

Corps du document html

- Le corps d'un document html est défini par la balise `<body>...</body>`
- Il peut être structuré comme suit:
 - En-tête: `<header>..</header>`
 - Menu de navigation: `<nav>..</nav>`
 - Contenu principal: `<main>...</main>`
 - Section: `<section>..</section>`
 - Article: `<article>..</article>`
 - Pied de page: `<footer>..</footer>`
- Chacun de ces éléments peut avoir des titres, paragraphes, images, liens, tableaux, formulaires, etc.

Titres

- Définir des titres et sous titres (*headings*)
- Il existe 6 niveaux: **h1** à **h6**
- **<h1>** le plus **important**: les moteurs de recherche leur affectent un poids plus **élevé**
- **<h6>** le moins important
- Par défaut, les titres sont affichés en **gras**
- **Ex.**
 - <h1>** Titre de niveau 1 **<h1>**
 - <h3>** Titre de niveau 3 **<h3>**

Paragraphe

- ❑ **Paragraphe** `<p> ...</p>` permet d'ajouter du texte à la page
 - Par défaut, une ligne **avant** et **après** le paragraphe seront affichées par les navigateurs
 - Un paragraphe peut contenir du **texte**, des **images** et les autres éléments en **ligne** (``, ``, etc).
- ❑ **Séparateurs :**
 - `
` : retour à la ligne
 - `<hr>` : ligne horizontale

Formatage du texte

- **Gras** ou texte important: `...` ,
` ...`
- **Italic**: `...` , `<i></i>`
- Mise en **évidence**: `<mark>...</mark>`
- Mise en **indice** : `_{...}`
- Mise en **exposant** : `^{...}`
- **Citation**: `<cite>...</cite>`
- etc

Liens (ancres) (1/3)

- Un lien hypertexte vers un endroit précis d'une page ou vers une autre page se fait à l'aide de l'élément `anchor` `<a>... `
- L'attribut `href` ((**h**ypertext **r**eference) est utilisé pour définir l'adresse ou l'URL de la page cible

Ex.

`` lien vers un endroit précis d'une page (section) ``

`` lien vers une autre page ``

Liens (2/3)

Deux types d'URLs: **absolue** et **relative**

❑ **Absolute**: adresse complète:

protocole, **nom du domaine**, **chemin** si nécessaire

- Utilisée pour faire référence à une ressource **externe** (serveur ou site différent) ie lien externe
- **Ex.** `` Le site de l'université `` https => aller sur le web

Liens (3/3)

❑ **Relative**: chemin vers un fichier ie lien **interne**

- Sans le http, le navigateur regarde sur le serveur **local**
- Utilisée pour faire référence à une ressource interne (même serveur ou site)
 - Lien à l'intérieur d'un dossier
 - Ex. `Page d'accueil`
 - Lien vers un **sous** dossier (1 niveau)
 - Ex. `Aide`

Liens (3/3)

- Lien vers un dossier **parent** (1 niveau)
- Ex. `Page d'accueil`
- Lien à partir de la racine, on ajoute un `/`
- **Nb**: on peut mettre n'importe quel élément html entre `<a>` et `` pour le transformer en un lien
- Ex. Une image comme lien:
- ` `

Images

- L'insertion d'images se fait à l'aide de la balise ``
- Cet élément ne possède pas de balise fermante
- La balise `` possède les attributs suivants:
 - **src** : la **source** de l'image ou le chemin vers l'image (obligatoire)
 - **height**: **hauteur** de l'image (px, em, %, etc)
 - **width**: spécifie la **largeur** de l'image (px, em, %, etc)
 - **alt**: texte **alternatif** en cas d'échec de l'affichage ou chargement de l'image
- **Ex.** ``

Liste

- 03 types de listes :
- **Non ordonnée.** l'ordre des items n'est pas important.
- **Ordonnée.** l'ordre des items est important.
- **Définition ou description.** Elle consiste en paires (nom, valeur)

Liste non ordonnée

- Liste non ordonnée est définie avec la balise ``
- Cette balise possède des items ``
- Les items `` **ne** sont **pas** ordonnés (précédés par des puces par défaut) => l'ordre n'est pas important
- Seul des `` sont permis entre `` et ``
- **Ex.**

```
<ul>
```

```
  <li> HTML</li>
```

```
  <li> CSS </li>
```

```
  <li> JavaScript </li>
```

```
</ul>
```

Liste ordonnée

- Liste ordonnée est introduite avec la balise ``
- Liste où l'ordre est **important** (Ex. recette de cuisine, liste des étudiants admis selon l'ordre de mérite, etc)
- À la place des puces, le navigateur insère des nombres (par défaut) ou des lettres
- Les items `` sont ordonnés avec des chiffres ou lettres alphabétiques
- Ex. `<ol type="a" >`
 ` HTML ` => Liste avec des lettres
 ` CSS `
 ``

Remarque

- Si l'on souhaite que la liste commence à partir d'un nombre qui est différent de "1", on utilise l'attribut **start** comme suit:

- Ex. `<ol start= "5 ">` // commence à partir de 5

` HTML `

` CSS `

` JavaScript `

``

Liste de description (1/2)

- Liste de *description* est définie avec la balise `<dl>`
`</dl>`
- Un élément `<dl>` contient un certain nombre de d'éléments `<dt>` et leur `<dd>` respectifs
- Le **nom** ou le **terme** est introduit avec `<dt></dt>`
- La **valeur** ou la définition est introduite avec `<dd>`
`</dd>`

Liste de description (2/2)

- **Ex.** `<dl>`

`<dt> HTML </dt>`

`<dd> Langage de balisage </dd>`

`<dt> PHP </dt>`

`<dd> Langage de script </dd>`

`</dl>`

- **NB:** Pour un même terme `<dt>`, on peut avoir plusieurs définitions `<dd>`

Tableau (1/3)

- Pour créer un tableau, on utilise la balise `<table>`
- Cet élément possède une balise ouvrante `<table>` et fermante `</table>`
- En HTML, un tableau consiste en un ensemble de **lignes**
- Chaque ligne est introduite à l'aide de la balise `<tr>`
- `<th>`: permet d'insérer **l'entête** du tableau
- `<td>`: permet d'insérer les **données** du tableau

Tableau (2/3)

- Ex.

```
<table>  
  <tr>  
    <th> Nom </th>  
    <th> Age </th>  
  </tr>  
  <tr>  
    <td> Jean </td>  
    <td> 30 </td>  
  </tr>  
</table>
```

Tableau à 2 lignes et 2 colonnes

Nom	Age
Jean	30

Tableau (3/3)

- Attributs:
 - **colspan**: fusion de cellules d'une même ligne en définissant la valeur de l'attribut **colspan** d'un élément `<td>` ou `<th>` avec un entier
 - Cet entier indique le nombre de cellules à fusionner en partant de la gauche
 - Ex. `<td colspan="N">Contenu de la cellule</td>` N cellules
 - **rowspan**: fusion de cellules situées dans les lignes *adjacentes*
 - Ex. `<td rowspan="N">Contenu de la cellule</td>`

Formulaire (1/9)

- Un formulaire est introduit avec l'élément `<form>...</form>` et peut contenir des champs de **saisie**, des **boutons**, **textes**, etc
- Il permet de **collecter** les informations des utilisateurs
- Les informations collectées sont traitées par un **script** ou une application au niveau du serveur
- Cet élément peut contenir d'autres éléments en bloc mais **ne** peut pas contenir d'autres éléments `<form>`

Formulaire (2/9)

- ❑ L'élément `<form>` possède les attributs: `action`, `method`
 - `action=" URL "` l'adresse du script ou de l'application qui traitera les données saisies
 - `action=""`: Si le code PHP se trouve sur la page elle-même
 - Ex. `<form action="/script.php" >...</form>`

Formulaire (3/9)

- **method**: spécifie comment l'information devrait être envoyée au serveur. Il existe deux méthodes: **GET** et **POST**
- Cet attribut est **optionnel**. S'il n'est pas spécifié, la méthode **GET** est choisie par défaut

Formulaire (4/9)

□ GET vs POST

- **GET**: données saisies sont insérées dans l'URL après « ? »
 - Nombre de caractères à envoyer est limité
 - Données sont visibles à tous
- **POST**: données insérées dans une requête HTTP
 - Nombre de caractères est illimité
 - Sécurisé: seul le serveur est capable de lire les données

Formulaire (5/9)

- ❑ Types de **contrôles**: la plupart des éléments sont introduits avec l'attribut « **type** ». **L'apparence** de l'élément change en fonction de la valeur de « type »
- ❑ Plusieurs types de contrôles en HTML5:
 - Champs de texte
 - **Monoligne**: `<input type=" text " name=" nom " value=" Jean" placeholder='Text' maxlength=" 25" >`
 - **name**: nom de l'élément (obligatoire)
 - **value**: valeur par défaut lors du chargement de la page
 - **placeholder**: son contenu n'est pas envoyé
 - **maxlength** et **minlength**: nombre de caractères

Formulaire (6/9)

- **Multilignes**: l'élément `<textarea>` est utilisé lorsque l'utilisateur souhaite saisir plus d'une ligne
- **Ex.** `<textarea name= "adresse " placeholder=" pas plus de 50 caractères " rows= "50 " cols= "5">`
- *placeholder*: Son contenu ne sera pas émis au serveur
- *content*: Son contenu sera transmis au serveur

Formulaire (7/9)

- Champs de texte particuliers:
- Mot de passe: `<input type= " password " ...>`
Texte **invisible** mais ne veut pas dire chiffré
- Recherche: `<input type= "search " ...>`
- Email: `<input type= " email " ...>`
- Téléphone : `<input type= " tel " ...>`
- Adresse mail: `<input type= "url " ...>`

Formulaire (8/9)

- Bouton: `<input type= " button " ...>`
`<input type= " radio " ...>`
`<input type= " checkbox " ...>`
- Menu: `<datalist id="lang">`
`<option value="HTML">`
`<option value="CSS">`
`</datalist>`
`<select name="nom">`
`<option>HTML</option>`
`<option> CSS</option>`
`</select>`

Formulaire (9/9)

- Fichier: `<input type="file" ...>`
- Date et heure: `<input type="date" ...>`
`<input type="time" ...>`
- Couleur: `<input type="color" ...>`
- etc

Division (1/2)

- L'élément `<div></div>` est utilisé pour créer une division ou section dans un document html
- Cet élément **n'a pas** de signification propre en HTML mais utilisé conjointement avec le CSS
- C'est un élément par **bloc** ie débute et finit par un retour à la ligne
- **Ex.** `<div>` Voici une liste d'articles:

``

`` Ordinateur ``

`` Imprimante ``

``

`</div>`

Division (2/2)

- L'élément **span** est similaire à **div** sauf que celui-ci est un élément en **ligne** ie il ne débute pas et ne finit pas par un retour à la ligne
- **Ex.** Le premier est un `<div>` élément par bloc, `</div>` le second est un élément `` en ligne ``.
- **Résultat**
Le premier est un
élément par bloc
, le second est un élément en ligne.

Section (1/2)

- ❑ Elle sert à diviser le corps du document HTML
- ❑ **04** éléments sont utilisés pour définir les différentes sections:
 - `<header>`
 - `<nav>`
 - `<main>`
 - `<footer>`
- ❑ Ces éléments font partie de l'élément **body**

Section (2/2)

- ❑ `<header>...</header>`: section en **haut** de la page.
Elle peut contenir le logo, etc
- Ne pas confondre avec l'élément `<head>` dont le contenu ne sera pas affiché dans la page (sauf le titre)
- ❑ `<nav>...</nav>` : liens de navigation (**menu**)
- ❑ `<main>...</main>`: section **principale** de la page
- ❑ `<footer>...</footer>`: section en **bas** de page (**liens** additionnels, **copyright**, **contacts**)

Commentaires

- Un commentaire est introduit par les symboles `<!--` et `-->`
- Permet de rendre le document HTML facile à lire et à comprendre
- Les commentaires ne seront pas affichés par les navigateurs

Validation du HTML

- Le site pour vérifier le code html en ligne est <https://validator.w3.org/>
- 03 techniques:
 - Validation par URI
 - Validation par fichier contenant le code HTML
 - Validation par la saisie directe du code HTML