

BDW :

BASES DE DONNÉES & PROGRAMMATION WEB

**PARTIE
PROGRAMMATION WEB**



HTML

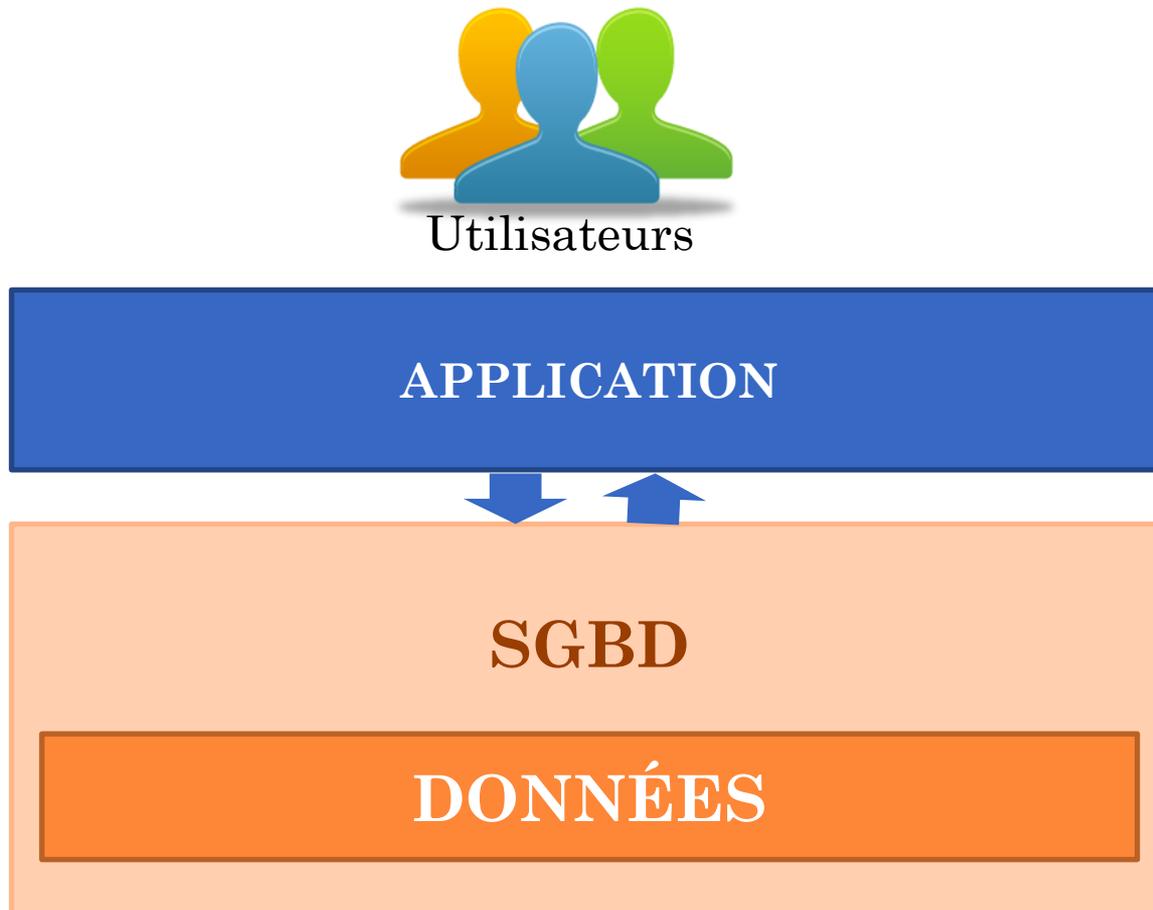
Nicolas Lumineau

nicolas.lumineau@univ-lyon1.fr

Remerciements à Emmanuel Coquery et Fabien Duchateau

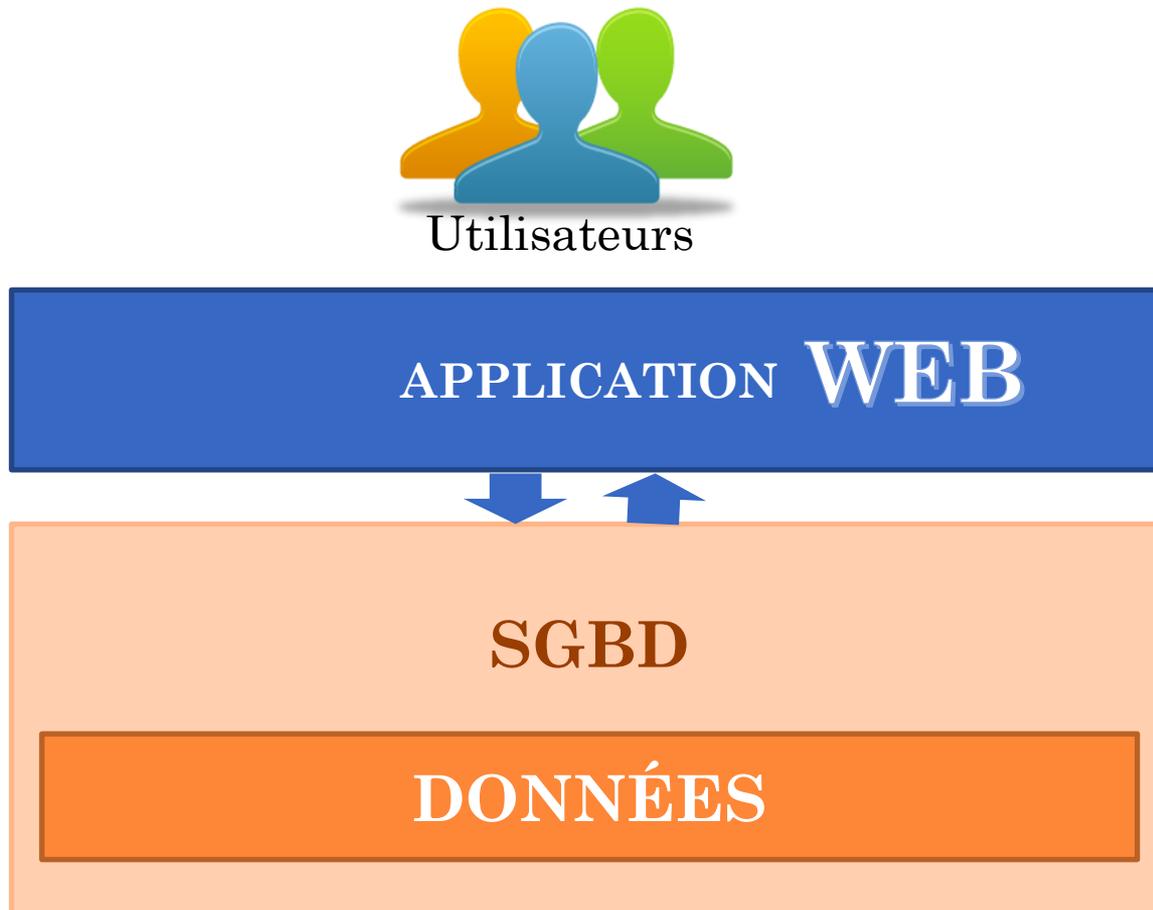
REMARQUE GÉNÉRALE

- Les SGBD sont rarement manipulés directement par les utilisateurs, mais ils le sont plutôt à travers une application.



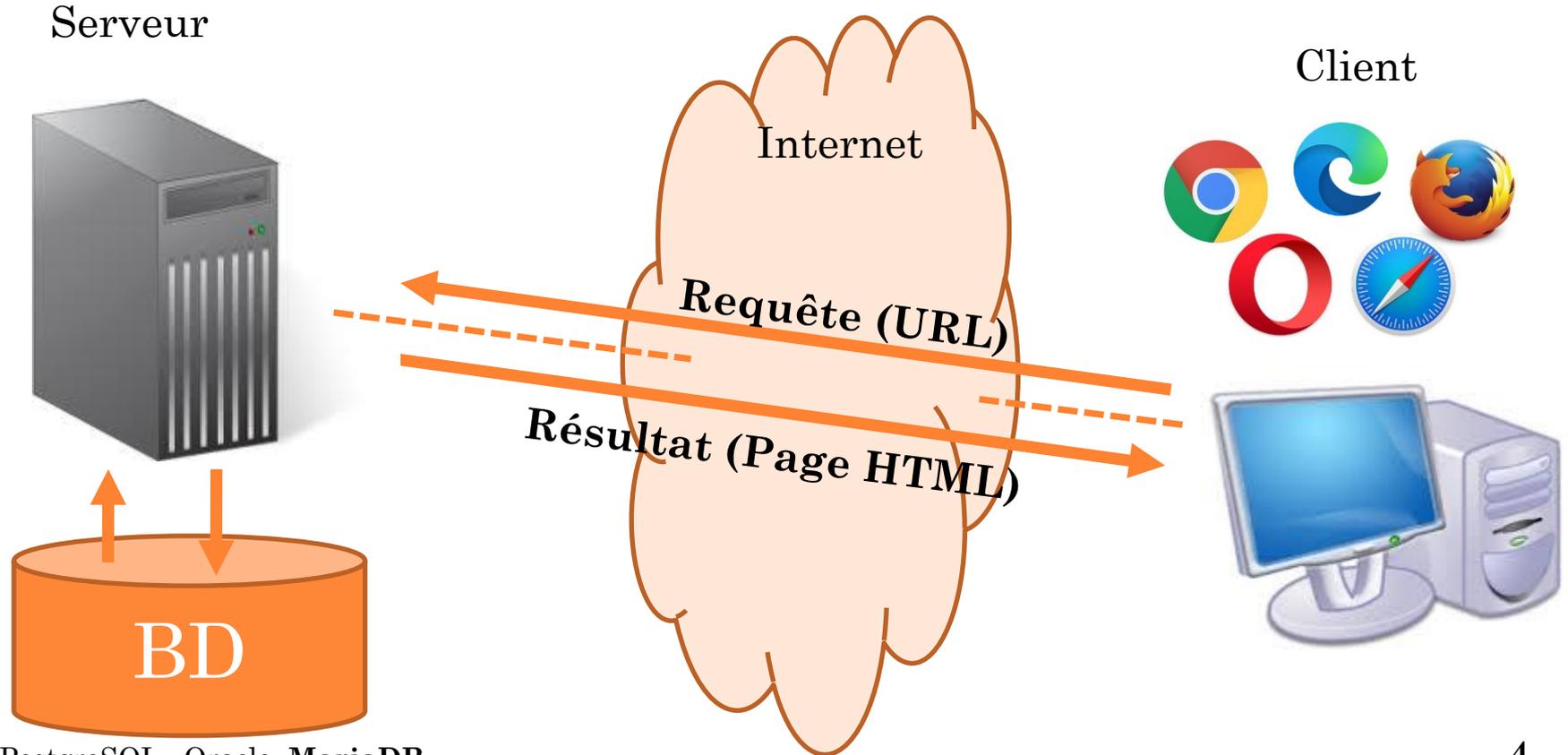
REMARQUE GÉNÉRALE

- Pour BDW, on se focalisera sur des applications Web
 - Approche très répandue



BD ET WEB

- De nombreuses bases de données sont accessibles via le Web



PostgreSQL, Oracle, MariaDB, MySQL, DB2, ...

PAGES WEB STATIQUES

○ Principe de fonctionnement :

- L'utilisateur demande l'accès à une page Web depuis son navigateur (Edge, FireFox, Chrome ...).
 - Adresse saisie, clic sur un lien, utilisation d'un signet...
- Le navigateur envoie une demande serveur Web ciblé.
- Le serveur Web lit sur son disque le fichier demandé.
- Le serveur Web envoie le contenu du fichier au navigateur.
- Le navigateur affiche le contenu de la page.

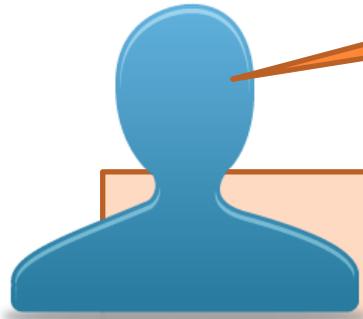
○ Remarque:

- Le fichier peut lui-même référencer d'autres fichiers nécessaires pour l'affichage (ex: images)

Dans un contexte statique, le serveur Web ne fait que transmettre le ou les fichiers demandés

ILLUSTRATION

Aller sur la page
<https://perso.liris.cnrs.fr/nicolas.lumineau>



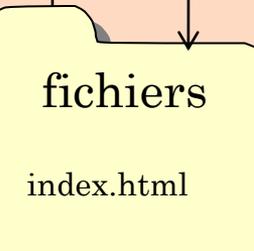
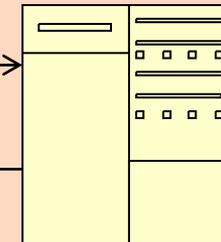
Client

navigateur



Serveur

Serveur Web



(1) : demande de la page **index.html**

(4) : envoi de **index.html**

(3) : lecture de **index.html**

(2) : accès à **index.html**

BDW1

(5) : affichage de **index.html**

REMARQUE

- Pour changer le contenu affiché d'une page statique, il est nécessaire d'éditer le fichier au niveau du serveur.
- Question
 - Que se passe-t-il si l'on souhaite changer le contenu d'une page au moment où elle est demandée ?
en fonction :
 - du contenu des données dans un SGBD
 - de paramètres fournis au moment de la requêtes
(cf : les formulaires)



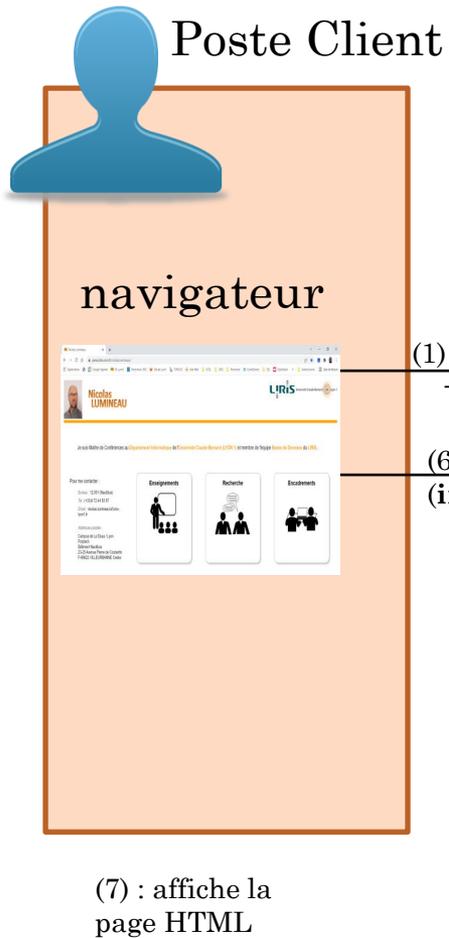
Utiliser des pages web dynamiques

PAGES WEB DYNAMIQUES

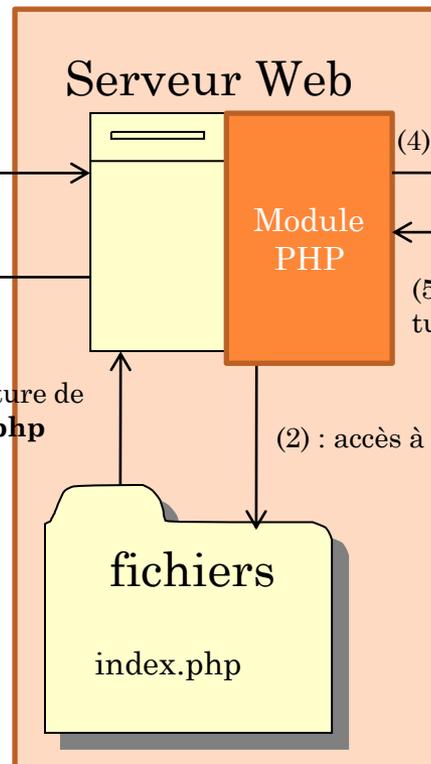
○ Principe

- L'utilisateur demande l'accès à une page Web depuis son 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 cherche sur le disque le code source d'un programme.
- Le serveur web exécute ce programme qui peut :
 - utiliser les paramètres transmis avec la demande du navigateur ;
 - aller chercher des données dans une base de données ou sur le disque.
- Durant l'exécution, le programme génère un contenu HTML.
- Le contenu HTML est envoyé au navigateur.
- Le navigateur affiche le résultat.

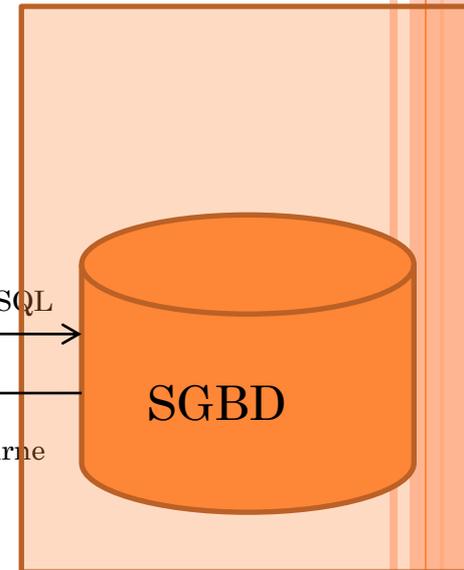
EXEMPLE



Poste Serveur Web



Poste Serveur de données



(1) : demande de la page **index.php** + paramètres

(6) : retourne page HTML (**index.php**)

(3) : lecture de **index.php**

(2) : accès à **index.php**

(4) : envoi SQL

(5) : retourne tuples

REMARQUE

- La majorité des sites Web sont dynamiques depuis les années 2000 (Web 2.0)
- Le temps nécessaire pour afficher une page dynamique intègre le temps d'exécution du programme.
 - ➔ en pratique, ce n'est souvent pas un problème!
- Mettre en place des pages dynamiques nécessite un langage de programmation côté serveur.

EXEMPLES DE LANGAGES POUR LA PROGRAMMATION WEB

- Pour la structure et le contenu des pages :
→ **HTML, XML**
- Pour la mise en forme des pages :
→ **CSS**
- Pour les traitements de génération de contenu (côté serveur)
→ **PHP, Python, Java, Perl**
- Pour la gestion des interactions (côté client)
→ **Javascript**
- Pour interroger les données
→ **SQL, Xpath, Xquery, SPARQL**

PROGRAMMATION WEB DANS BDW1

- Pour la structure et le contenu des pages :
 ➔ HTML
- Pour la mise en forme des pages :
 ➔ CSS
- Pour les traitements de génération de contenu (côté serveur)
 ➔ PHP
- Pour interroger les données
 ➔ SQL

QUELS OUTILS POUR LA PROGRAMMATION WEB ?

- Editeurs textuels :
 - Emacs, Atom, SublimeText...
- Editeurs WYSIWYG
 - Dreamweaver, NVU...
- CMS (Content Management System)
 - Drupal, Joomla, Spip...
- Framework
 - Symfony, Codeigniter, Hibernate ...

DANS BDW

- L'objectif étant de comprendre ce que vous faites et d'être opérationnel rapidement:
 - Editeur de texte
 - Pas de *framework*

HTML

Introduction

HTML : Généralités et Syntaxe de base

Balises structurantes

Balises de contenu

Identifiant & Classe

Formulaires

QUELQUES GÉNÉRALITÉS SUR HTML

- HTML : HyperText Markup Language
 - Fichier avec extension : **.html**
 - Langage de balisage : **<balise>**, **</balise>**, **<balise/>**
 - Dates clés:
 - 1989-1990 : création au CERN
 - 1996 : Standard W3C
 - Langage qui a évolué : HTML2 (1996), HTML3/4 (1997), HTML5(2014)
- A quoi servait HTML ?
 - A définir la structure et la mise en forme d'un document
 - Le document est interprété par le navigateur qui gère l'affichage
- A quoi ne doit plus servir HTML?
 - A définir la mise en forme du document → **CSS**



HTML : PRINCIPE DES BALISES

- L'organisation des balises définit la structure du document:
 - Le **nom de la balise** indique le **type d'élément** que l'on ajoute dans le document
 - Ex: Un titre, une section, un menu, un paragraphe, une image
 - Les balises peuvent **être imbriquées**
 - Ex: Une section contient des paragraphes et une image
- Remarque :
 - Certaines balises n'apportent pas de contenu mais sont utilisées qu'à des fins de structuration du document
 - Ex : `<div>`

HTML : TYPES DE BALISE

- **Les balises en paires** qui délimitent un contenu avec une balise ouvrante puis une balise fermante:

`<nom_balise> contenu </nom_balise>`

 La balise fermante a un slash (/) **avant** le nom dans la balise

- **Les balises orpheline** qui introduit un élément dans le document :

`<nom_balise_orpheline />`

 La balise orpheline a un slash (/) **après** le nom dans la balise

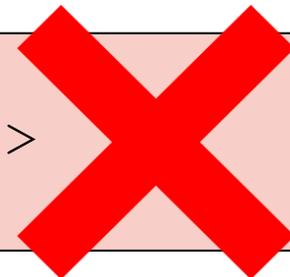
ILLUSTRATION

```
<html>
  <head> </head>
  <body>
    <h1> <br /> </h1>
  </body>
</html>
```



Il est important d'être rigoureux dans l'imbrication des balises sans les entremêler

```
<html> <head> <h1> </head>
  <body> <br /> </h1> </html>
</body>
```



HTML : ATTRIBUT DE BALISE

- Une balise ouvrante ou orpheline peut contenir des **attributs** qui permettent de préciser des propriétés de la balise:

```
<nom_balise att="val" > contenu </nom_balise>
```

```
<nom_balise_orpheline att1="val1" att2="val2" />
```

ILLUSTRATION

```
<html lang = "fr" >
  <head> </head>
  <body>
    <h1> <br /> </h1>
    
  </body>
</html>
```

HTML : STRUCTURE D'UNE PAGE

- Une page HTML a la structure suivante :

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <!-- Bloc entête contenant Métadonnées, scripts... -->
  </head>
  <body>
    <!-- Bloc affiché dans le navigateur -->
  </body>
</html>
```

- Les commentaires (non affichés) sont délimités par `<!--` et `-->`

HTML: DÉFINITION DU TYPE DE DOCUMENT

- La balise `<!DOCTYPE html>` spécifie le type de document:
 - Existence de plusieurs versions et variantes de la norme HTML
 - ➔ Nécessaire pour préciser la version utilisée et pour que le navigateur interprète correctement le code HTML fourni
 - ➔ Obligatoire pour valider son document (W3C Markup Validation Service)
- La balise `DOCTYPE` est simplifiée en HTML5 ...

```
<!DOCTYPE html>
```

... ce qui n'était pas le cas en HTML4:

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

HTML : DÉBUT DU DOCUMENT

- La balise `<html>` spécifie le début du document
- Remarque:
Avec `<html lang="fr">`, on spécifie que le document sera en langue française grâce à l'attribut 'lang'.
- La balise `</html>` spécifie donc la fin du document

HTML : EN-TÊTE DU DOCUMENT

- La balise `<head>` délimite la partie du document qui spécifie :
 - Des **métadonnées** sur le document qui sont utilisables par le navigateur ou les moteurs de recherche (pour indexation)
 - Des **scripts** (ex: javascript) contenant du code utilisable pour l'affichage du corps du document
 - Des **liens vers des scripts externes** contenant du code utilisable pour l'affichage du corps du document
 - Des **liens vers des feuilles de style** gérant la mise en forme du document
 - Un **lien vers une image servant d'icône** dans l'onglet du navigateur
- ➔ L'entête est donc une partie qui ne s'affiche pas dans le contenu de la page.

HTML : EN-TÊTE DU DOCUMENT

- A propos des Métadonnées

- Syntaxe générale :

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

- Il existe des métadonnées spécifiques:

- Balise `<title>` permet de spécifier le titre de page apparaissant dans l'onglet du navigateur (**devenu obligatoire**)
- Balise `<meta charset="utf-8" />` permet de spécifier l'encodage du document (gestion des caractères spéciaux)
- Balise `<link rel="stylesheet" href="./style.css" />` permet de spécifier la feuille de style associée au document

ILLUSTRATION DE L'ENTÊTE HTML

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <!-- Bloc entête contenant Métadonnées, scripts... -->
    <meta charset=utf-8 />
    <title>Page BDW1</title>
    <link rel="stylesheet" href="./css/style.css" />
    <link rel="icon" type="image/png" href= "./img/logo.png" />
    <meta name= "author"  content= "Nicolas Lumineau" />
    <meta name= "keywords"  content= "Cours HTML, HTML5, BD" />
  </head>
  <body>
    <!-- Bloc affiché dans le navigateur -->
  </body>
</html>
```

HTML : CONTENU DU DOCUMENT

- La balise `<body>` délimite le corps/contenu du document qui est structuré par les balises.

➔ Le corps du document est donc la partie qui s'affiche dans le navigateur

- Le *corps* peut contenir :
 - Titre, sous-titre
 - Paragraphe
 - Mots plus ou moins importants
 - Listes ordonnées ou non
 - Liens hypertextes
 - Images
 - Regroupement
 - *Formulaires*
 - *Onglets de saisie, boutons*

HTML : BONNES PRATIQUES

- Avoir de la rigueur dans l'écriture des balises
 - Les navigateurs sont assez tolérants aux erreurs de programmation ... jusqu'à un certain point!
- Ecrire les balises en minuscule
- Commenter le code `<!-- ... -->`
- Indenter le code
- Eviter d'écrire des blocs de code trop compacts
- *Eviter les balises spécifiques à un navigateur*
- Valider son fichier HTML avec un validateur

HTML

Introduction

HTML : Généralités et Syntaxe de base

Balises structurantes

Balises de contenu

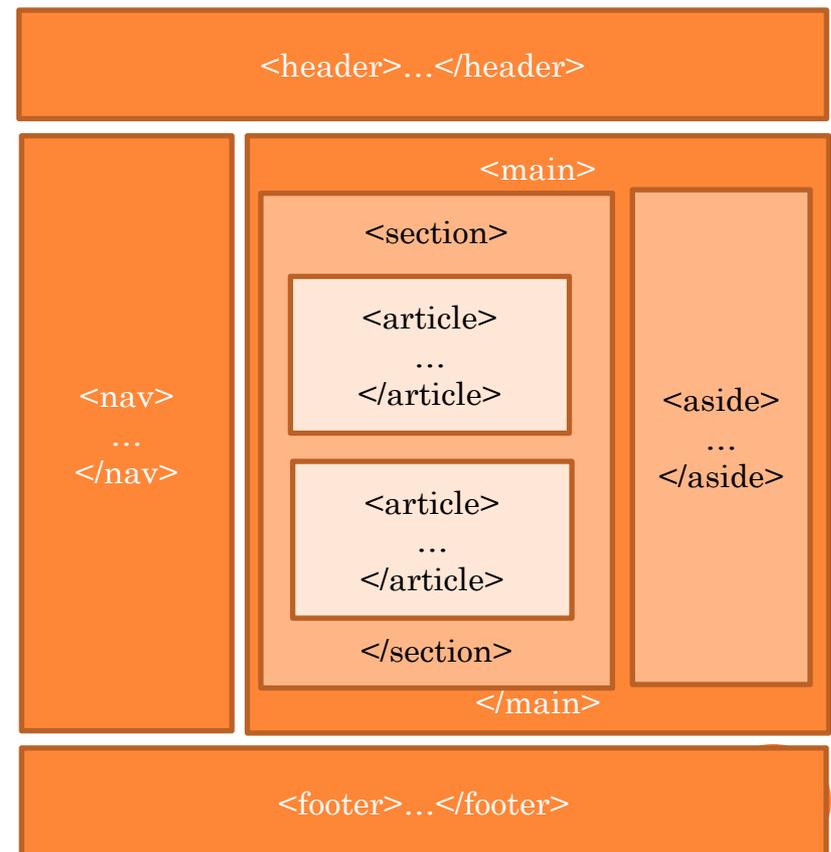
Identifiant & Classe

Formulaires

LES BALISES STRUCTURANTES

- Depuis HTML5, certaines balises apportent de la sémantique à la structuration de la page

- *header*
- *footer*
- *nav*
- *main*
- *section*
- *article*
- *aside*



BALISES STRUCTURANTES: ENTÊTE ET PIED DE PAGE

Balise: `<header>...</header>`

- L'entête d'une page est située en haut de la page et il inclut généralement:
 - le logo ou une bannière,
 - le nom du site,
 - un slogan

Balise: `<footer>...</footer>`

- Le pied de page est situé en bas de la page et il inclut généralement:
 - les crédits,
 - les mentions légales,
 - un lien de contact

BALISES STRUCTURANTES: MENU DE NAVIGATION DE PAGE

Balise: <nav>...</nav>

- Le menu de navigation regroupe les principaux liens de navigation à travers le site
- Les liens de navigation seront des liens hypertextes

BALISES STRUCTURANTES: SECTION / ARTICLE / APARTÉ DE PAGE

Balise: <section>...</section>

- La section regroupe des contenus (paragrapes...) selon leur thématique.

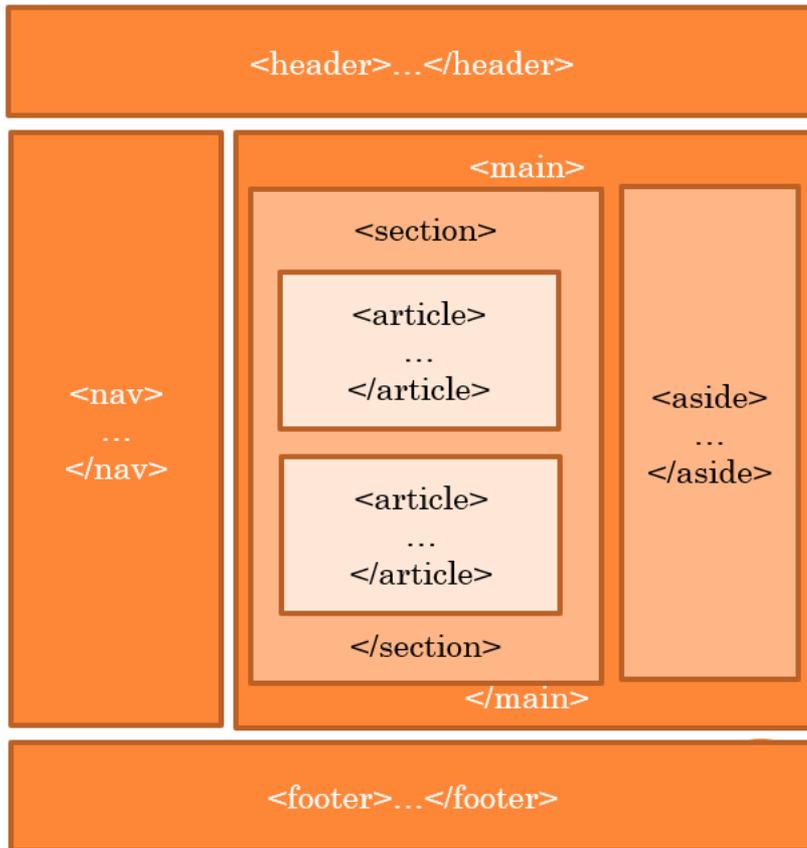
Balise: <article>...</article>

- L'article est une zone de contenu indépendant qui en théorie peut être repris sur un autre site.

Balise: <aside>...</aside>

- L'aparté est une zone de contenu complémentaire ou indirect avec le contenu principal.

ILLUSTRATION



```
1 <!DOCTYPE html>
2 <!-- Script CM cours BDW1-->
3 <html>
4 <head>
5   <meta charset="utf-8" />
6   <title>BDW1 - Démo</title>
7 </head>
8 <body>
9
10 <header>
11   <!-- Espace de l'entête de la page -->
12 </header>
13
14 <nav>
15   <!-- Espace de navigation / Menu -->
16 </nav>
17
18 <main>
19   <!-- Espace principal -->
20
21   <section> <!-- Section -->
22
23     <article> <!-- Article 1 -->
24     </article>
25
26     <article> <!-- Article -->
27     </article>
28
29     <article> <!-- Article -->
30     </article>
31
32   </section>
33
34   <aside>
35     <!-- Espace a parté de page -->
36   </aside>
37
38 </main>
39
40 <footer>
41   <!-- Espace Pied de Page -->
42 </footer>
43
44 </body>
45 </html>
```

BALISES DE REGROUPEMENT:

TYPE DE RENDU

- Chaque élément d'une page possède un type de rendu qui permet de définir la manière dont il est placé dans la page par rapport aux autres éléments.
- Ainsi des éléments peuvent être placés:
 - les uns au-dessous des autres.
 - les uns à côté des autres
- Différents types de rendu:
 - BLOCK : l'élément est considéré comme un bloc unique sur toute la largeur de la page. Les autres éléments seront placés au-dessus ou au-dessous mais pas à côté de celui-ci.
 - Les éléments de type de rendu BLOCK sont placés les uns en-dessous des autres
 - INLINE : l'élément n'impose pas d'être unique sur la largeur de la page
 - Les éléments de type de rendu INLINE sont placés les uns à côté des autres

BALISES DE REGROUPEMENT

Balise: `<div>...</div>`

- `<div>` est par défaut de type de rendu : BLOCK

```
<div><p>un paragraphe</p><p>et un autre</p></div>
```

Balise: `...`

- `` est par défaut de type de rendu : INLINE

```
<p>une ligne <span>et</span> une autre</p>
```

○ Remarque:

- Balises neutres sans effet sur la page ... sauf en cas de mise en forme associée (cf cours CSS)

BONNES PRATIQUES

- Réfléchir à la structuration du site dès la conception
- Privilégier l'utilisation des balises structurantes
 - Apport de structure et de sémantique
- Compléter la structuration du document avec des balises de regroupement : `<div>` et ``

HTML

Introduction

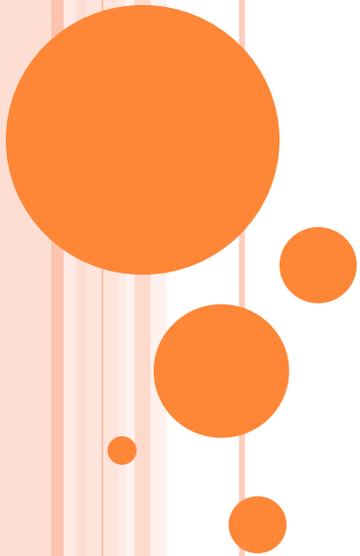
HTML : Généralités et Syntaxe de base

Balises structurantes

Balises de contenu

Identifiant & Classe

Formulaires



LES BALISES DE CONTENU

- En HTML, il est possible de définir des contenus spécifiques correspondant à :
 - Un titre, sous-titre
 - Un paragraphe
 - Des listes d'éléments (numérotés ou non)
 - Des Tableaux
 - Des images
 - Des liens hypertextes
 - ...

PARAGRAPHE ET RETOUR À LA LIGNE

Balise paragraphe : `<p>...</p>`

```
<p>
```

```
Un texte qui s'étend sur plusieurs  
lignes et qui forme un paragraphe
```

```
</p>
```

- un paragraphe est précédé et suivi d'un saut de ligne!

Balise retour à la ligne : `
` (*balise orpheline*)

```
<p>
```

```
Un texte qui s'étend <br/>
```

```
sur plusieurs <br/>
```

```
lignes et qui forme un paragraphe
```

```
</p>
```

MOTS PLUS OU MOINS IMPORTANTS

Balise pour mots très importants : `<mark>...</mark>`

`<p> Un contenu avec <mark>des mots très importants </mark> à l'intérieur.</p>`

Balise pour mots importants : `...`

`<p> Un contenu avec des mots importants à l'intérieur.</p>`

Balise pour mots relativement importants : `...`

`<p> Un contenu avec des mots faiblement important.</p>`



Ces balises définissent l'importance des mots d'un point de vue **sémantique**.

Par défaut, le navigateur appliquera un style particulier (gras, italique) qui est possible de personnaliser via le CSS.

TITRES

Balise : `<h1>...</h1>`, `<h2>...</h2>`, ..., `<h6>...</h6>`

- `<h1>un titre</h1>`
 - Spécifie un titre principal (avec généralement un affichage très gros, en gras par défaut).
- `<h2>second titre</h2>`
 - Spécifie un titre moins important (avec affichage un peu moins gros que pour `<h1>` par défaut).
- ...
- `<h6>sixième titre</h6>`
 - Titre peu important.



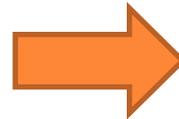
Ces balises définissent une importance des titres d'un point de vue **sémantique**.

LES LISTES

○ Les listes non ordonnées

Balise : ` contenu ligne `

```
<ul>  
  <li>Ligne1</li>  
  <li>Ligne2</li>  
  <li>Ligne3</li>  
</ul>
```

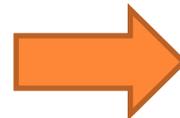


- Ligne1
- Ligne2
- Ligne3

○ Les listes ordonnées

Balise : ` contenu ligne `

```
<ol>  
  <li>Ligne1</li>  
  <li>Ligne2</li>  
  <li>Ligne3</li>  
</ol>
```



1. Ligne1
2. Ligne2
3. Ligne3

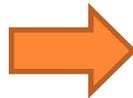
LES TABLEAUX

Balise : <table>...</table>

- Une table dispose :
 - Un partie « entête » : **Balise : <thead>...</thead>**
 - Un partie « contenu » : **Balise : <tbody>...</tbody>**
- Pour créer une ligne (dans l'entête ou le contenu)
Balise : <tr> ... </tr>
- Pour créer une cellule dans l'entête
Balise : <th> ... </th>
- Pour créer une cellule dans le contenu
Balise : <td> ... </td>

LES TABLEAUX

```
<table>
  <thead>
    <tr>
      <th>Col1</th>
      <th>Col2</th>
      <th>Col3</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Cell11</td>
      <td>Cell12</td>
      <td>Cell13</td>
    </tr>
    <tr>
      <td>Cell21</td>
      <td>Cell22</td>
      <td>Cell23</td>
    </tr>
  </tbody>
</table>
```



Col1	Col2	Col3
Cell11	Cell12	Cell13
Cell21	Cell22	Cell23

LES TABLEAUX

- Dans un tableau, il est possible de fusionner des cellules:

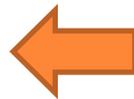
- Horizontalement :

Attribut : colspan

- Verticalement :

Attribut : rowspan

Col1	Col23	
	Col2	Col3
Cell11	Cell2	Cell13
Cell21		Cell23



```
<table>
  <thead>
    <tr>
      <th rowspan="2">Col1</th>
      <th colspan="2">Col23</th>
    </tr>
    <tr>
      <th>Col2</th>
      <th>Col3</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Cell11</td>
      <td rowspan="2">Cell2</td>
      <td>Cell13</td>
    </tr>
    <tr>
      <td>Cell21</td>
      <td>Cell23</td>
    </tr>
  </tbody>
</table>
```

LES LIENS HYPERTEXTES

```
<a href="adresse_web">zone_cliquable</a>
```

- texte devient un lien cliquable.
- lorsque l'on clique sur texte, le navigateur va à l'adresse (URL) *adresse_web*.

○ Exemples :

```
<a href="http://www.w3.org/TR/html401/">Documentation HTML</a>
```

- Le texte « Documentation HTML » est un lien vers la page web dont l'adresse est `http://www.w3.org/TR/html401/`

```
<a href="/truc.html">Des trucs</a>
```

- Le texte « Des trucs » est un lien vers la page web */truc.html* située au même endroit que la page courante.

LES LIENS HYPERTEXTES

`zone_cliquable`

- texte devient un lien cliquable.
- lorsque l'on clique sur texte, le navigateur va à l'adresse (URL) *adresse_web*.

○ Exemples :

`Documentation HTML`

- Le texte « Documentation HTML » est un lien vers la page web dont l'adresse est `http ://www.w3.org/TR/html401/`

Adresse absolue

Adresse relative

`Des trucs`

- Le texte « Des trucs » est un lien vers la page web *.truc.html* située au même endroit que la page courante.

LES LIENS HYPERTEXTES

```
<a href="adresse_web"
  name="ancrage"
  target="cible">zone_cliquable</a>
```

○ Autres attributs de <a>

• target :

- Permet de spécifier où la page ciblée doit s'ouvrir.
 - **target="_self"** (Par défaut) : ouvre le document dans l'onglet courant
 - **target="_blank"** : ouvre le document dans un autre onglet

• name:

- Permet de définir une zone d'ancrage dans un document
 - En plaçant **** dans le fichier index.html, je peux accéder directement à cette position grâce à l'adresse **http://mon_serveur.fr/index.html#Conclusion**

LES IMAGES

- Inclure une image :

```

```

- `` est une balise orpheline
- *adresse_web_image* : adresse web où chercher l'image.
- *texte_alternatif* : attribut obligatoire pour des raisons d'accessibilité

LES IMAGES

- Inclure une image avec une légende :

```
<figure>
  
  <figcaption>
    Figure 1 : un exemple
  </figcaption>
</figure>
```



- La balise `<figcaption>` introduit une légende à une figure
- La balise `<figure>` regroupe l'image et sa légende

ILLUSTRATION

```
1  <!DOCTYPE html>
2  <!-- Script CM cours BDW1-->
3  <html>
4  <head>
5      <meta charset="utf-8" />
6      <title>Titre de la page</title>
7  </head>
8  <body>
9      <h1>Titre principal</h1>
10     <p>Ceci est un premier paragraphe!</b>
11     <!-- ceci est un commentaire, il ne sera pas affiché par le navigateur -->
12
13     <h2>Sous-titre ou Section</h2>
14     <p>bla bla bla...</p>
15     <a name="sectionBlabla"></a><!-- ceci est une ancre appelée sectionBlabla -->
16     <h3>Sous-sou-titre ou Sous-section</h3>
17     <p>
18     toto titi tata</br>
19     pim pam poum</br>
20     </p>
21
22     <h2>Un exemple de liste sans numérotation</h2>
23     <ul>
24         <li>item 1</li>
25         <li>item 2</li>
26         <li>item 3</li>
27         <li>...</li>
28     </ul>
29
30     <h2>Un exemple de liste avec numérotation</h2>
31     <ol>
32         <li>item 1</li>
33         <li>item 2</li>
34         <li>item 3</li>
35         <li>...</li>
```

BILAN

- Il existe de nombreuses balises pour définir du contenu
 - En HTML 5 : 101 balises

HTML

Introduction

HTML : Généralités et Syntaxe de base

Balises structurantes

Balises de contenu

Identifiant & Classe

Formulaires

IDENTIFIANT

- Il est possible d'associer un identifiant unique à une balise
 - Attribut : **id**
- Intérêt :
 - Pour référencer certaines balises → cf Formulaire
 - Pour la manipulation des éléments via un langage de programmation
- Exemple

```
<p id="resume"> ... </p>
<p id="conclusion"> ... </p>
<h2 id="soustitre"> ... </h2>
<img id="logo" href="..." />
```

CLASSE

- Il est possible d'associer une classe à plusieurs balises pour les lier.
 - Attribut : **class**
- Intérêt :
 - Pour l'application de mise en forme → cf CSS
 - Pour la manipulation des éléments via un langage de programmation (cf JS)

○ Exemple

```
<p class="para1"> ... </p>
```

```
<p class="para1"> ... </p>
```

```
<img id="logo" href="..." />
```

HTML

Introduction

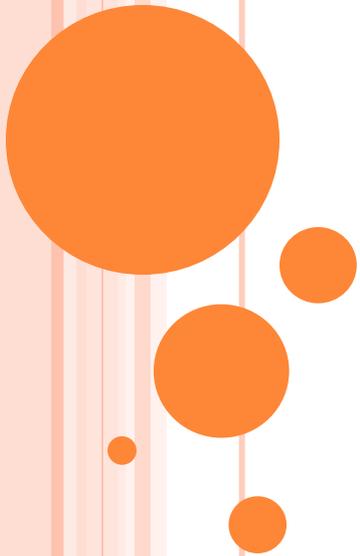
HTML : Généralités et Syntaxe de base

Balises structurantes

Balises de contenu

Identifiant & Classe

Formulaires



LES FORMULAIRES

- Objectifs des formulaires web :
 - permettre à l'utilisateur de saisir ces **paramètres**
 - Envoyer les valeurs saisies pour chaque paramètre au serveur
 - Accéder à une page dynamique générée par le serveur selon les valeurs des paramètres

- Comment définir les paramètres?
 - en utilisant des composants de saisie:
 - champs textuels, des listes déroulantes, des cases à cocher ...
 - à chacun des composants correspond un paramètre.

EXEMPLES DE FORMULAIRES



Faites une recherche et financez un projet 🔍

4 369 628 euros déjà collectés. Vos petites gouttes d'eau vont faire les grandes rivières.

[Ajouter mes sites favoris](#)

sncfconnect [Voyager](#) [Billets](#) [Offres](#) [Compte](#)

Rechercher

Départ : Lyon ↕

Arrivée : Marseille

Aller : Aujourd'hui, 16h + Ajouter un retour

Voyageur 1 9 voyageurs max. [En savoir plus](#)

👤 Voyageur 1 : 30 - 59 ans
Sans carte de réduction
Sans carte de fidélité ➤

Service Central d'Authentification - CAS

Entrez votre identifiant et votre mot de passe.

Identifiant: ?

Ex: p1234567 ou prenom.nom

Mot de passe: ?

Votre mot de passe

- Prévenez-moi avant d'accéder à d'autres services.?
- Je suis sur un ordinateur public.
- Rester connecté ?

SE CONNECTER

[Français](#) | [English](#) | [Español](#)

[Aide](#)

BDW1

FORMULAIRE: BALISE PRINCIPALE

```
<form action="adresse_web"  
      method="POST"  
      name="nomF">  
  contenu_formulaire  
</form>
```

- **action** : spécifie l'adresse de la page web dynamique à laquelle on souhaite accéder via ce formulaire ;
- **method** : spécifie le mode de transmission des paramètres:
 - POST : les paramètres sont transmis dans le corps de la requête HTTP
 - GET : les paramètres sont transmis dans l'URL de la page
- **name** : le nom du formulaire (optionnel).
- **contenu_formulaire** : du texte et des balises représentant les différents composants de saisie ;

MÉTHODE DE TRANSMISSION DES PARAMÈTRES

○ Deux méthodes de transmission des paramètres :

● GET

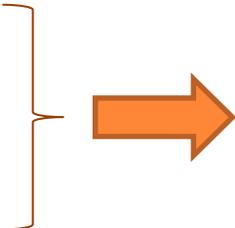
- Les paramètres sont encodés avec l'adresse de la page
 - à la fin de l'adresse, on ajoute le caractère ?
 - puis pour chaque paramètre on ajoute nom=val
 - les paramètres sont séparés par le caractère &
- Utile pour spécifier des paramètres dans un lien hypertexte.

`http://serveur.fr/index.php?ue=bdw1&annee=2021&semestre=2`

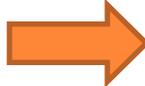
● POST

- Les paramètres sont encodés séparément de l'adresse web.
- Plus pratique pour les formulaires.

COMPOSANTS DE FORMULAIRES

- Un formulaire peut contenir
 - Des onglets de saisie de texte
 - Des cases à cocher
 - Des groupes d'option
 - Des boutons

Balise de type `<input>`

- Des zones de texte long 

Balise de type `<textarea>`

- Des listes déroulantes 

Balise de type `<select>`
- Les composants sont identifiés et nommés
 - Attribut *id* : identification du composant dans la page
 - Attribut *name* : permet la récupération de la valeur du paramètre par le nom du composant
 -
- Les composants sont introduits par des textes descriptifs (*labels*)

LABELS

- Un label permet d'associer un texte descriptif à un composant de formulaire

```
<label for="id_Composant">Texte_descriptif</label>
```

- Affiche un label " Texte_descriptif" qui décrit le composant identifié par "id_Composant"
- Remarque:
 - Un clic sur le label donne le focus au composant associé

INPUT DE TYPE TEXT

```
<label for="id1">Un champ texte</label>
```

```
<input id="id1" type="text" name="nom_input"/>
```

- Crée un champ de saisie pour une ligne de texte (accompagné d'un label).
- **name** : le nom du paramètre correspondant à ce composant.
- On peut ajouter les attributs suivants :
 - `size="un nombre"` : la taille du champ en caractères ;
 - `value="une valeur"` : texte pré-saisi
 - utile pour modifier des informations.
- *input* est une balise orpheline

Nom

Prénom

INPUT DE TYPE PASSWORD

```
<input id="id2" type="password" name="nom_input" />
```

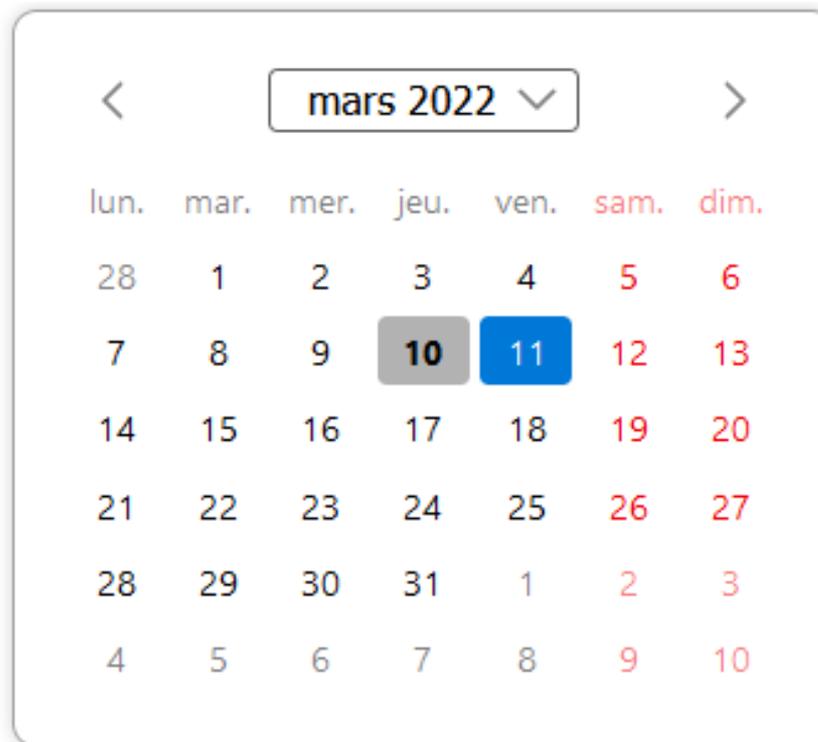
- Crée un champ de saisie pour une ligne de texte cachée.

Mot de passe

INPUT DE TYPE DATE

```
<input id="id3" type="date" name="nom_input" />
```

- Crée un champ avec la possibilité de choisir une date dans un calendrier



En date du :

INPUT DE TYPE NUMBER

```
<input id="id4" type="number" name="nom_input"
      min=0 max=10/>
```

- Crée un champ avec la possibilité de choisir un entier compris entre la valeur min et la valeur max.

Entier (0-10) :

INPUT DE TYPE SUBMIT OU RESET

```
<input id="id5" type="submit" value="texte">
```

- Crée un bouton déclenchant le chargement de la page de destination (attribut action de la balise <form>).
 - *texte* est un texte qui sera affiché sur le bouton.

```
<input id="id6" type="reset" value="texte">
```

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

Nom

Prénom

INPUT DE TYPE CHECKBOX

```
<label>Choix1<input type="checkbox" name="nomC"
id="id1" value="val1"></label>
```

```
<label>Choix2<input type="checkbox" name="nomC"
id="id2" value="val2"></label>
```

- Crée deux cases à cocher identifiées respectivement par *id1* et *id2* ayant le même nom *nomC*
 - Si la première case est cochée, alors le paramètre retourne la valeur *val1*, sinon si la seconde case est cochée, alors le paramètre retourne la valeur *val2*
 - Si les deux cases sont cochées, alors le paramètre retourne la liste [*val1*, *val2*]
 - Attribut optionnel *checked* permet de spécifier la case sélectionnée par défaut

Loisirs préférés :

Sports collectifs Sports individuels Musique

Lecture Ecriture Cinéma

INPUT DE TYPE RADIO

```
<label>Choix1<input type="radio" name="nomC"
id="id1" value="val1"></label>
```

```
<label>Choix2<input type="radio" name="nomC"
id="id2" value="val2"></label>
```

- Crée deux boutons radios identifiés respectivement par *id1* et *id2* ayant le même nom *nomC*
 - **Un seul bouton radio peut être sélectionné à un instant donné**
 - Si la première case est cochée, alors le paramètre retourne la valeur *val1*, sinon si la seconde case est cochée, alors le paramètre retourne la valeur *val2*
 - Attribut optionnel *checked* permet de spécifier la case sélectionnée par défaut

Sexe : Homme Femme Autre

INPUT DE TYPE HIDDEN

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

- C'est un paramètre caché qui ne peut être saisi par l'utilisateur
- Permet de donner la valeur *val* au paramètre *nom*.
- Ce composant n'est pas affiché (mais accessible dans le code de la page).
- Utile pour spécifier un identifiant dans un formulaire de modification des informations de la base.

FORMULAIRE : SAISIE D'UN GRAND TEXTE

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

- Crée un champ de saisie pour du texte sur plusieurs lignes.
 - *nom* est le nom du paramètre correspondant à ce composant.
 - *h* est la hauteur du composant en nombre de lignes.
 - *l* est la largeur du composant en nombre de caractères.
- *contenu* peut être vide et ne contient pas de balise.

LISTE DÉROULANTE

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

- Crée une liste déroulante ayant comme sélection possible : Texte 1, Texte 2, . . .
- La valeur du paramètre *nom* est donnée par la sélection choisie par l'utilisateur :
 - val1 pour Texte 1
 - val2 pour Texte 2
 - . . .
- L'attribut *value* est optionnel.
 - Par défaut c'est le texte dans la balise `<option></option>`
- On peut ajouter `selected="true"` dans une des balises
 - options pour pré-sélectionner cette option.

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

QUELQUES ÉLÉMENTS SUPPLÉMENTAIRES DE FORMULAIRES HTML5

- Spécificité HTML5:
 - tel, url, email
 - search
 - datalist (auto-complète un champ grâce à une liste de valeurs prédéfinies)
 - time, datetime, week, month
 - range (barre de progression)
 - color

QUELQUES ATTRIBUTS SPÉCIFIQUES À HTML5

- **placeholder** : texte indicatif dans un champ textuel, qui s'efface quand l'utilisateur commence à saisir un texte
- **required** : le formulaire ne peut être soumis si un des composant avec l'attribut *required* n'est pas rempli.
- **pattern** : vérifie que le champ respecte une expression régulière donnée (ex: une url)
- ...

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

ET LA MISE EN FORME DANS TOUT ÇA ?

CSS