

Développement Web 1

Antoine JAMIN

H111

antoine.jamin@univ-angers.fr

leria-info.univ-angers.fr/~a.jamin



9 séances CTD de 1h20

14 séances TP de 1h20

CT machine de 1h30

Vendredi 10 février à 14h

(1h30 pour 2nde chance)

Moodle : n°10739, clé r9vn53

https://leria-info.univ-angers.fr/~jeanmichel.richer/ensl1i_cm_xhtml.php

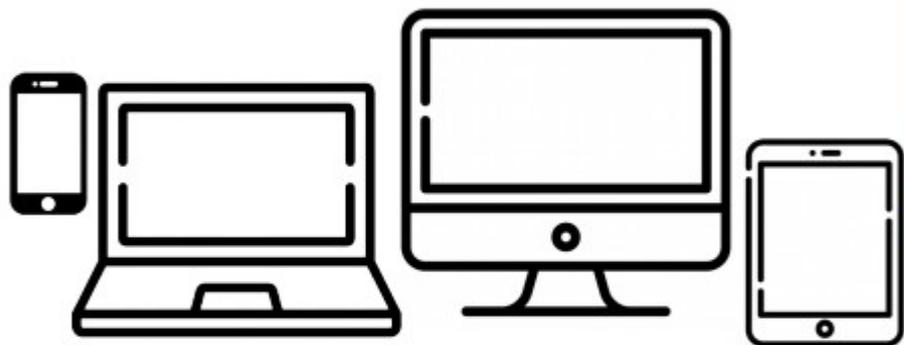
Séance 1 – HTML

Base Web

Structure HTML

Balises liées au texte

Front-end *côté client*



HTML



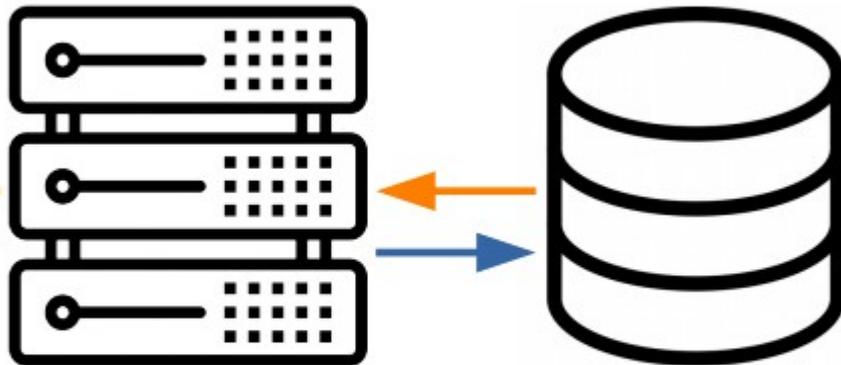
CSS



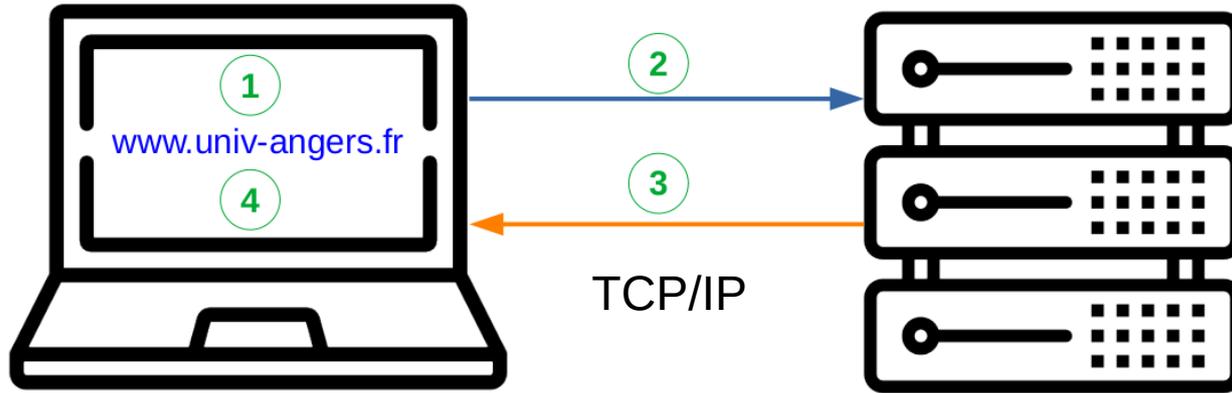
JS



Back-end *côté serveur*



Fonctionnement du web



1. Navigateur demande au DNS l'adresse du serveur contenant l'application web
2. Requête HTTP au serveur pour récupérer le contenu de l'application web
3. Réponse du serveur au navigateur. Si OK envoi du contenu de l'application web sous forme de paquets. Si NOK code erreur.
4. Assemblage des paquets reçus pour affichage de l'application.

Pour aller plus loin :

https://developer.mozilla.org/fr/docs/Learn/Getting_started_with_the_web/How_the_Web_works

```
<!DOCTYPE html>
<html dir="ltr" lang="fr-FR">
<!-- entete -->
<head>
  <meta charset="utf-8" />
  <title>XHTML 5</title>
</head>
<!-- contenu de la page -->
<body>
  <h1>Ma première page</h1>

  <p>Bonjour !</p>

</body>
</html>
```

Préambule

Entête (head)

Corps (body)

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4     <meta charset="UTF-8">
5     <title>Document</title>
6     <style>
7         /* CSS code here */
8     </style>
9     <link href="style.css" rel="stylesheet">
10 </head>
11 <body>
12     <div>
13         
14         <p>Un <em>exemple</em> de <b>code</b> :</p>
15     </div>
16 </body>
17 </html>
```


Balise Ouvrante / Balise Fermante

<nom> </nom>

- Commentaire : <!-- -->
- Paragraphe : <p> </p>
- Titres : <h1></h1> à <h6></h6>
- Gras :
- Italique/Emphase : <i> </i> /
- Mise en style :
- Indice et Exposant : et
- Citer du texte : <cite> </cite> et/ou <q> </q> et/ou <blockquote> </blockquote>
- Abréviation : <abbr title="Hypertext Markup Language">HTML</abbr>
- Saut de ligne (à utiliser avec parcimonie) :

- Extrait de code : <code> </code>
- Texte préformaté : <pre> </pre>
- Tracé de ligne horizontale : <hr>
- Conteneur : <div></div>

Balise Auto-Fermante

<nom />

Démo : https://leria-info.univ-angers.fr/~a.jamin/l1_dw/demos-ctd/balises-textuelles.html

Attributs universels

```
<nom attribut=" "> </nom>
```

- `class` : liste de classes. Permet d'être manipulé en CSS ou Javascript.
- `hidden` : booléen pour masquer un élément,
- `id` : identifiant UNIQUE. Permet d'être manipulé en CSS ou Javascript. Utile pour les références internes (ancres, ...).
- `style` : pour injecter du CSS (mise en forme) dans un élément (à utiliser avec parcimonie).
- ...

Documentation : https://developer.mozilla.org/fr/docs/Web/HTML/Global_attributes

Séance 2 – HTML

Ancres
Chemins
Images

Liens hypertextes

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

- `href` : url (site, fichier, mail, téléphone, référence interne)
- `target` : cible (`_blank` : dans un nouvel onglet)

Documentation : <https://developer.mozilla.org/fr/docs/Web/HTML/Element/a>

Démo : https://leria-info.univ-angers.fr/~a.jamin/l1_dw/demos-ctd/liens.html

Images

```

```

- `src` : source image
- `alt` : description de l'image (affichée si non chargée et/ou lu par certains outils)
- `width/height` : largeur/hauteur intrinsèque de l'image

Documentation : <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Img>

Démo : https://leria-info.univ-angers.fr/~a.jamin/l1_dw/demos-ctd/images.html

Séance 3 – HTML

Listes

Puces

Tableaux

Fusion de cellules

Listes

Ordonnée

```
<ol>
  <li> </li>
  <li> </li>
</ol>
```

Éléments

```
<ul>
  <li> </li>
  <li> </li>
</ul>
```

Définitions

```
<dl>
  <dt> </dt>
  <dd> </dd>
</dl>
```

Documentation :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/ol>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/ul>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/dl>

Démo : https://leria-info.univ-angers.fr/~a.jamin/l1_dw/demos-ctd/listes.html

Tableaux

- Tableau de données : `<table>` `</table>`
 - Entête des colonnes : `<thead>` `</thead>`
 - Corps du tableau : `<tbody>` `</tbody>`
 - Ligne : `<tr>` `</tr>`
 - Cellule d'entête : `<th>` `</th>`
 - Cellule : `<td>` `</td>`
- `colspan` : étendue de la cellule (#colonnes)
• `rowspan` : étendue de la cellule (#lignes)

Documentation :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/table>

Démo : https://leria-info.univ-angers.fr/~a.jamin/l1_dw/demos-ctd/tableaux.html

Séance 4 – HTML

Formulaires
Champs

<form> </form>

- `name` : nom UNIQUE.
- `action` : url qui va traiter le formulaire. Si vide ou non renseigné : dans la page courante
- `method` : méthode HTTP utilisée pour envoyer les données du formulaire (POST, GET, dialog). Si non renseigné : GET.
- ...

Documentation : <https://developer.mozilla.org/fr/docs/Web/HTML/Element/form>

GET vs POST : <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/get-vs-post/>

```
<input> </input>
```

type :

- `checkbox` : case à cocher.
- `radio` : bouton radio.
- `text` : champs saisi de texte
- `reset` : bouton RAZ champs
- `submit` : bouton d'envoi du formulaire
- ..

HTML5 : validation des champs lors de l'envoi en fonction des attributs des inputs :

- `required/pattern/min/max/...`

Documentation : <https://developer.mozilla.org/fr/docs/Web/HTML/Element/input>

```
<select> </select>
```

```
<select name=" ">  
  <option value="a">Valeur 1</option>  
  <option value="1">Valeur 2</option>  
  <option value="azerty">Valeur 3</option>  
</select>
```

Documentation : <https://developer.mozilla.org/fr/docs/Web/HTML/Element/select>

```
<label> </label>
```

Lier un label à un champ de saisie

```
<label for="myId">Mon label : </label><input id="myId" />
```

Implicitement

```
<label>Mon label : <input /></label>
```

Documentation : <https://developer.mozilla.org/fr/docs/Web/HTML/Element/label>

Formulaires

Balises Abordées en TD/TP :

- `<fieldset> <legend> </legend> </fieldset>`
- `<textarea></textarea>`
- `<button></button>`

Documentation : <https://developer.mozilla.org/fr/docs/Learn/Forms>

Démo : https://leria-info.univ-angers.fr/~a.jamin/l1_dw/demos-ctd/formulaires.html

Séance 5 et 6 – CSS

Cascading Style Sheets

Introduction / Bases
Sélecteurs / Propriétés

<https://developer.mozilla.org/fr/docs/Learn/CSS>

Démonstrations

Évolution du web (design)

<https://fabianburghardt.de/webolution/>

Différentes mises en forme à partir d'un même code HTML

<https://codepen.io/shadeed/project/full/ca9fa044eb2f01f2a378fec3898f8a8d>

Intégration

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4     <meta charset="UTF-8">
5     <title>Document</title>
6     <style>
7         /* CSS code here */
8     </style>
9     <link href="style.css" rel="stylesheet">
10 </head>
11 <body>
```

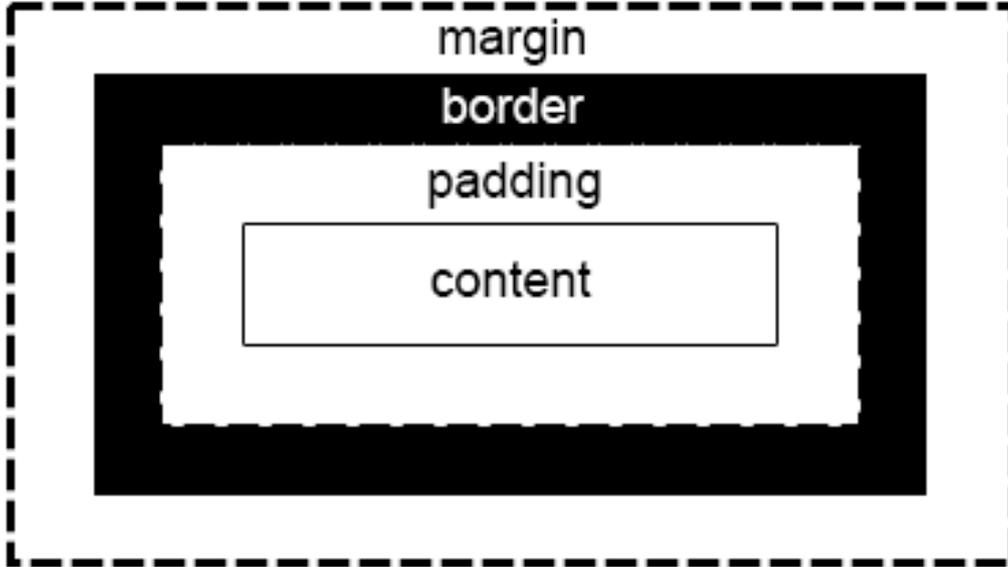
`<nom style=" " > </nom>` : à éviter.

Privilégier l'utilisation de classe(s)

Format

```
sélecteur {  
    propriété1 : valeur1;  
    propriété2 : valeur2;  
    ...  
}
```

Box



```
margin : top left down right ;  
margin : size ;
```

```
padding : top left down right ;  
padding : size ;
```

```
border : size mode color ;  
border : mode ;
```

Documentation :

- https://www.w3schools.com/css/css_boxmodel.asp
- https://www.w3schools.com/css/css_border.asp

Emboîtements

`<nom><nom></nom></nom>`

`display : inline ;` : l'un à côté de l'autre.

`display : inline-block ;` : l'un à côté de l'autre avec respect des dimensions (top and bottom margins/paddings).

`display : block ;` : l'un en dessous de l'autre.

`display : none ;` : élément masqué (n'occupe plus leur emplacement)

Documentation et exemples : https://www.w3schools.com/css/css_inline-block.asp

Sélecteurs Simples

- type : `h1`
- id : `#monId`
- classe : `.maClasse`
- universel : `*`
- attribut :
 - `[attr]` : élément possédant l'attribut `attr`
 - `[attr=valeur]` : élément dont l'attribut `attr` est exactement égal à `valeur`
 - `[attr~=valeur]` : élément dont l'attribut `attr` contient `valeur`
 - `[attr^=valeur]` : élément dont l'attribut `attr` commence par `valeur`
 - `[attr$=valeur]` : élément dont l'attribut `attr` finit par `valeur`
 - `[attr*=valeur]` : élément dont l'attribut `attr` contient au moins une occurrence de `valeur`
 - ...

Sélecteurs - Combinateurs

- Plusieurs attributs : ,
a, b : n'importe quel(s) élément(s) a et/ou b
- Voisin direct : +
a + b : n'importe quel élément b qui suit directement a
- Voisins : ~
a ~ b : n'importe quel élément b qui suit a
- Éléments enfants : >
a > b : n'importe quel élément b enfant direct d'un parent a
- Éléments descendants : " " (espace)
a b : n'importe quel élément b situé à l'intérieur de a

Sélecteurs

pseudo-classes et pseudo-éléments

- Pseudo-classes : :

Sélection selon une information d'état.

`a:hover` : élément `a` survolé par le pointeur de la souris

- Pseudo-éléments : ::

Sélection d'élément(s) ciblé(s) par la règle.

`p::first-line` : première ligne de chaque élément `p`

Grandeurs et unités

Absolues :

- `cm` : centimètres
- `mm` : millimètres
- `in` : pouces (96px)
- `px` : pixel
- `pt` : point (1in/72)
- `pc` : pica (12pt)

Relatives :

- `em` : taille de la police d'un élément
- `ex` : hauteur de la police d'un élément
- `ch` : largeur du 0
- `rem` : taille de la police de l'élément principal
- `vw` : 1% de la largeur de la fenêtre du navigateur
- `vh` : 1% de la hauteur de la fenêtre du navigateur
- `vmin` : 1% de la dimension minimale de la fenêtre du navigateur
- `vmax` : 1% de la dimension maximale de la fenêtre du navigateur
- `%` : un pourcentage par rapport à l'élément parent

Quelques propriétés

`font-*` : police de caractères

- `size` : taille
- `family` : famille (Arial, Arial Black , sans-serif, serif, ...)
- `style` : normal, italique ou oblique
- `weight` : normal, bold, lighter, ...

`text-*` : alignement et transformation du texte

- `align` : alignement (left, right, justify, ...)
- `transform` : transformation (upper, lower, none, ...)
- `decoration` : décoration (underline, overline, none, ...)

Couleurs (`color` et `background-color`) :

- `#rgb`
- `#rrggbb`
- `rgb(r, g, b)`
- **Nom des couleurs**

Positionnement des éléments

`position` :

- `static` : par défaut (les uns à la suite des autres)
- `fixed` : fixe par rapport aux dimensions de la fenêtre
- `relative` : en fonction de la position à occuper
- `absolute` : relative à son parent

Documentation : <https://developer.mozilla.org/fr/docs/Web/CSS/position>

Éléments flottants

Retiré du flux normal et placé par rapport à son conteneur.

`float` :

- `none` : ne doit pas flotter
- `left` : doit flotter côté gauche du conteneur
- `right` : doit flotter côté droit du conteneur
- `inline-start` : doit flotter du côté du début du bloc
- `inline-end` : doit flotter du côté de la fin du bloc

Documentation : <https://developer.mozilla.org/fr/docs/Web/CSS/float>

Propriétés personnalisées

Déclaration de valeurs globales (:root)

```
--myColor : rgb(255, 255, 0, 0.5) ;
```

Utilisation au sein du CSS

```
background-color : var(--myColor) ;
```

Documentation : https://developer.mozilla.org/fr/docs/Web/CSS/--*

Calcul

```
width : calc(50% - 5px) ;
```

Opérateurs : + - * /

Constantes prédéfinies : e pi infinity -infinity NaN

Recommandations : séparer les opérateurs par des espaces
(obligatoire pour l'addition et la soustraction)

Documentation : <https://developer.mozilla.org/fr/docs/Web/CSS/calc>

Animations

`transform :`

`transition : transform ... ;`



Démos

`animation :`  Démos

Documentation :

- <https://developer.mozilla.org/fr/docs/Web/CSS/transform>
- <https://developer.mozilla.org/fr/docs/Web/CSS/transition>
- <https://developer.mozilla.org/fr/docs/Web/CSS/animation>

Autres

Propriétés abordées en TD/TP :

- `list-style-type`
- `table-layout`
- `writing-mode`
- `visibility`
- `font-variant`
- `border-collapse`
- `box-shadow`
- `vertical-align`
- `line-height`

Pseudo-classe abordées en TD/TP :

- `:link`
- `:hover`
- `:visited`
- `:nth-child()`
- `:last-child`
- `:first-child`
- `:last-of-type`
- `:nth-of-type()`

Séances 7 8 et 9 – CSS

Blocs
Grilles
FlexBox

Blocs

`display : block|inline-block`

Utiliser les propriétés des éléments

- `position` : comment l'élément est positionné dans le document (*cf. slide position*)
- `width/height` : dimensions de l'élément
- `top/bottom/left/right` : distance par rapport à un élément (*en fonction de position*)
- `z-index` : ordre de superposition

Grilles

`display : grid|inline-grid`

Définition de la grille

- `grid-template-rows` : définition des lignes de la grille
- `grid-template-columns` : définition des colonnes de la grilles

Définition du placement (et propriétés) des éléments sur la grille

- `grid-row-start|grid-row-end|grid-row` : placement sur les lignes de la grille
- `grid-column-start|grid-column-end|grid-column` : placement sur les colonnes de la grille
- `grid-area` : combinaison de `grid-row` et `grid-column`
- `order` : ordre de superposition et/ou d'apparition.

display : block | inline-block

VS.

display : grid | inline-grid

Démonstration

https://leria-info.univ-angers.fr/~a.jamin/l1_dw/demos-ctd/css-block-layout.html

VS.

https://leria-info.univ-angers.fr/~a.jamin/l1_dw/demos-ctd/css-grid-layout.html

Apprendre les grilles en jouant : <https://cssgridgarden.com/#fr>

Documentation `grid` : https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Grid_Layout

display : flex

- `flex-direction` : affichage en ligne ou colonne (ordre normal ou inverse)
- `flex-wrap` : contrainte (ou non) à une ligne
- `flex-flow` : combinaison de `flex-direction` et `flex-wrap`
- `align-* | justify-*` : gestion des alignements (`content`, `self`, ...)
- `order` : ordre de superposition et/ou d'apparition.

Démonstration : https://leria-info.univ-angers.fr/~a.jamin/l1_dw/demos-ctd/css-flexbox-layout.html

Apprendre en jouant : <https://flexboxfroggy.com/#fr>

Documentation : https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Flexible_Box_Layout