navigateurs sont assez tolérants aux erreurs de programmation en HTML
- Aérer le code (mise à jour plus facile ⇒ soutenance de projet)
- Commenter le code, avec les balises <! − − −− >

```
<!-- ceci est un commentaire -->
```

- Balises en minuscules, éviter les balises hors standard
- Penser à l'accessibilité (e.g., attributs `alt`)
- Valider son fichier HTML avec un validateur

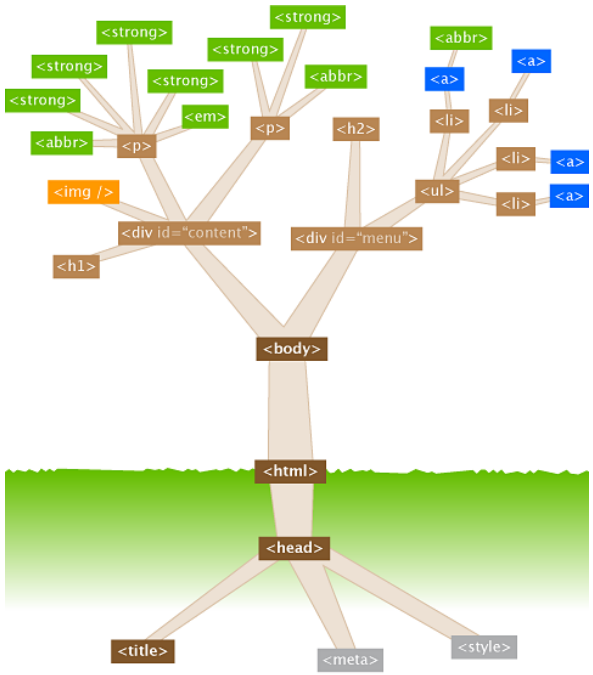
---

https://validator.w3.org/
https://caniuse.com/

En résumé

HTML, un langage de balises utilisé pour structurer le contenu :

- Partie head = métadonnées de la page
- Partie body = contenu de la page



http://watershedcreative.com/naked/html-tree.html

Plan

Syntaxe de base du HTML

Balises structurantes

Balises de contenu

Formulaires

Généralités

Depuis HTML5, certaines balises apportent de la sémantique, en particulier pour organiser le contenu d'une page :

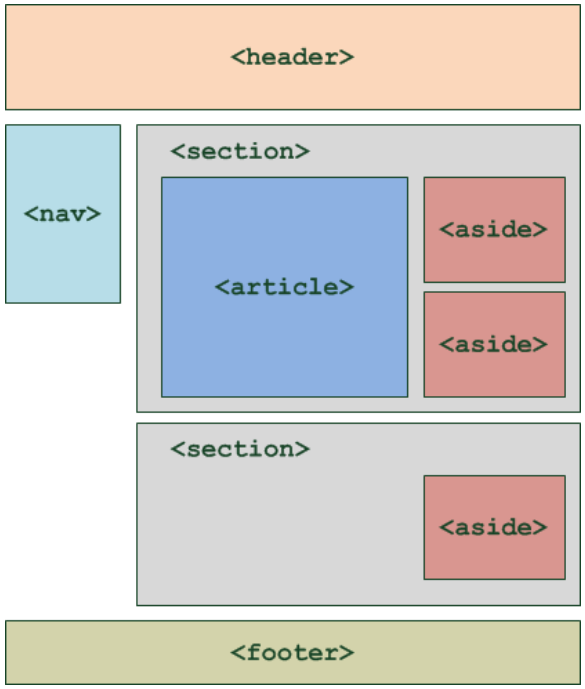
- `header`
- `footer`
- `nav`
- `main`
- `section`, `article`
- `aside`

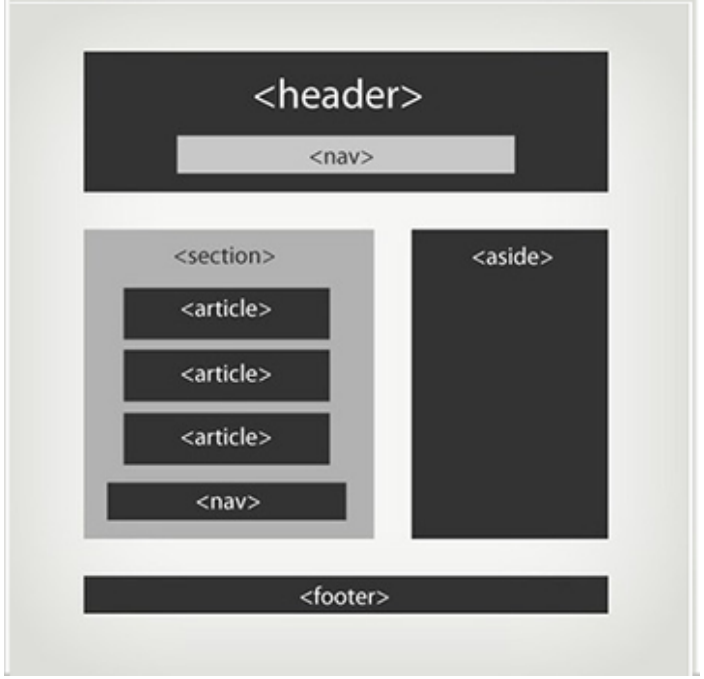
En général, certaines parties d'un site sont communes à toutes les pages du site (e.g., header, footer, nav)

---

http://www.vectorskin.com/referentiels-standards-w3c/balises-html5/
http://freehtml5templates.com/

Exemple d'organisation d'une page (design)





Deux exemples d'organisation de page avec ces balises structurantes. Attention, la mise en page et mise en forme se font ailleurs, avec le CSS !

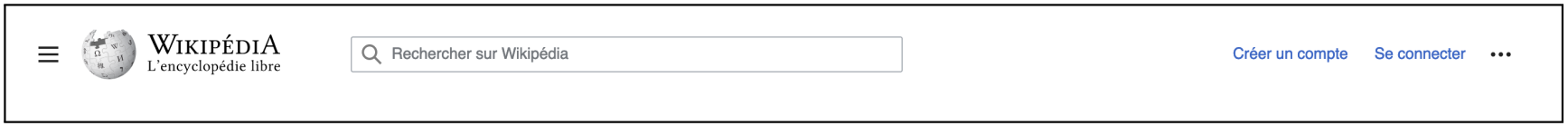
---

http://www.siteduzero.com/informatique/tutoriels/apprenez-a-creer-votre-site-web-avec-html5-et-css3/

Balise `header`

L'entête d'une page, située en haut, inclut souvent le nom du site, un slogan, un logo ou bannière. Il peut y avoir plusieurs entêtes (e.g., une par section)

```
<header>
  contenu de l'entête de la page
</header>
```



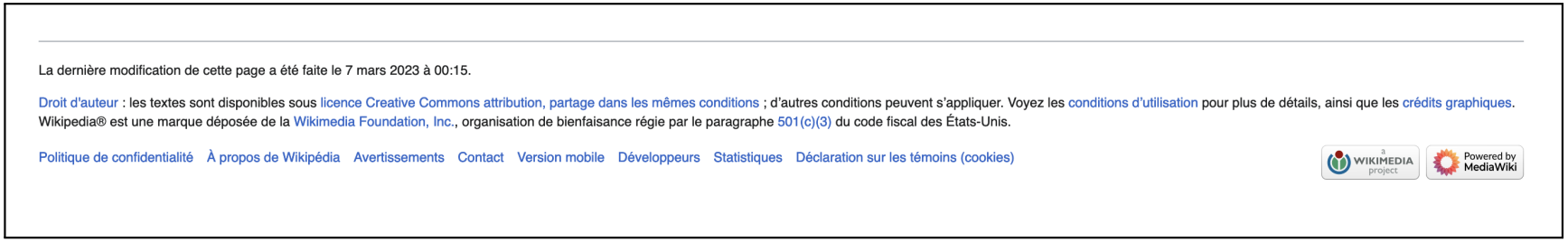
Un exemple d'entête (Wikipedia)

Ne pas confondre le `header` avec la partie `head` qui contient l'entête du document (métadonnées, scripts, etc.)

Balise `footer`

Le pied de page, situé en bas, inclut en général des crédits, des mentions légales, un lien de contact, etc.

```
<footer>
  contenu du pied de page
</footer>
```

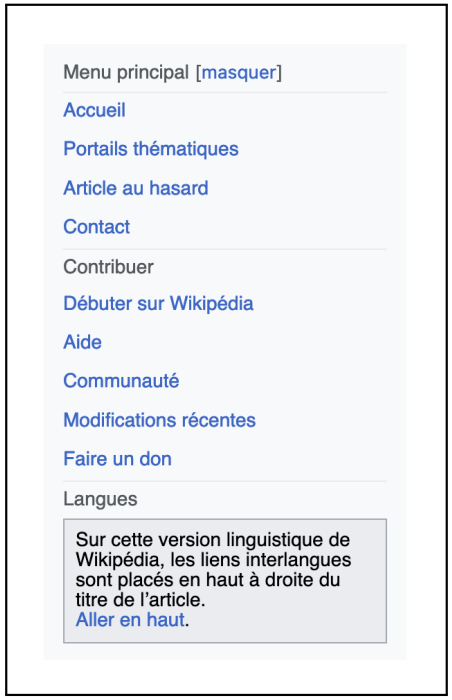


Un exemple de pied de page (Wikipedia)

Balise nav

La balise nav regroupe les principaux liens de navigation (menu principal)

```
<nav>
  contenu du menu principal
</nav>
```

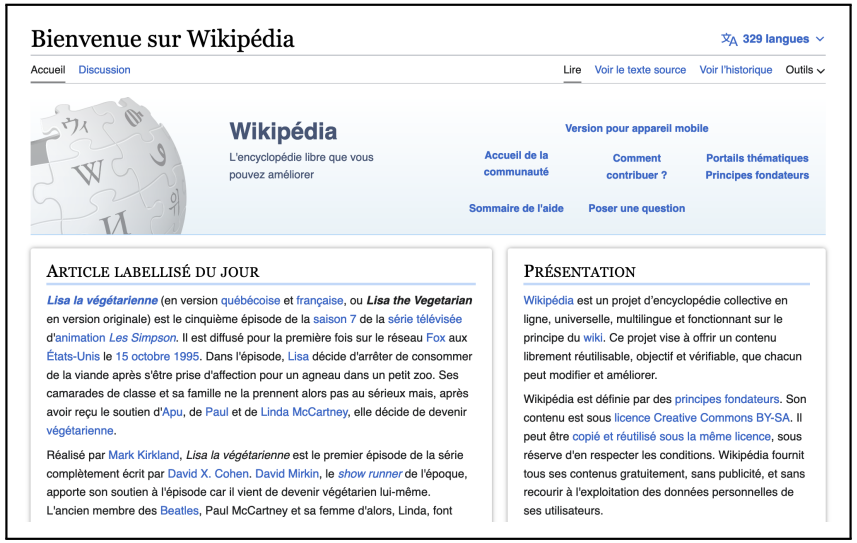


Un exemple de menu (Wikipedia)

Balise `main`

La balise `main` est utilisée pour le contenu principal d'une page

```
<main>
  <h1>Titre du contenu</h1>
  <section> contenu section 1 </section>
  <section> contenu section 2 </section>
</main>
```

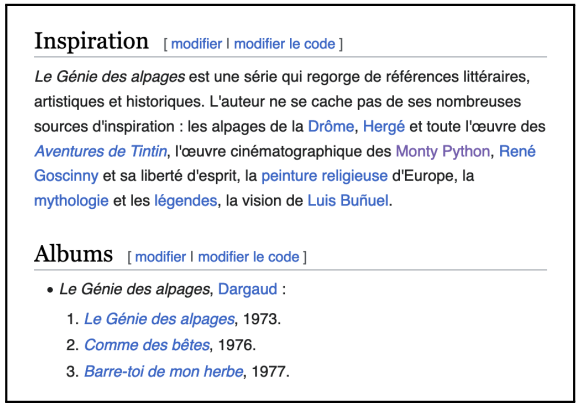


Un exemple de zone de contenu principal (Wikipedia)

Balises `section` et `article`

Balises utilisées pour découper le contenu principal. La balise `article` est indépendante (qui peut être reprise sur un autre site)

```
<section>
  <h1>Titre de la section</h1>
  <p> contenu </p>
</section>
```



Inspiration [modifier | modifier le code]

Le Génie des alpages est une série qui regorge de références littéraires, artistiques et historiques. L'auteur ne se cache pas de ses nombreuses sources d'inspiration : les alpages de la Drôme, Hergé et toute l'œuvre des *Aventures de Tintin*, l'œuvre cinématographique des Monty Python, René Goscinny et sa liberté d'esprit, la peinture religieuse d'Europe, la mythologie et les légendes, la vision de Luis Buñuel.

Albums [modifier | modifier le code]

- *Le Génie des alpages*, Dargaud :
  1. *Le Génie des alpages*, 1973.
  2. *Comme des bêtes*, 1976.
  3. *Barre-toi de mon herbe*, 1977.

IMAGE LABELLISÉE DU JOUR

Pêcheurs indiens de retour du travail sur une plage du Kerala.

(définition réelle 3 852 × 2 568)

Archives · Voir les images

Images de qualité sur Wikimédia Commons

Exemples de zones section et article (Wikipedia)

Balise aside

La balise aside est une zone d'informations complémentaires

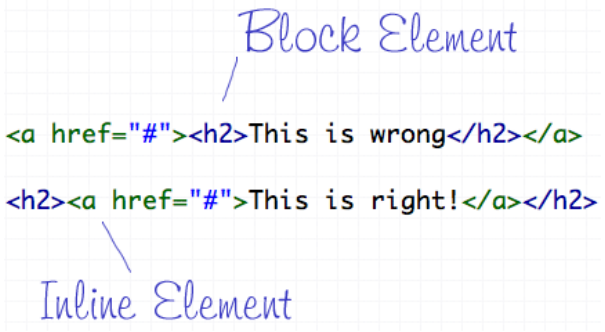
```
<aside>
  informations complémentaires
</aside>
```

Un exemple de zone aside

Balises de regroupement

Chaque balise possède un "type de rendu", parmi lesquels :

- Inline = éléments placés les uns à côté des autres (affichage horizontal)
- Block = éléments placés les uns en dessous des autres (affichage vertical)



Un élément inline peut être placé dans des éléments de type block, mais pas l'inverse

http://line25.com/articles/10-html-tag-crimes-you-really-shouldnt-commit

Balises de regroupement (2)

Pour regrouper des éléments (e.g., pour leur appliquer un style), deux balises de regroupement :

- `<div>` et `<span>`
- Balises neutres (pas d'effet visible sur la page) et sans sémantique
- Respect de l'affichage (block ou inline)

`<div>` regroupe des éléments de type bloc ou inline :

```
<div><p>un texte</p><p>et un second</p></div>
```

`<span>` regroupe des éléments de type inline :

```
<p><span>un texte <em> et la suite</em></span><p>
```

En résumé

- Réfléchir à la structuration du site en amont (conception)
- Privilégier l'utilisation des balises structurantes (e.g., `header`, `main`)
- Compléter la structuration avec les balises `div` et `span`

Plan

Syntaxe de base du HTML

Balises structurantes

Balises de contenu

Formulaires

Généralités

À l'intérieur des balises structurantes, le contenu est (encore) organisé par des balises de contenu :

- Titres
- Paragraphes
- Listes
- Tableaux
- Images
- Liens
- ...

Paragraphes

```
<p>
un texte qui s'étend <br>
sur plusieurs lignes<br>
et qui forme un paragraphe
</p>
```

- Le texte entre les balises `<p>` et `</p>` forme un paragraphe (saut de ligne avant et après le paragraphe)
- Les balises `<br/>` indiquent un saut de ligne dans le paragraphe

Texte important

Ces balises signalent un texte important, mais sont purement sémantiques : c'est le navigateur qui décide de leur appliquer un style (gras, italique), ce qui est personnalisable avec les CSS

```
un texte avec des <mark>mots très importants</mark>
```

```
un texte avec un mot <strong>important</strong> dedans
```

```
un texte avec des <em>mots moins importants</em>
```

Titres

Balises pour différents niveaux de titre, balises uniquement sémantiques !

- `<h1>un premier titre</h1>`
  c'est un titre très important (traduit par le navigateur par un affichage plus gros, en gras)
- `<h2>un second titre</h2>`
  c'est un titre important (traduit par le navigateur par un affichage un peu moins gros)
- ...
- `<h6>un sixième titre</h6>`
  c'est la balise pour un titre peu important (dernier niveau)

Listes

Syntaxe d'une liste sans numérotation :

```
<ul>
  <li>entrée 1</li>
  <li>entrée 2</li>
  <li>entrée 3</li>
  <li>...</li>
</ul>
```

- entrée 1
- entrée 2
- entrée 3
- ...

Syntaxe d'une liste avec numérotation :

```
<ol>
  <li>entrée 1</li>
  <li>entrée 2</li>
  <li>entrée 3</li>
  <li>...</li>
</ol>
```

1. entrée 1
2. entrée 2
3. entrée 3
4. ...

Tableaux

Un tableau est délimité par `<table>` et `</table>` :

- Balises `<tr>` pour créer une nouvelle ligne
- Balises `<th>` pour une cellule d'entête ou `<td>` pour une cellule de contenu

```
<table>
  <tr>
    <th>Colonne 1</th>
    <th>Colonne 2</th>
  </tr>
  <tr>
    <td>case 1</td>
    <td>case 2</td>
  </tr>
  <tr>
    <td>case 3</td>
    <td>case 4</td>
  </tr>

</table>
```

Colonne 1	Colonne 2
case 1	case 2
case 3	case 4

http://fr.wikibooks.org/wiki/Le_langage_HTML/Tableaux

Liens hypertextes

```
<a href="adresse_web">zone cliquable</a>
```

- L'attribut `href` spécifie l'adresse (URL, fichier, ancre) de destination
- Ce qui est entre les balises `<a>` et `</a>` (*zone cliquable*) devient un lien cliquable
- Autres attributs de la balise `<a>` :
  - `target="cible"` pour ouvrir l'URL dans *cible* (`target="_blank"` pour une nouvelle fenêtre)
  - `name="mon_ancre"` pour définir une ancre *mon_ancre* (endroit précis d'une page)

Liens hypertextes (2)

Caractéristiques de la destination (`href`) :

- Absolu : adresse_web a un chemin complet (obligatoire pour un lien situé sur un autre site web ou avec protocole différent)
- Relatif : adresse_web a un chemin défini par rapport à la page actuelle (préférable sur son site)

Exemples de liens hypertextes

****un lien interne, avec chemin relatif****

****un lien externe, avec chemin absolu****

****un lien externe, avec chemin absolu, qui s'ouvre dans une nouvelle fenêtre****

****un lien interne, pointant sur l'ancre *section3* de la page *page.html*****

Images

```
<img src="url_image" alt="texte alternatif" />
```

- Insertion de l'image située à l'adresse *url_image*
- Pas de balise fermante pour `<img>`
- Attribut `alt` conseillé (pour l'accessibilité)
- Balises `<figure>` et `<figcaption>` pour ajouter une légende

http://fr.wikibooks.org/wiki/Le_langage_HTML/Images
http://vidberg.blog.lemonde.fr/

Mise en forme du texte

Il existe des balises `<u>`, `<b>`, `<i>`, `<big>`, etc.

```
<b><i>un texte souligné et gras</i></b>
```

Mise en forme du texte

Il existe des balises `<u>`, `<b>`, `<i>`, `<big>`, etc.

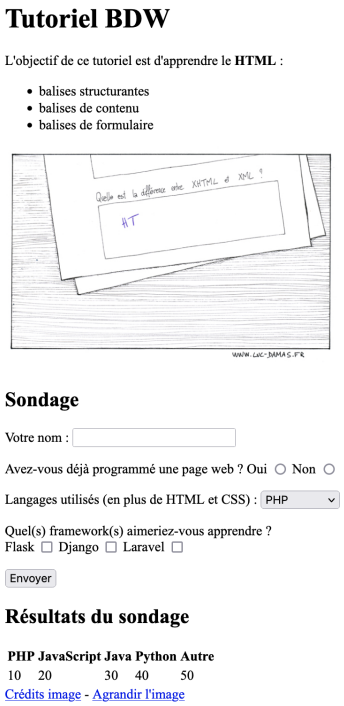
```
<b><i>un texte souligné et gras</i></b>
```

OBSOLETE

Elles sont obsolètes, il ne faut pas les utiliser !
⇒ La mise en forme sera réalisée avec les CSS

En résumé

- Balises principales pour organiser le contenu
- Autres balises pour le multimédia, les cadres, applets, etc. (> 100 balises et > 200 attributs en HTML5)



Tutoriel BDW

L'objectif de ce tutoriel est d'apprendre le **HTML** :

- balises structurantes
- balises de contenu
- balises de formulaire

Sondage

Votre nom :

Avez-vous déjà programmé une page web ? Oui Non

Langages utilisés (en plus de HTML et CSS) : PHP

Quel(s) framework(s) aimeriez-vous apprendre ?
Flask Django Laravel

Envoyer

Résultats du sondage

PHP	JavaScript	Java	Python	Autre
10	20	30	40	50

Crédits image - Agrandir l'image

Voir également le tutoriel web

Plan

Syntaxe de base du HTML

Balises structurantes

Balises de contenu

Formulaires

Généralités

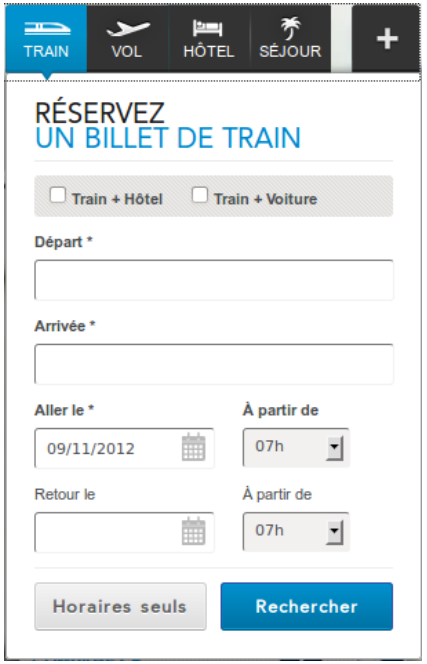
Objectifs des formulaires web :

- Permettre à l'utilisatrice de saisir des **paramètres**
- Envoyer les valeurs saisies pour chaque paramètre au serveur
- Accéder à une page dynamique générée par le serveur selon les valeurs des paramètres

Quels moyens ?

- Différents composants (champs textuels, listes déroulantes, cases à cocher, ...)
- À chaque composant correspond un paramètre

Généralités (2)



Des formulaires bien connus pour générer des pages dynamiques

Balise de formulaire

```
<form action="url" method="GET|POST" name="nomF">
 contenu_formulaire
</form>
```

- Déclaration d'un formulaire (sans composant)
- Un attribut `name` optionnel
- Une `action` sous forme d'URL vers le fichier qui sera exécuté après soumission du formulaire
- Le corps du formulaire `contenu_formulaire` contient les balises représentant les différents composants de saisie

Balise de formulaire (2)

- Valeurs pour l'attribut `method` (méthode de transmission des paramètres) :
  - `POST` : données dans le corps de la requête HTTP
  - `GET` : données encodées dans l'URL de la page
    - à la fin de l'URL, on ajoute le caractère ?
    - puis pour chaque paramètre on ajoute `nom=val`
    - les paramètres sont séparés par le caractère `&`

Exemple d'URL avec deux paramètres passés par `GET` :
`http://www.monserveur.org/index.php?page=ajout&k=4`

Composants de formulaire

Composant (ou élément) de formulaire :

- Généralement défini par la balise `input`
- Type du composant (liste, case à cocher, etc.) spécifié par l'attribut `type` de la balise `input`
- Aussi des composants liste (balise `select`) et grand texte (balise `textarea`)
- Attributs recommandés d'un composant : `id` (identification dans la page) et `name` (récupération de la valeur par ce nom)

Labels

```
<label for="idDuComposant">Texte descriptif</label>
```

- Un label permet d'associer un texte descriptif (label) à un composant
- Le label *texte descriptif* décrit le composant identifié par `idDuComposant`
- Un clic sur le label donne le focus au composant pour permettre la saisie
- Important pour l'accessibilité

Composant de saisie de texte

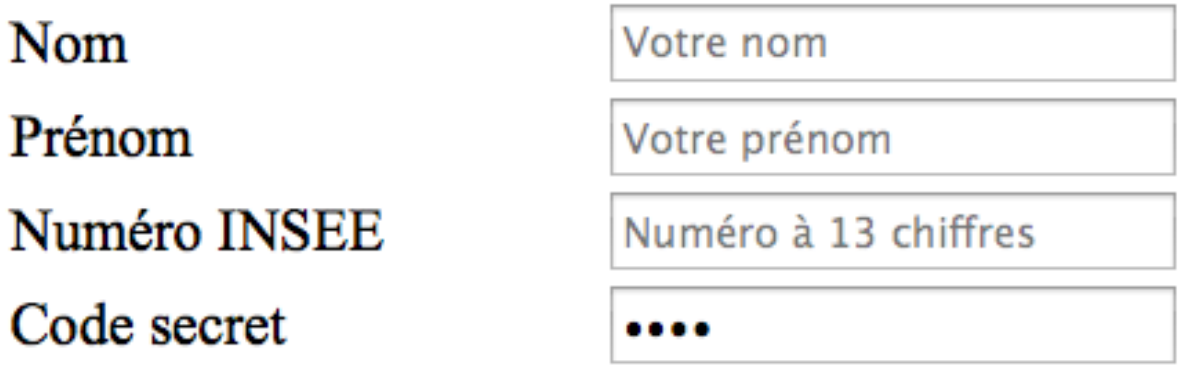
```
<input type="text" name="nom" id="idTexte" />
```

- Champ de saisie pour une ligne de texte
- L'attribut `name` précise le nom du composant (ici *nom*)
- L'attribut `id` indique l'identifiant du composant (ici *idTexte*, la valeur à utiliser dans l'attribut `for` de `<label>`)
- Attributs optionnels :
  - `size="un_nombre"` : la taille du champ en caractères
  - `value="une_valeur"` : texte pré-saisi
    - utile pour les opérations de modification
- Pas de balise fermante

Composant de saisie de mot de passe

```
<input type="password" name="nom" />
```

- Même composant que le champ texte, mais les caractères saisis sont remplacés par des '⋆' (typiquement utilisés pour saisir un mot de passe)



Composant de saisie d'un grand texte

```
<textarea name="nom" rows="h" cols="l">
 contenu pré-saisi
</textarea>
```

- Champ de saisie pour du texte sur plusieurs lignes
- L'attribut `name` précise le nom du composant (ici *nom*)
- Les attributs `rows` et `cols` précisent la hauteur et la largeur du composant en nombre de lignes et nombre de caractères respectivement (ici *h* et *l*)
- Le contenu pré-saisi peut être vide et ne contient pas de balise

Paramètres cachés

```
<input type="hidden" name="nom" value="val" />
```

- Un composant caché n'est pas affiché (mais visible dans le code donc peu sécurisé)
  - utile pour spécifier un identifiant dans un formulaire de modification des informations de la base
- Le paramètre caché a la valeur spécifiée par l'attribut `value` (ici *val*)

Listes déroulantes

```
<select name="nom">
  <option value="val1">Texte 1</option>
  <option value="val2">Texte 2</option>
  ...
</select>
```

- Liste déroulante avec les choix possibles `Texte 1`, `Texte 2`, ...
- L'attribut optionnel `value` indique la valeur transmise (éventuellement différente de son texte)
- Par défaut, la valeur est égale au texte de la balise `option`
- Une balise `option` peut être pré-sélectionnée en lui ajoutant l'attribut `selected="true"`

Case à cocher

```
<label for="id1">Texte 1</label><input type="checkbox"
name="nom" id="id1" value="val1" />
<label for="id2">Texte 2</label><input type="checkbox"
name="nom" id="id2" value="val2" />
```

- Deux cases à cocher portant sur le même paramètre (même valeur pour leur attribut `name`), ayant des valeurs respectives *val1* et *val2*
- Les cases portant le même *nom* peuvent être **toutes cochées** à un moment donné
- Attribut optionnel `checked` pour pré-sélectionner une case
- Utilisation recommandée des balises `<label>`

Transport(s) utilisé(s) Vélo ☐ Train ☑ A pied ☐

Boutons radio

```
<label for="id1">Texte 1</label><input type="radio"
name="nom" id="id1" value="val1" />
<label for="id2">Texte 2</label><input type="radio"
name="nom" id="id2" value="val2" />
```

- Deux boutons radio portant sur le même paramètre (même valeur pour leur attribut `name`), ayant des valeurs respectives *val1* et *val2*
- Parmi les boutons radio portant le même nom, **un seul coché** à un moment donné
- Attribut optionnel `checked` pour pré-sélectionner un bouton
- Utilisation recommandée des balises `<label>`

Sexe Homme ○ Femme ◉

Boutons de soumission

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

- Bouton déclenchant le chargement de la page de destination (attribut `action` de la balise `form`)
- *texte* est le texte affiché sur le bouton

```
<input type="reset" value="texte" />
```

- Bouton déclenchant la réinitialisation du formulaire, en utilisant les valeurs pré-saisies lorsqu'elles existent
- *texte* est le texte affiché sur le bouton

Autres éléments de formulaires

Nouveaux types de la balise `<input>` en HTML5 :

- `tel, url, email`
- `search`
- `datalist` (auto-complète un champ grâce à une liste de valeurs prédéfinies)
- `date, time, datetime, week, month`
- `number`
- `range` (barre de progression)
- `color`

Nouvelles balises en HTML5 :

- `<output>` (somme d'un calcul)
- `<keygen>` (génération de clés de cryptage)

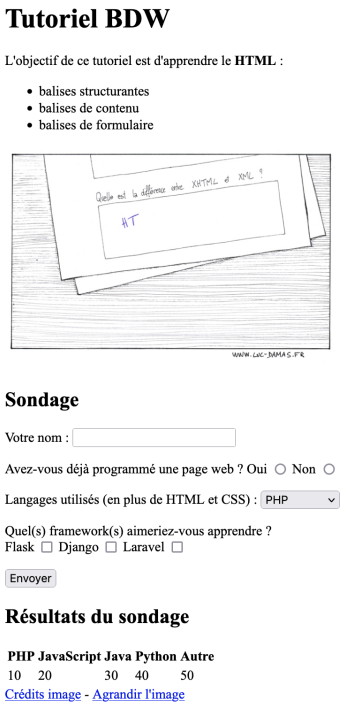
Autres attributs en HTML5

- `required` : le formulaire ne peut être validé si un de ses composants avec attribut `required` n'est pas rempli
- `pattern` : vérifier qu'un champ respecte une expression régulière donnée (e.g., pattern pour *url*)
- Ajout de sémantique pour l'attribut `rel` (valeurs *stylesheet*, *next*, *author*, *etc.*)
- ...

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

En résumé

- Dans un formulaire, chaque élément de saisie devient un paramètre (nom donné par l'attribut `name`)
- Définition d'une page *traitement* (attribut `action`) et d'un mode de transmission des paramètres (attribut `method`)



Tutoriel BDW

L'objectif de ce tutoriel est d'apprendre le **HTML** :

- balises structurantes
- balises de contenu
- balises de formulaire

Sondage

Votre nom :

Avez-vous déjà programmé une page web ? Oui Non

Langages utilisés (en plus de HTML et CSS) : PHP

Quel(s) framework(s) aimeriez-vous apprendre ?
Flask Django Laravel

Envoyer

Résultats du sondage

PHP	JavaScript	Java	Python	Autre
10	20	30	40	50

Crédits image - Agrandir l'image

Voir également le tutoriel web

Un moment de réflexion

- Avantages et inconvénients des méthodes `GET` ou `POST` pour transmettre les données d'un formulaire ?

Un moment de réflexion

- Avantages et inconvénients des méthodes `GET` ou `POST` pour transmettre les données d'un formulaire ?

- Que peut-on utiliser pour agencer les éléments d'une page web ? Frames ? Tableaux ? Autre ?

Un moment de réflexion

- Avantages et inconvénients des méthodes `GET` ou `POST` pour transmettre les données d'un formulaire ?
- Que peut-on utiliser pour agencer les éléments d'une page web ? Frames ? Tableaux ? Autre ?
- Quels types de métadonnées connaissez-vous au niveau des BD ?

Bilan

Utilisation de HTML pour structurer et afficher le contenu d'un site web :

- Réflexions sur l'organisation du site (balises structurantes)
- Balises principales de contenu et éléments de formulaire
- Pas de mise en forme ou de mise en page en HTML !

Bilan

Utilisation de HTML pour structurer et afficher le contenu d'un site web :

- Réflexions sur l'organisation du site (balises structurantes)
- Balises principales de contenu et éléments de formulaire
- Pas de mise en forme ou de mise en page en HTML !

Prochain cours :
l'habillage du contenu avec CSS