



- 1. HTML (« *Hypertext Markup Language* »)

- **Généralités**

- » Langage à **balises** conçu pour représenter des **pages web**
      - Inspiré de SGML et XML (langages de description de données)
    - » Permet l'écriture d'**hypertextes**
    - » A la base, il s'agit d'un **langage statique**
    - » Peut être interprété par de nombreux outils, notamment les **navigateurs**
  
    - » Décrit la **sémantique** des pages web
      - Sémantique = signification des différents éléments d'une page
      - Ex: titre, paragraphe, image, etc.
      - Interopérabilité des applications (« **web sémantique** »)
        - Ex: référencement Google
  
    - » Aspect visuel (mise en page) traité par **CSS**
    - » Aspect dynamique (traitements client) traités par **Javascript**

La première ligne indique le type de document (ici, HTML 5)

Document Objet  
Model (DOM)

Structure d'un document HTML

Source HTML	Modèle du document
<pre> &lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;title&gt;       Exemple de HTML     &lt;/title&gt;   &lt;/head&gt;   &lt;body&gt;     Ceci est une phrase avec un &lt;a href="cible.html"&gt;hyperlien&lt;/a&gt;.     &lt;p&gt;       Ceci est un paragraphe où il n'y a pas d'hyperlien.     &lt;/p&gt;   &lt;/body&gt; &lt;/html&gt; </pre>	<pre> html ├── head │   └── title │       └── texte └── body     ├── a     │   └── texte     ├── texte     └── p         └── texte </pre>



## – Quelques balises HTML courantes (types sémantiques)

- » Paragraphe: <p>
- » Texte gras (bold): <b>
- » Retour à la ligne: <br>
- » Titres de niveaux 1, 2, 3, 4, 5 ou 6
  - <h1>, <h2>, <h3>, <h4>, <h5>, <h6>
- » Image: <img>
- » Hyperlien: <a>
- » Listes ordonnées (<ol>) ou non-ordonnée (<ul>) avec éléments <li>
- » Tableau (<table>) avec lignes (<tr>) et colonnes (<td>)
- » <div>: pas de valeur sémantique mais utilisé comme support du CSS ou du Javascript
- » Formulaire <form>

## – Paramètres de balises

- » En plus de leur contenu, les balises peuvent recevoir des paramètres (ou attributs)
- » Exemple:

```
<a id='lien1' name='geo' class='lien' href='www.geomatics.ulg.ac.be'>  
    Lien vers le site de l'Unité de Géomatique  
</a>
```

## – Balises META

- » Métadonnées de la page à placer dans la balise <head>
- » Destinées aux navigateurs, aux moteurs de recherche et à tous les outils d'indexation
- » Exemple

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

→ Codage des caractères

```
<meta name="description" content="Une description de la page..." >
```

→ Description de la page

```
<meta name="viewport" content="width=device-width,  
initial-scale=1">
```

→ Indispensable pour les pages « responsives »

## – Formulaire

- » Envoi d'informations (« **inputs** ») encodées par l'utilisateur **vers un serveur**
- » Introduit par la balise **<form>**
- » **<form>** compte deux paramètres obligatoires
  - **Action**: URL (absolu ou relatif) vers lequel le formulaire est envoyé
  - **Method**: méthode d'envoi des données
    - Get
      - Données transmises dans l'URL:  
<http://www.exemple.com?nom=kasprzyk&nom=jean-paul>
      - Max 256 caractères
      - Pour données non-sensibles
    - Post:
      - Paquet attaché arrivant au serveur
      - Pas de limitation de taille
      - Non-visible dans URL (mais quand-même piratable)
  - **Exemple**: `<form action='traitement.php' method='get'>...</form>`

## – Formulaire et inputs

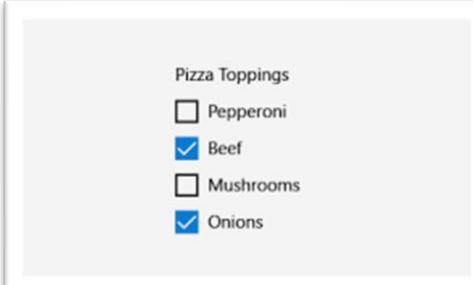
- » Les **inputs** sont des éléments html permettant à l'utilisateur **d'encoder des données** au sein d'un formulaire
- » Plusieurs types d'input existent (attribut « **type** »):

- 'text': Texte libre

– Exemple:

```
<input type='text' name='prenom' id='prenom' value='Texte par défaut'></input>
```

- 'password': texte « caché »
- 'number': envoi d'un nombre
- 'mail': texte avec un « @ »
- 'file': upload d'un fichier
- 'checkbox': case(s) à cocher
- 'radio': un choix possible parmi plusieurs options
- 'submit': bouton d'envoi du formulaire
- 'hidden': élément caché à l'utilisateur
- ...



Pizza Toppings

- Pepperoni
- Beef
- Mushrooms
- Onions



Male  Female

– **Formulaire et menu déroulant**

- » Menu déroulant = objet HTML « **select** »
- » Choix d'une **option** dans une liste
- » Peut servir, par exemple, à filtrer les entités d'une carte

HTML

CSS

```
1 <label for="pet-select">Choose a pet:</label>
2
3 <select name="pets" id="pet-select">
4   <option value="">--Please choose an option--</option>
5   <option value="dog">Dog</option>
6   <option value="cat">Cat</option>
7   <option value="hamster">Hamster</option>
8   <option value="parrot">Parrot</option>
9   <option value="spider">Spider</option>
10  <option value="goldfish">Goldfish</option>
11 </select>
12
```

Output

Choose a pet:

Hamster

--Please choose an option--

Dog

Cat

Hamster

Parrot

Spider

Goldfish

– Exemple de formulaire avec ses inputs

```
<form name="nom_formulaire" action="traitement" method="post">  
  Nom: <input name='nom' id='nom' type="text"></input><br>  
  Prénom: <input name='prenom' id='prenom' type="text"></input><br>  
  Fonction:  
    <select name="fonction" id='fonction'>  
      <option value="etudiant" selected>Etudiant</option>  
      <option value="chercheur">Chercheur</option>  
      <option value="professeur">Professeur</option>  
    </select><br>  
  <input type="submit" value="envoyer"></input>  
</form>
```

Nom:

Prénom:

Fonction:

## – Evolution vers HTML5

- » Aujourd'hui, **HTML 5** est le nouveau standard des navigateurs
- » Plus **dynamique** que le HTML de base
- » Nouvelles balises pour la **sémantique**
  - <nav>, <header>, <footer>, <figure>, <datalist>, <button>, ...
- » Nouvelles balises pour les **applications web** et **l'interactivité**
  - <video>, <audio>, <meter>, <progress>, ...
- » Nouveaux **types** pour les inputs de formulaires: « tel », « date », « number »
- » Nouveaux **attributs** pour les inputs de formulaires: « required », « min/max », « maxlength »
- » Nouvelles **API javascript**
  - **Autocomplete**: affichage de suggestions sur base des premiers caractères rentrés dans un champ (exploite la balise <datalist>)
  - **ContentEditable**: rend un élément html éditable par l'utilisateur
  - **Geolocation**
    - Permet l'exploitation de la position de l'utilisateur
    - Sur base du wifi et/ou de la 4G et/ou du récepteur GNSS de l'appareil (smartphone, tablette)
    - Nécessite un serveur web HTTPS et l'autorisation de l'utilisateur

» Exemple d'utilisation de la balise <meter>

```
1 | <p>Chauffez le four à  
2 |   <meter min="100" max="250" value="180">180 degrés</meter>.  
3 | </p>
```

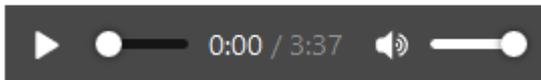
## Résultat

Chauffez le four à  .

» Exemple d'utilisation de la balise <audio>

```
<p>Neutrino</p>  
<audio controls="controls" controlsList="nodownload">  
  <source src="mp3/neutrino.mp3" type="audio/mp3" />  
  Votre navigateur n'est pas compatible  
</audio>
```

Neutrino





- 2. CSS (« Cascading Styles Sheets »)

- **Rôle**

- » Langage décrivant la présentation (mise en page) des documents HTML et XML

- **Trois endroits où placer du code CSS**

- » Directement **dans une balise HTML** via le paramètre « style »

- ```
<p style="...">...</p>
```

- » Dans la partie **<head>** de la page HTML

- ```
<head>
```

- ```
<style>
```

- ```
/* on place les styles ici */
```

- ```
</style>
```

- ```
</head>
```

- » Dans un **fichier CSS** externe

- On utilise la balise `<link />` dans la partie `<head>...</head>` pour faire le lien avec ce fichier CSS

- Méthode recommandée pour un site pro

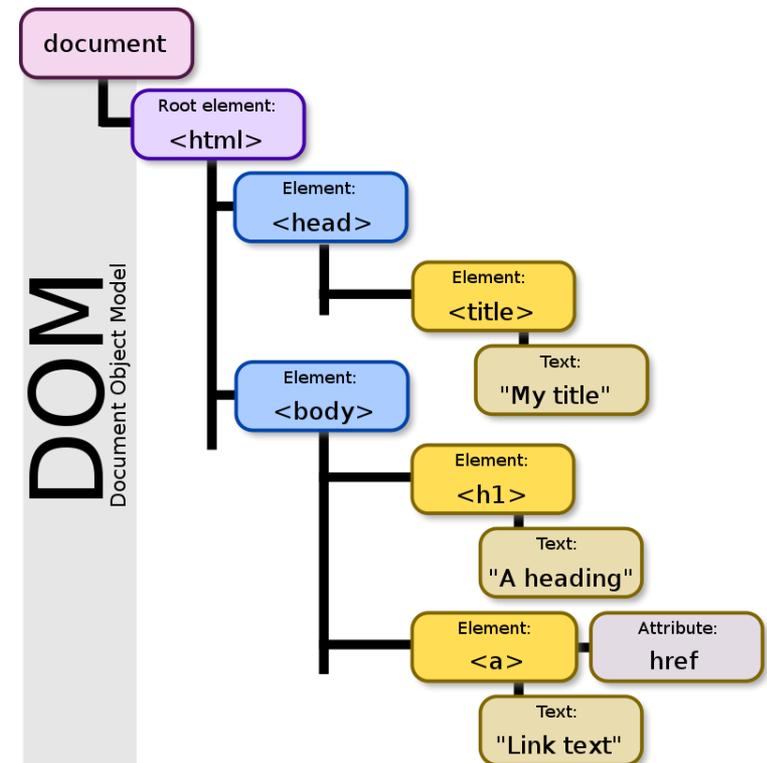
- ```
<head>
```

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

- ```
</head>
```

## – DOM (« Document Object Model »)

- » Les **balises** d'un document HTML forment une **arborescence** pouvant être exploitée par le code CSS ou Javascript (JS)
- » Un **élément** du document peut être **identifié par**:
  - Son ID ou son « name » (attribut de balise « id » ou « name »)
  - Son appartenance à une classe (attribut de balise « class »)
  - Son élément parent
  - Son élément enfant
- » Notion d'**héritage CSS** entre parent et enfants
  - Exemple: tous les enfants du <body> héritent de ses propriétés CSS (mais aussi ses petits enfants, etc)



**– Les sélecteurs CSS**

- » Un sélecteur permet l'identification d'un ou plusieurs éléments html pour lui/leur définir un style
- » Utilisé dans un fichier css externe ou dans la partie <head> du document
  - Jamais dans une balise du code HTML!
- » Syntaxe générale
  - Si le style porte sur un objet (ou une classe)

```
selecteur {  
    propriété1: valeur1;  
    propriété2: valeur2;  
}
```

- Si le style porte sur plusieurs objets (ou classes)

```
selecteur1, selecteur2 {  
    propriété1: valeur1;  
    propriété2: valeur2;  
}
```

### – Les trois types de sélecteur

» Sélecteur portant sur un **type** de balise

• Ex: `h1{color:red;}`

➔ Toutes les balises `<h1>` seront affichées en rouge

» Sélecteur portant sur une balise **unique**

• Ex: `#para1{color:red;}`

➔ La balise avec l'identifiant (id) `#para1` sera affichée en rouge

» Sélecteur portant sur une **classe**

• Ex: `.texte-rouge{color:red;}`

➔ Toutes les balises de classe « `texte-rouge` » seront affichées en rouge

### – Combinaison de sélecteurs et DOM

» Ex: `p.texte-rouge{...}`

➔ Toutes les balises `<p>` de classe « `texte-rouge` »

» Ex: `p .texte-rouge{...}`

➔ Toutes les balises de la classe « `texte-rouge` » compris dans une balise `<p>`

» Ex: `ul#monMenu .texte-rouge{...}`

➔ tous les éléments de la classe « `texte-rouge` » et qui se trouvent dans le `<UL>` identifié par 'monMenu'

**– Quelques propriétés CSS utiles**

- » Background: définition de la couleur de fond ou d'une image de fond
- » Border: bord de la balise
  - Ex: border : rgb(0,0,0) 3px solid → bord noir de 3 pixels de large
- » Display: affichage de la balise
  - Display: none → élément non-affiché
  - Display: block → retour à la ligne après l'élément
  - Display: inline → pas de retour à la ligne après l'élément
- » Height et Width: respectivement largeur et hauteur de l'élément
- » Max-width et min-width: respectivement largeurs maximales et minimales
- » Margin: taille de la marge extérieur
- » Padding: taille de la marge intérieure
- » Font-family: police (ex: verdana, arial, ...)
- » Color: couleur de la police
- » Font-size: taille de la police

**– Unités de taille**

- » Pixels: px
- » Unité de typographie: em
- » Pourcents: %

## – « Responsive Design »

### » Définition

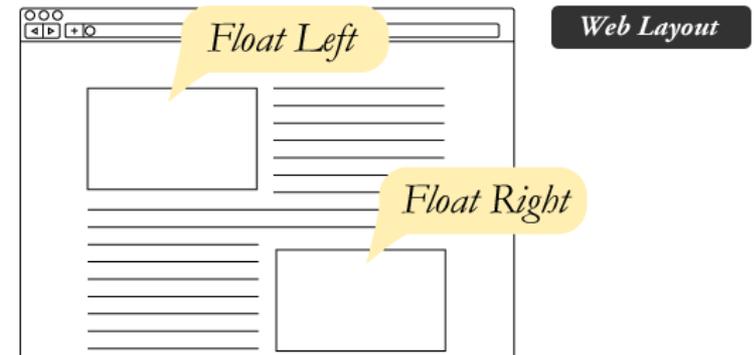
- Une page web « responsive » est une page dont le style est adapté à tous les types d'appareils: ordinateur, tablettes et smartphone

### » Contexte

- Actuellement, plus de 50% de l'usage d'internet se fait sur mobile
- Google prévoit de ne plus référencer les sites « responsive »

### » Technique CSS de base pour créer une page responsive

- Utilisation de **tailles relatives** (%)
- Utilisation de « **media queries** »
- Utilisation de **balises flottantes** (paramètre « float »)
  - Les éléments « s'empilent » par la gauche ou par la droite
  - Ex: `<div style='float:left'></div>`



## – Bootstrap

- » Un **framework** est une « surcouche » dans un environnement de programmation permettant d'en simplifier son utilisation générale
- » Bootstrap est un framework open source simplifiant la programmation « frontend » d'une page web en proposant des classes CSS/Javascript pour:
  - le style d'éléments HTML simples (boutons, menus déroulants, ...)
  - les menus « dropdowns »
  - les barres de navigation
  - les carousels
  - Le « **responsive design** » (grid)
  - ...
- » Installation sous formes de librairies CSS et Javascript
  - Librairie CSS à inclure dans le <head> du document HTML (fichier css)
  - Librairies Javascript à inclure à la fin du <body> du document HTML
- » Un des projets les plus populaires de la plate-forme de gestion de développement **GitHub**
- » <https://getbootstrap.com/>

## – Le système « grid » de Bootstrap

.col-12 .col-md-8		.col-6 .col-md-4
.col-6 .col-md-4	.col-6 .col-md-4	.col-6 .col-md-4
.col-6		.col-6

```

<div class="container">
  <!-- Stack the columns on mobile by making one full-width and the other half-width -->
  <div class="row">
    <div class="col-12 col-md-8">.col-12 .col-md-8</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
  </div>

  <!-- Columns start at 50% wide on mobile and bump up to 33.3% wide on desktop -->
  <div class="row">
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
  </div>

  <!-- Columns are always 50% wide, on mobile and desktop -->
  <div class="row">
    <div class="col-6">.col-6</div>
    <div class="col-6">.col-6</div>
  </div>
</div>

```

Copy