

HTML / CSS

Moussa Camara

Msa.camara@gmail.com



OBJECTIF DE LA FORMATION

- ★ Présentation des nouvelles fonctionnalités HTML5/CSS3.
- ★ Mise en place de site statique avec une architecture HTML5/CSS3.
- ★ Base du Responsive Design.
- ★ Présentation du bootstrap twitter.

HTML 5 / CSS 3: C'EST QUOI DÉJÀ ?



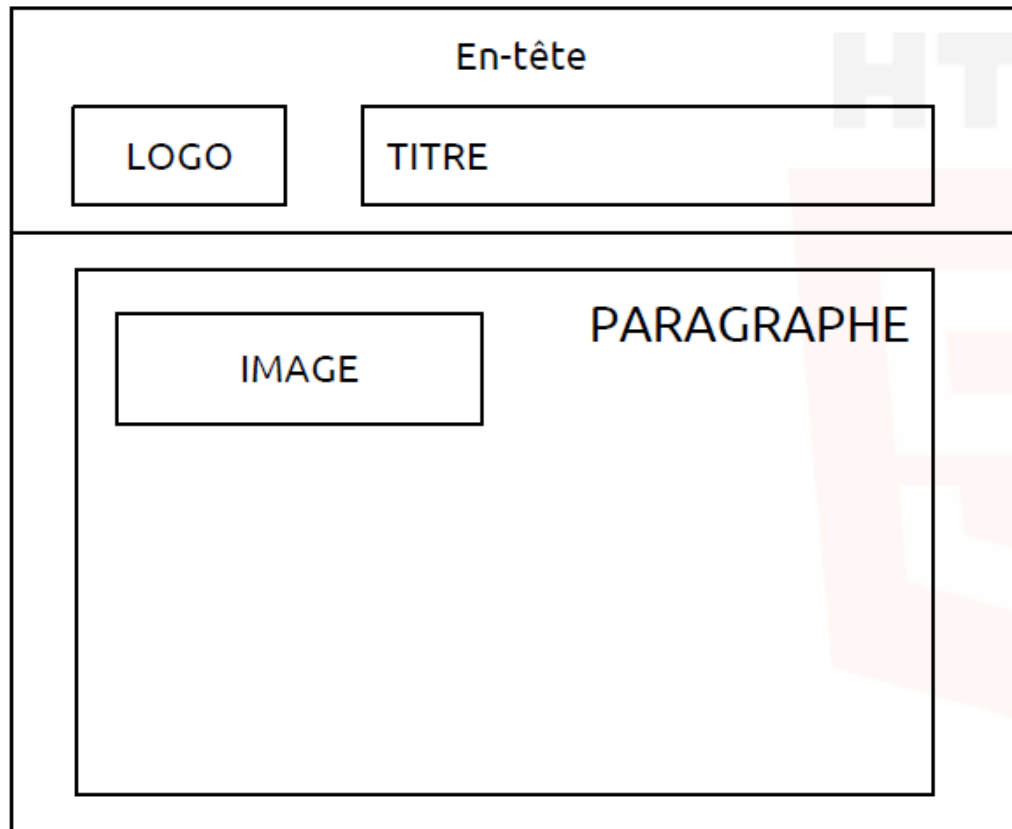
DEUX LANGAGES QUI VONT ENSEMBLE

- ★ Le HTML (HyperText Markup Language) sert à structurer sémantiquement le contenu des pages internet.
- ★ Le CSS sert à mettre en page le contenu des pages internet.

HTML 5 / CSS 3



HTML



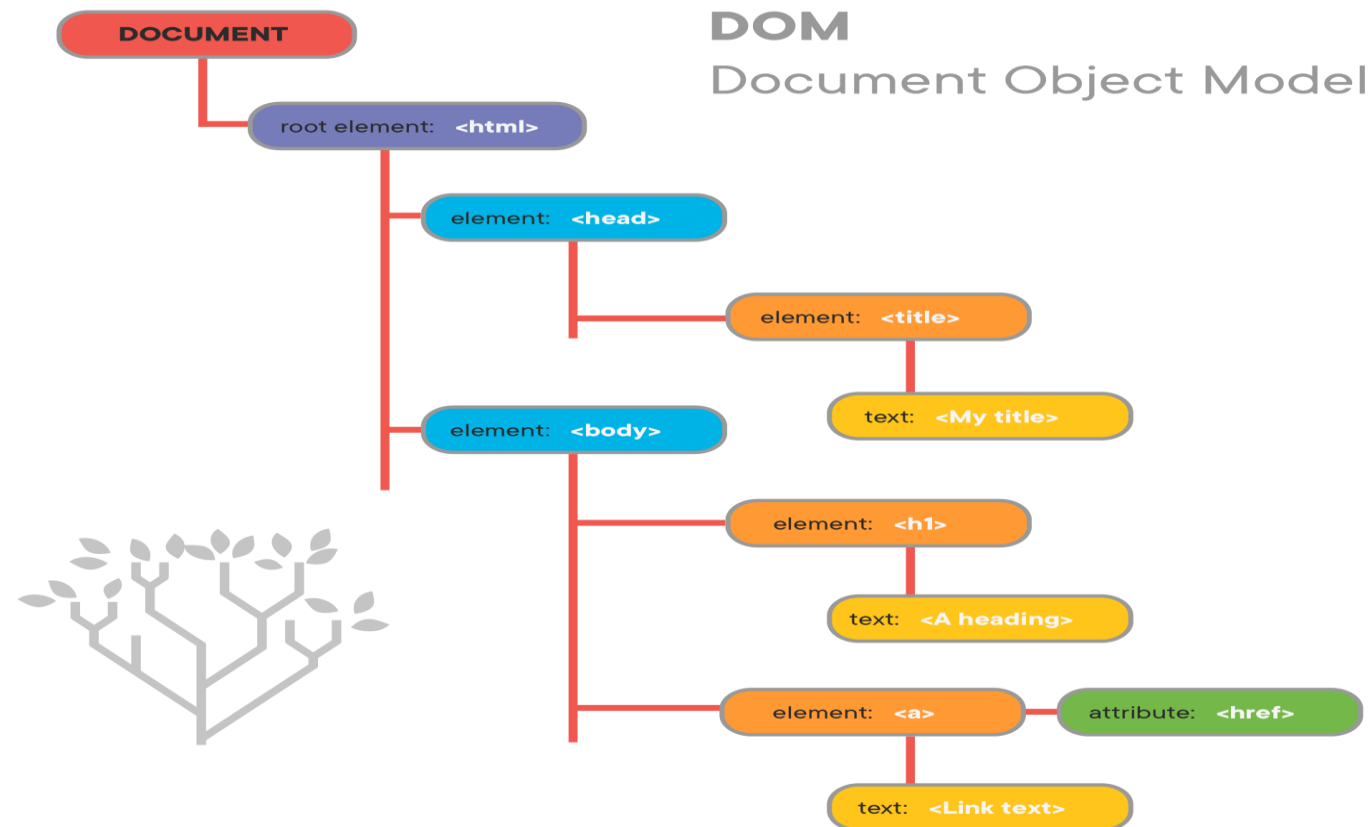
CSS

- ★ COULEURS
- ★ FAMILLE DE POLICE
- ★ TAILLE DE POLICE
- ★ *STYLE DE POLICE*
- ★ BORDURES
- ★ OMBRE
- ★ ...

LE DOM -DOCUMENT OBJECT MODEL



- Le **Document Object Model** ou **DOM** (pour modèle objet de document) est une interface de programmation pour les documents HTML, XML et SVG.
- Il représente le document comme un ensemble de nœuds et d'objets possédant des propriétés et des méthodes. Les nœuds peuvent également avoir des gestionnaires d'événements qui se déclenchent lorsqu'un événement se produit.



UN PEU D'HISTOIRE



- ★ 1990 : création par Tim Berner-Lee, premiers documents hypertextes
- ★ 1993 : premier navigateur (Mosaic), ~50 sites Web
- ★ 1994 : navigateur Netscape, HTML avec plugins
- ★ 1995 : Microsoft lance Internet Explorer
- ★ 1999 : standardisation de HTML 4.0 et CSS 2.1
- ★ 2004 : 2ème guerre des standards avec firefox
- ★ 2006 : Début de la réflexion HTML5
- ★ 22 Mai 2011 : clôture des ajouts de fonctionnalités
- ★ 2014: finalisation de la spécification en 2014

HTML5



- HTML: l'acronyme signifie **HyperText Markup Language**, ce qui signifie en français "**langage de balisage d'hypertexte**".
- L'**HTML** est un langage **informatique** utilisé sur l'internet. Ce langage est utilisé pour créer et représenter le contenu d'une page web et sa structure.

PREMIERS PAS



- On va créer notre première page html
 - Index.html
 - On aura l'extention **.html** (Obligatoire)

HTML5

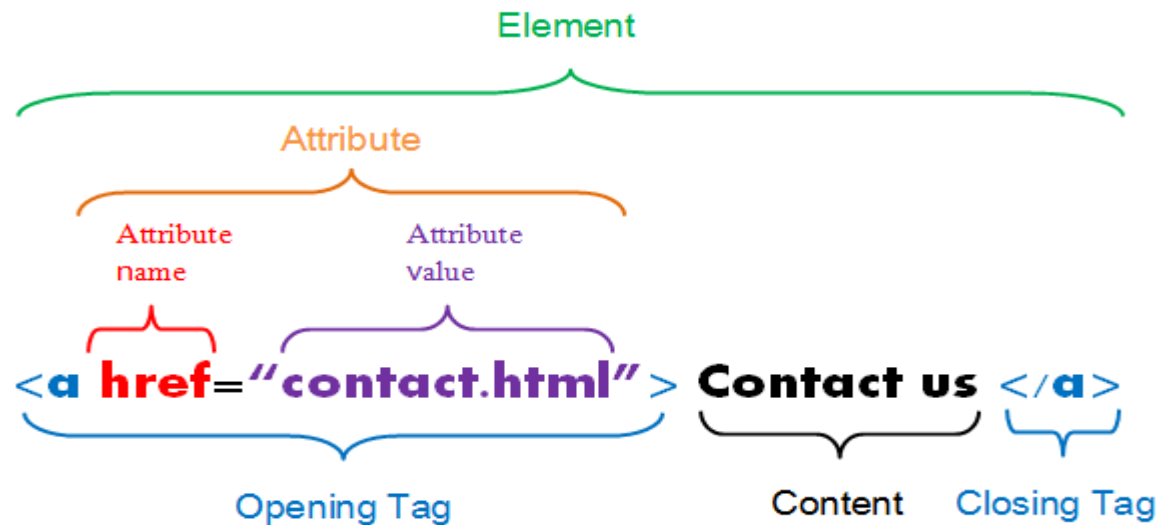


- Le HTML repose sur **une syntaxe simple et rigide, dont l'unité de base est la balise**. Les balises vont toujours par deux, une pour ouvrir l'élément, l'autre pour le fermer.
- Elles se construisent toujours de la même façon :
 - un chevron < pour ouvrir la balise
 - Le nom de l'élément contenu dans la balise
 - Un attribut, si nécessaire, pour qualifier l'élément.
 - un chevron > pour fermer la balise

HTML5



- Il y a trois termes dont vous devez absolument comprendre le sens en HTML. Ce sont les termes **élément**, **balise** et **attribut**.



PREMIERS PAS



Maintenant les balises...

```
<!DOCTYPE html>  
<html>  
...  
</html>
```

Le Doctype:

Tout comme les pages HTML ou XHTML, les documents HTML5 nécessitent une déclaration Doctype indiquant la méthode standard de rendu par le navigateur.

STRUCTURE



```
<!DOCTYPE html>  
<html>
```

```
  <head>  
    <meta charset="utf-8" />  
    <title>Notre titre</title>  
  </head>
```

```
  <body>  
    <!-- Corps de la page -->  
  </body>
```

```
</html>
```

L'en-tête <head> :

Cette balise donne des informations générales sur la page. Les informations que contient l'en-tête ne sont pas affichées sur la page, ce sont simplement des informations générales à destination de l'ordinateur.

A noter: Il y a des balises traditionnelles et des balises "auto-fermantes".

POURQUOI HTML 5 ?



- ★ Allègement du code
- ★ De nouvelles fonctionnalités
- ★ Interopérabilité (plus de pluggins, compatibilité mobiles, etc...)
- ★ Une organisation en modules

HTML5 – ALLÈGEMENT DU CODE



AVANT	APRES
<code><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"></code>	<code><!DOCTYPE html></code>
<code><html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"></code>	<code><html lang="fr"></code>
<code><meta http-equiv="Content-Type" content="text/html; charset=utf-8" /></code>	<code><meta charset="utf-8" /></code>
<code><link rel="stylesheet" type="text/css" href="design.css"/></code>	<code><link rel="stylesheet" href="design.css" /></code>
<code><script type="text/javascript" src="script.js"></script></code>	<code><script src="script.js"></script></code>

HTML



AVANT

```
<div id="header">
```

```
<div id="nav">
```

```
<div class="article">
```

```
<div class="section">
```

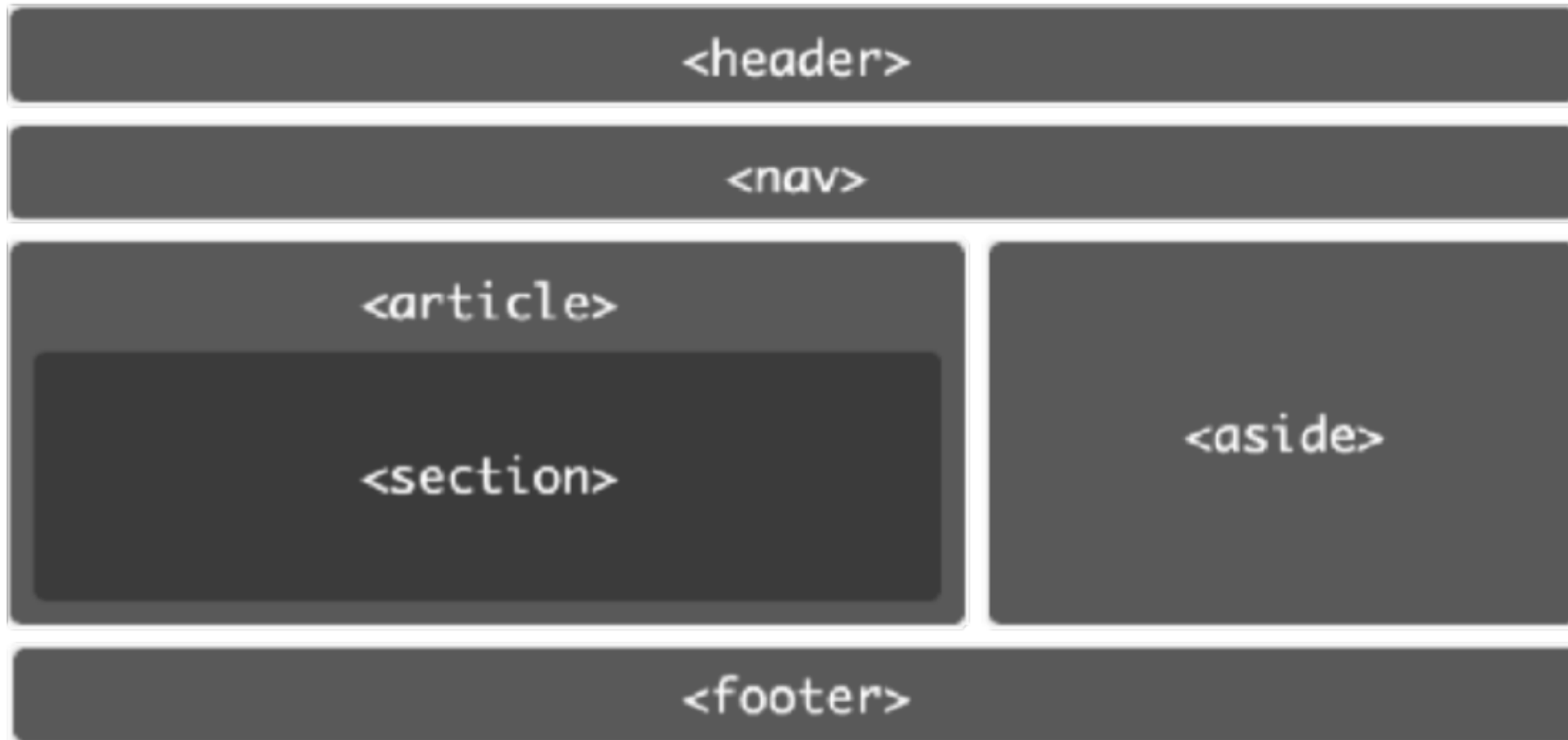
```
<div id="sidebar">
```

```
<div id="footer">
```

HTML5



APRES



HTML5



- ★ `<header>` : Qui indique que l'élément est une en-tête
- ★ `<footer>` : Qui indique que l'élément est un pied-de-page
- ★ `<nav>` : Qui indique un élément de navigation tel qu'un menu
- ★ `<aside>` : Qui correspond à une zone secondaire non liée au contenu principal de la page
- ★ `<article>` : Qui représente une portion de la page qui garde un sens même séparée de l'ensemble de la page (comme un article de blog par exemple)

HTML 5



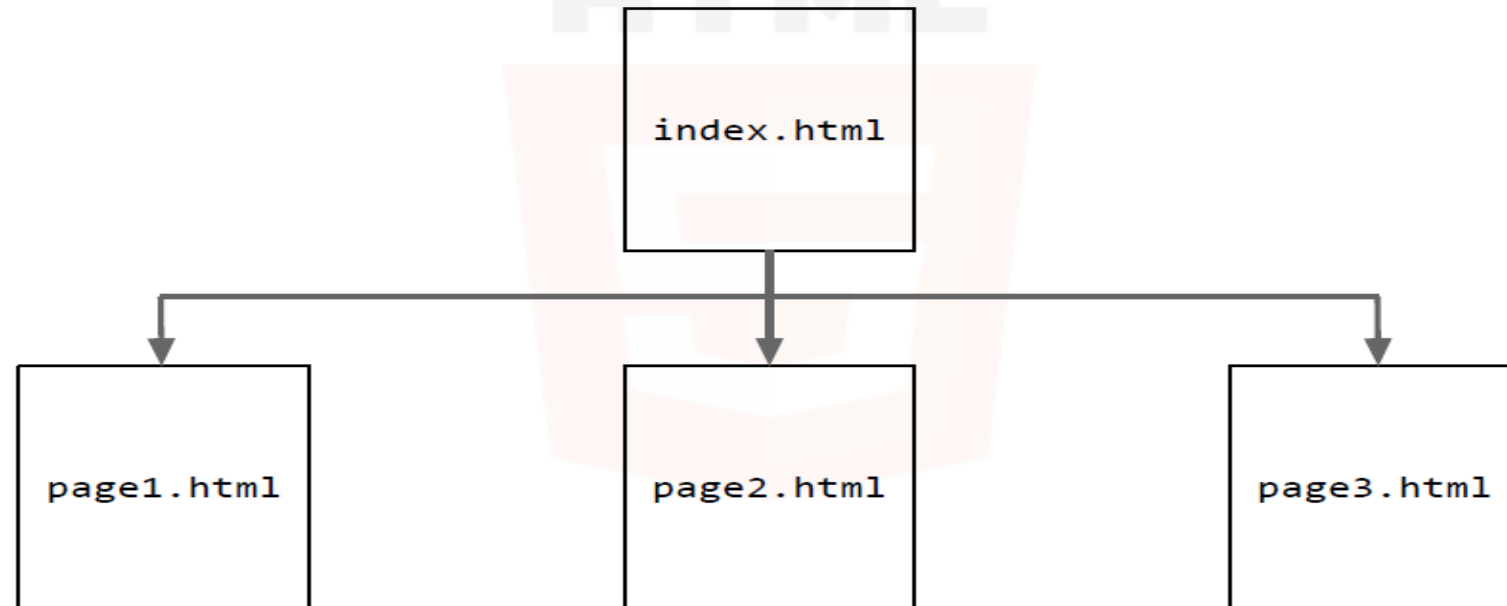
- ★ Le HTML 5 a été découpé selon les modules suivants :
 - Structuration du document,
 - Gestion de pages par modèles (templates),
 - Html forms (Web form 2.0),
 - Audio/video,
 - Canvas, effets graphiques,
 - Mode offline et stockage local

- ★ Cela englobe la structure et également les API Javascript pour y accéder.

HTML5



Une page HTML = un fichier texte .html



QUELQUES ELEMENTS HTML 5



Les titres par ordre d'importance **sémantique**:

```
<h1>Notre titre</h1>
```

```
<h2>Notre titre</h2>
```

```
<h3>Notre titre</h3>
```

```
<h4>Notre titre</h4>
```

```
<h5>Notre titre</h5>
```

```
<h6>Notre titre</h6>
```

QUELQUES ELEMENTS HTML 5



- **Les paragraphes:**

<p>

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod

tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,

quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

</p>

LIENS INTERNES ET LIENS EXTERNES



Un **lien interne** est un lien créé entre deux pages d'un même site web tandis qu'un **lien externe** est un lien menant d'un site web vers un autre site web.

Dans tous les cas, pour créer un lien, nous allons utiliser l'élément « a » accompagné de son attribut **href** (pour Hypertext Reference) qui sert à indiquer la cible (c'est à dire la destination) du lien.

Les liens hypertext:

```
<p>Ceci est mon paragraphe, dans lequel il y a un  
<a href="http://example.com/" title="Lien externe">un lien  
vers un site internet</a> et un <a href="page1.html" title="  
Lien interne">lien vers une autre page de mon site.</p>
```

A noter: il y a des balises de type "bloc" et des balise de type "inline".

LIEN ANCRE



Les ancres, des liens spéciaux:

Les liens de type ancre sont des liens menant à un autre endroit d'une même page web. Ils peuvent être utiles dans le cas d'une page web très longue pour donner à vos visiteurs un accès rapide à une section en particulier par exemple.

```
<h1 id="titre-page">Titre de la page</h1>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Aenean eget dictum magna. Mauris gravida risus vitae  
placerat cursus. Fusce vel nulla augu...</p>
```

```
<a href="#titre-page" title="Vers le haut">Retour en  
haut</a>
```

AUTRES LIENS



- Mailto:
 - L'URL mailto: permet de générer un nouveau mail sans que l'utilisateur n'ait à saisir l'adresse du destinataire.
 - `envoie un mail `

Télécharger un fichier

- Pour télécharger un fichier, on a juste définir l'url dans notre href
 - `télécharger`

QUELQUES BALISES



Mettre en valeur du texte **sémantiquement**:

<code></code>	Texte important
<code></code>	Texte à faire valoir
<code><code></code>	Code informatique
<code><blockquote></code>	Citation
<code><kbd></code>	Saisie de clavier d'ordinateur
<code><dfn></code>	Définition

LES TYPES DE LISTES



- En HTML, les listes vont avoir deux grands intérêts pour nous : on va pouvoir les utiliser pour créer des menus ou, dans leur forme brute, pour mieux présenter du contenu pour un blog par exemple. Il existe trois grands types de listes en HTML : **les listes ordonnées**, **les listes non-ordonnées** et un dernier type un peu particulier : **les listes de définition**.
- La différence **entre les listes ordonnées et non-ordonnées** est que les listes ordonnées possèdent un aspect de subordination, d'ordre logique, de classement tandis que ce n'est pas le cas pour les listes non-ordonnées.

Les listes:

Listes à puces

```
<ul>
  <li>Paris</li>
  <li>Lyon</li>
  <li>Marseille</li>
</ul>
```

Listes ordonnées

```
<ol>
  <li>Paris</li>
  <li>Lyon</li>
  <li>Marseille</li>
</ol>
```

LISTES NON ORDONNÉES (À PUCES)



Pour créer une liste non-ordonnée, on va avoir besoin de deux nouveaux éléments : l'élément `ul` (abréviation de `unordered list`), qui va contenir toute la liste et l'élément `li` (pour `list item`) que l'on va utiliser pour créer chaque élément de la liste.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>test</title>
5 </head>
6 <body>
7   <h1 id="titre-page">Mon titre </h1>
8   <h2 class="titre-section"> Mon autre titre</h2>
9   <ul>
10    <li>Banane</li>
11    <li>Pomme</li>
12    <li>Ananas</li>
13    <li>Raisin</li>
14  </ul>
15 </body>
16 </html>
```

LISTES ORDONNÉES



Pour créer une liste ordonnée maintenant, nous allons simplement remplacer l'élément `ul` par l'élément `ol` (pour ordered list).

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>test</title>
5 </head>
6 <body>
7   <h1 id="titre-page">Mon titre </h1>
8   <h2 class="titre-section"> Mon autre titre</h2>
9   <ol>
10    <li>Banane</li>
11    <li>Pomme</li>
12    <li>Ananas</li>
13    <li>Raisin</li>
14  </ol>
15 </body>
16 </html>
```

LISTES DE DÉFINITION



Les listes de définition sont utilisées pour définir des termes.

Pour créer une liste de définition, il va nous falloir utiliser l'élément `dl` (pour definition list), l'élément `dt` (pour definition term) et l'élément `dd` pour la définition en soi.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>test</title>
5 </head>
6 <body>
7   <h1 id="titre-page">Mon titre </h1>
8   <h2 class="titre-section"> Mon autre titre</h2>
9   <dl>
10     <dt>Banane</dt>
11     <dt>Pomme</dt>
12     <dt>Ananas</dt>
13     <dt>Raisin</dt>
14   </dl>
15 </body>
16 </html>
```

LISTES IMBRIQUÉES



- Le HTML nous offre la possibilité d'imbriquer les listes les unes dans les autres très simplement. Pour imbriquer des listes, il suffit de commencer une liste, puis d'en ouvrir une seconde à l'intérieur d'un élément de la première (on peut évidemment imbriquer plus de deux listes en répétant le même processus).

```
<!DOCTYPE html>
<html>
<head>
  <title>test</title>
</head>
<body>
  <h1 id="titre-page">Mon titre </h1>
  <h2 class="titre-section"> Mon autre titre</h2>
  <ol>
    <li>Banane</li>
    <li>Pomme</li>
    <li>Ananas</li>
    <li>
      <ul>
        <li>Raisin</li>
        <li>Orange</li>
        <li>
          <dl>
            <dd>Fraise</dd>
            <dt>Fraisesss</dt>
          </dl>
        </li>
      </ul>
    </li>
  </ol>
</body>
</html>
```


INSERTION D'IMAGE



L'insertion d'images en HTML va se faire au moyen de l'élément HTML **img**. Cet élément est représenté par une **balise orpheline**.

Le **src** va prendre comme valeur l'adresse de l'image.

L'**alt** va contenir un texte alternatif décrivant l'image

Les images:

HTML

[Logo](#)

```

```

step4-images

QUELQUES ELEMENTS

L'élément **figure** est un élément de type block.

On l'utilise autour d'images, de contenus audio ou vidéo ou de blocs de code.

L'idée est d'utiliser l'élément figure pour envelopper du contenu lorsque celui-ci n'est pas strictement décoratif.

A l'intérieur de l'élément figure, on peut utiliser l'élément HTML **figcaption** qui va nous permettre d'accoler une légende à notre contenu.

Les figures:

[Image](#)

```
<figure>
  
  <figcaption>Obtenir des badges</figcaption>
</figure>
```

LES TABLEAUX

icon	text	text	full text	icon	full text
👉	1	-1	similar	👉	bat
👉	10	-10	light	👉	light
👉	1	-1	bad	👉	try
👉	10	-10	week	👉	evil
👉	100	-100	shock	👉	terrible
👉	1	-1	fun	👉	roll
👉	10	-10	cool	👉	1
👉	10	-10	mad	👉	1
👉	1	-1	neutral	👉	data
👉	1	-1	neutral	👉	neutral
👉	1	-1	weak	👉	imagine

Un tableau est un ensemble structuré de données (**table de données**) présentées en lignes et colonnes.

Un tableau vous permet de retrouver rapidement et facilement des valeurs au croisement entre différents types de données.

```
<table>
  <caption>Liste de vos cours</caption>
  <tr>
    <th>Titre</th>
    <th>Création</th>
    <th>Action</th>
  </tr>
  <tr>
    <td>HTML5 / CSS3</td>
    <td>le 14/03/2014</td>
    <td>Modifier</td>
  </tr>
</table>
```

LES TABLEAUX

icon	text	text	full text	icon	full text
☹	1	-1	similar	☹	bat
☹	0	-0	grim	☹	angel
☹	1	-1	bad	☹	try
☹	0	-0	week	☹	evil
☹	00	0-0	shock	☹	twisted
☹	7	-7	rim	☹	roll
☹	0	0-0	cool	☹	1
☹	x	-x	mad	☹	7
☹	P	-P	rizz	☹	dece
☹	1	-1	neutral	☹	arise
☹	0	-0	twink	☹	imagine

```
<table>
  <caption>Liste de vos cours</caption>
  <thead>
    <tr>
      <th>Titre</th>
      <th>Création</th>
      <th>Action</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>HTML5 / CSS3</td>
      <td>le 14/03/2014</td>
      <td>Modifier</td>
    </tr>
  </tbody>
</table>
```

```
<tfoot>
  <tr>
    <th>Nom</th>
    <th>Âge</th>
    <th>Pays</th>
  </tr>
</tfoot>
```

LES TABLEAUX- STEP5-TABLES

- On peut aussi combiner des cellules de notre tableau

```
<table>
  <tr>
    <th>nom</th>
    <th>prenom</th>
    <th>age</th>
  </tr>
  <tr>
    <td>lupin</td>
    <td>Arsène</td>
    <td rowspan="2">15</td>
  </tr>
  <tr>
    <td>Machin</td>
    <td>Bidule</td>
    <td colspan="2">col1 clo2</td>
  </tr>
</table>
```

```
table{
  border-collapse: collapse;
}

tr, td{
  border:1px solid black;
}
```

RECAP

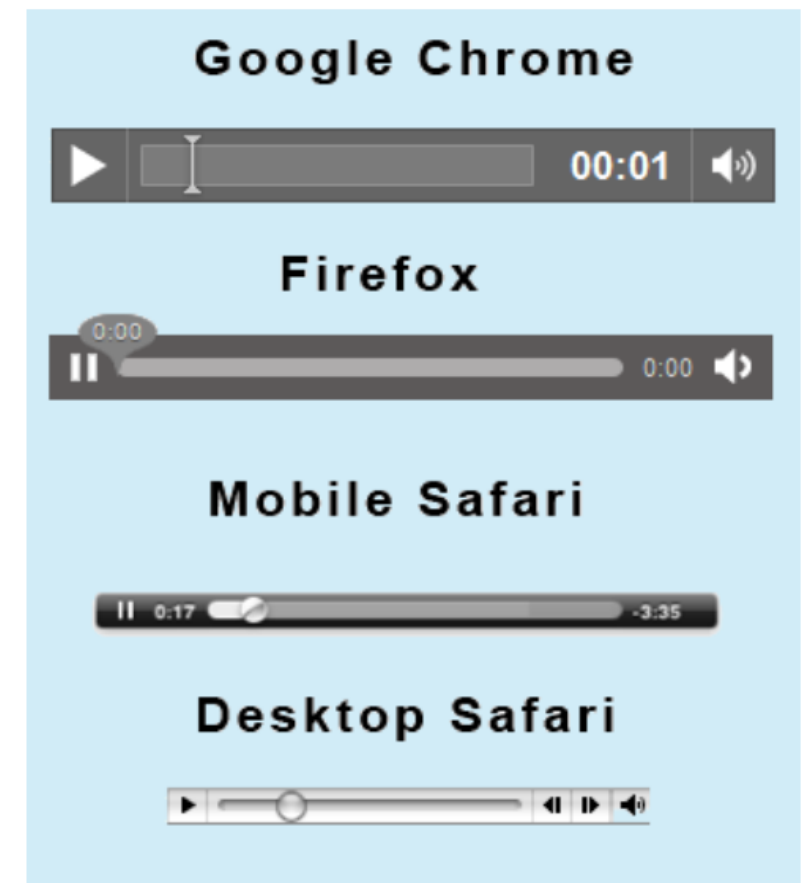
Balises	Description
table	Tableau
thead	en tête du tableau
th	cellule d'en tête
tr	ligne du tableau
td	cellule du tableau
col	colonne du tableau
caption	titre du tableau
col	groupe de colonne
tbody	corp du tableau
colspan	titre et nombre de case
tfoot	bas du tableau

HTML5 – BALISE AUDIO



<audio>

Cette balise est l'équivalent de la balise video mais pour l'audio. En 3 lignes de code vous avez un lecteur MP3! Nous pouvons d'ailleurs voir sur l'image ci-dessous que chaque navigateur utilise un design qui lui est propre pour styliser son lecteur. Dans tous les cas il vous est possible de créer votre propre design si vous préférez un rendu uniforme quel que soit le navigateur utilisé.



HTML5 – BALISE AUDIO



```
<!DOCTYPE html>
<html>
<head>
  <title>test</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <a href="#ancree1">ancree1 </a>
  <a href="#ancree2">ancree2 </a>
  <a href="#ancree3">ancree3 </a>
  <a href="#ancree4">ancree4 </a>
  <h1 id="titre-page">Mon titre </h1>
  <h2 class="titre-section"> Mon autre titre</h2>

  <audio src="future-feat-drake-life-is-good-official-music-video-lyrics.mp3" controls></audio>

</body>
</html>
```

L'audio:

<audio src="audio.mp3" controls></audio>

<audio src="audio.mp3" controls>

Veillez mettre à jour votre navigateur !

</audio>

- ★ controls
- ★ width
- ★ loop
- ★ autoplay
- ★ preload: indique si la musique peut être préchargée dès le chargement de la page ou non. Cet attribut peut prendre les valeurs :
 - auto (par défaut)
 - metadata
 - none

Son

HTML 5 -BALISE VIDÉO



<video>

Cette balise intègre directement un lecteur vidéo dans la page, avec des boutons Lecture, Pause, une barre de progression, du volume... Un vrai petit Youtube intégré à votre page et natif au navigateur !



HTML 5 -BALISE VIDÉO



```
<!DOCTYPE html>
<html>
<head>
  <title>test</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <a href="#ancre1">ancre1 </a>
  <a href="#ancre2">ancre2 </a>
  <a href="#ancre3">ancre3 </a>
  <a href="#ancre4">ancre4 </a>
  <h1 id="titre-page">Mon titre </h1>
  <h2 class="titre-section"> Mon autre titre</h2>
  <video width="640" height="360" src="la-video.mp4" controls>
</video>
</body>
</html>
```

Les vidéos:

`<video src="video.ogg"></video>`

`<video controls poster="poster.jpg" width="600">`

`<source src="video.mp4" />`

`<source src="video.webm" />`

`<source src="video.ogv" />`

Il est temps de mettre à jour votre navigateur !

`</video>`

[Vidéo](#) [Poster](#)

- ★ controls
- ★ poster
- ★ width
- ★ height
- ★ loop
- ★ autoplay
- ★ preload
 - auto (par défaut)
 - metadata
 - none

TEST DE VALIDER DE SON CODE

- Pour vérifier la validité d'un code HTML ou CSS, le w3c (World Wide Web Consortium), c'est-à-dire l'organisme qui gère l'évolution des langages lus par les navigateurs comme le HTML et le CSS entre autres, a mis à disposition des validateurs de code qui sont gratuits.
 - Pour tester son HTML: <http://validator.w3.org/>
 - Pour tester son Css: <http://jigsaw.w3.org/css-validator/>

CSS3



Cascading Style Sheets (CSS) est un langage de feuille de style utilisé pour décrire la présentation d'un document écrit en HTML ou en XML:eXtensible Markup Language (on inclut ici les langages basés sur XML comme SVG (**Scalable Vector Graphics**) est un langage de balisage XML décrivant des images vectorielles) ou XHTML).

CSS décrit la façon dont les éléments doivent être affichés à l'écran, sur du papier, en vocalisation, ou sur d'autres supports.

POURQUOI LE CSS ?



- Les avantages de CSS:
 - simplifie le code HTML ce qui permet de consommer moins de bande passante lors du chargement de la page.
 - permet à l'utilisateur de personnaliser une page web
 - facilite le positionnement dans les moteurs de recherche
 - rend également le code HTML plus "propre" facilitant le travail des robots d'indexation.
 - permet de modifier les caractéristiques des éléments d'un site web en fonction de la nécessité de la situation.

CSS 3



- ★ Backgrounds et bordures
- ★ Gestion des boîtes (box-sizing, resize, etc...)
- ★ Effets de texte
- ★ Transformations 2D/3D
- ★ Mise en page en colone multiple
- ★ Etc ...

CSS3 – LES SÉLECTEURS



Par balise HTML

p, div, header, input,
etc...

Par attribut HTML

#conteneur, .first

Par pseudo-classe

:hover, :active, :focus, :
required, :invalid, etc...

CSS3 – LES SÉLECTEURS



- On a maintenant la structure de notre page HTML, on va définir une mise en page.
- On va voir les éléments fondamentaux du css:
 - Un **sélecteur** CSS est un mot-clef qui permet de désigner une catégorie d'éléments de la page éventuellement de nature différente ou une relation entre deux éléments.
 - Les **propriétés** vont nous permettre de choisir quel(s) aspect(s) (ou "styles") d'un élément HTML on souhaite modifier.
 - La **valeur** permet de définir le comportement de la propriétés

```
selecteur {  
    propriete: valeur;  
    propriete: valeur;  
    ...  
}
```

CSS3 – LES SÉLECTEURS



- **Sélecteur universel**

- Le sélecteur étoile (*) sélectionne toutes les balises du document web.
- Il correspond à un élément de n'importe quel type.

```
* {  
    color: blue;  
}
```

- **Sélecteur de type**

- Lorsqu'on utilise `h2` pour changer le style de tous les titres de niveau 2, on utilise un sélecteur de type

```
h2 {  
    color: blue;  
}
```

- **Sélecteurs d'attributs**

- Les deux attributs les plus couramment utilisés sont **class** et **id**

```
.nom-de-la-classe {  
    color: yellow;  
}
```

CSS3 – EMBLEMES POSSIBLES



PAS TOP

Dans le code HTML

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

Dans la balise <head>

PAS TOP

```
<head>  
  <style> ... </style>  
</head>
```

Dans une feuille de style

TOP

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

LES ATTRIBUTS ID ET CLASS



- **Id**(identifiant unique pour l'ensemble du document) désigne un élément unique dans la page.
 - Le but de cet attribut est de pouvoir identifier un élément lorsqu'on crée un lien, avec un fragment et qu'on souhaite le manipuler avec un script ou qu'on le met en forme avec CSS.
- **Class**(liste de classes associées à l'élément courant) peut-être utilisé plusieurs fois.
 - Les classes permettent de manipuler les éléments, via CSS

```
<!DOCTYPE html>
<html>
<head>
  <title>test</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <a href="#ancre1">ancre1 </a>
  <a href="#ancre2">ancre2 </a>
  <a href="#ancre3">ancre3 </a>
  <a href="#ancre4">ancre4 </a>
  <h1 id="titre-page">Mon titre </h1>
  <h2 class="titre-section"> Mon autre titre</h2>
  <div id="ancre1">
    Mon ancre 1
  </div>
  <div id="ancre2">
    Mon ancre 2
  </div>
  <div id="ancre3">
    Mon ancre 3
  </div>
  <div id="ancre4">
    Mon ancre 4
  </div>
</body>
</html>
```


QUELQUES ELEMENT: LES DIV ET SPAN



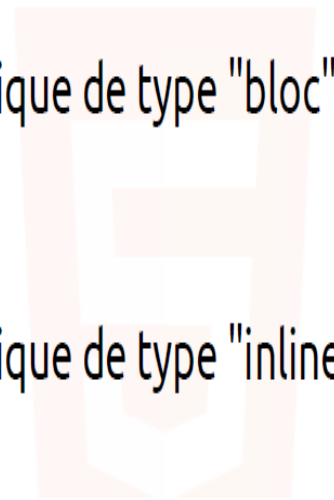
Le **div** (division du document) a un rôle structurant. C'est un élément de type block. C'est un conteneur qui va encapsuler d'autres éléments.

Le **span** est un élément de type inline. Le plus souvent vous vous en servirez pour apporter du style à un contenu textuel.

Balises génériques:

- `<div>` Balise générique de type "bloc"
- `` Balise générique de type "inline"

HTML



LES ÉLÉMENTS DE TYPE BLOCK ET INLINE



- En HTML, tout élément est soit de type block, soit de type inline.
 - **Block**: Occupe toute la largeur disponible. Lorsque 2 éléments blocs se suivent dans une page, ils sont positionnés (par défaut) l'un **sous** l'autre.
 - **Inline**: N'occupe que la largeur indispensable à l'affichage du contenu et ne provoque pas de retour à la ligne. Lorsque 2 éléments en-ligne se suivent dans une page, ils sont positionnés l'un **à côté** l'autre (si la largeur de page le permet).

Éléments block	Éléments inline
p	em
h1, h2, h3...	strong
header, article, footer...	mark
ol, ul, dl	a
table	img

CSS3 – LES PROPRIÉTÉS



Couleur du texte

```
color: red;  
color: #F00;  
color: rgb(255, 0, 0);  
color: rgba(255, 0, 0, 0.5);
```

Taille de police

```
font-size: 14px;  
font-size: 1.5em;  
font-size: 20%;  
font-size: 20pt;
```

Famille de police

```
font-family: Arial, monospace;  
  
oblique;
```

Style de police

```
font-weight: normal, bold;  
font-style: italic,  
  
text-shadow: 5px 5px solid;
```

CSS3 – LES PROPRIÉTÉS



- **font-family** permet de choisir la police de notre texte.
 - `body { font-family: Arial, Helvetica, sans-serif; }`
- **font-style** permet de fixer l'inclinaison d'un texte.
 - Normal (valeur par défaut) ;
 - Italic (change le texte en italique) ;
 - Oblique (penche le texte) ;
 - Inherit (héríte des propriétés de l'élément parent).
 - `body { font-style: italic; }`
- **font-size** : valeur en unité permet de définir la taille d'une police
 - `body { font-size: 18px; }`

CSS3 – LES PROPRIÉTÉS



- **text-align** : permet d'aligner le texte à gauche, droite ou milieu.
 - Les valeurs possibles sont : left, right, center, justify (comme sur Word)
- **text-transform** permet de transformer le texte en majuscule,minuscule
 - Valeurs possibles :
 - none
 - capitalize
 - uppercase
 - lowercase

CSS3 – LES PROPRIÉTÉS



- **text-decoration** : permet de changer différents valeurs de « décoration »
 - Valeurs possibles :
 - none: permet d'empêcher l'héritage de la propriété et plus particulièrement de supprimer le soulignement par défaut des liens
 - overline : surlignement
 - underline : soulignement
 - line-through: texte barré
- **text-indent** : valeur (négative ou positive) permet de spécifier un décalage de la première ligne d'un texte

CSS3 – LES PROPRIÉTÉS



- **list-style-type** : permet de spécifier le type de puce ou de numérotation (appliqué sur le ul / ol)
 - Valeurs possibles
 - none (pas de puce), disc, circle, square
 - Valeurs possibles pour les listes des ordonnées
 - decimal (1.), decimal-leading-zero (01.), lower-roman (vii), upper-roman (VII)
- **list-style-image** : pour utiliser une image personnalisée
 - Valeur : le chemin d'une image url(« cheminversmonimage.png")

EXO – EN LIVE

- On va créer des elements avec les types block et inline
- Définir un identifiant et des classes sur les propriétés
 - Afficher les elements de l'id et classes de couleur différente
- Créer un tableau d'articles:
 - Structure de table (header): titre,auteur, date de sortie et résumé
 - Données:
 - J'aime le Html, Afpa, 25/09/2021, Ceci est le résumé de mon article1
 - J'aime aussi le CSS, Moussa Camara, 01/04/2020, Ceci est mle résumé de l'article2
 - HTML/CSS, Tim Berner, 01/01/1996, Language hyperText

HTML 5 - STEP 1 - STRUCTURE

- Exercice
 - Créer un fichier html
 - Découper les différentes parties
 - Header
 - Nav
 - Article
 - Section
 - footer



EXERCICE 2

Statistiques			
Pays	France	Italie	Espagne
Nombre d'habitants	65 000 000	60 157 214	46 661 950
Capitale	Paris	Rome	Madrid

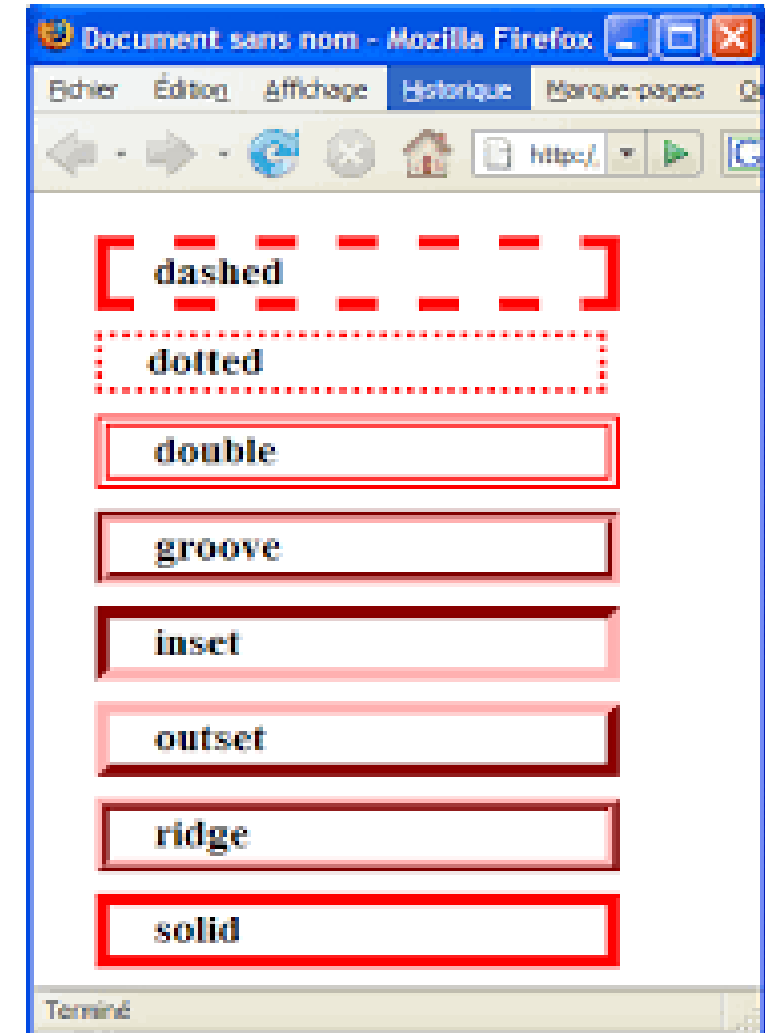
CORRECTION EXO 2

EN LIVE !

LES BORDURES



- La propriété border permet d'ajouter une bordure à un élément
- Elle a 3 « sous propriétés » possibles : **width style color**.
- On peut prendre de nombreuses valeurs différentes.
- Huit styles sont possibles (la valeur par défaut étant none) :
 - dotted (pointillé)
 - dashed (tirets)
 - solid (solide)
 - double (double)
 - groove (rainurée)
 - ridge (relief)
 - inset (relief intérieur)
 - outset (relief extérieur)



LES BORDURES ARRONDIS



- border-radius : valeur permet de faire des bords arrondis
 - Valeur : une valeur en unité de mesure (px, em ou même %)
 - Le même arrondi partout : border-radius : 10px;
 - Un arrondi par angle (en haut à gauche, en haut à droite, en bas à droite, en bas à gauche) :
 - border-radius : 2px 20px 5px 10px;

border-radius: 20px 30px



border-radius: 20px 30px 40px



border-radius: 20px 30px 40px 50px



DIMENSIONS - WIDTH ET HEIGHT



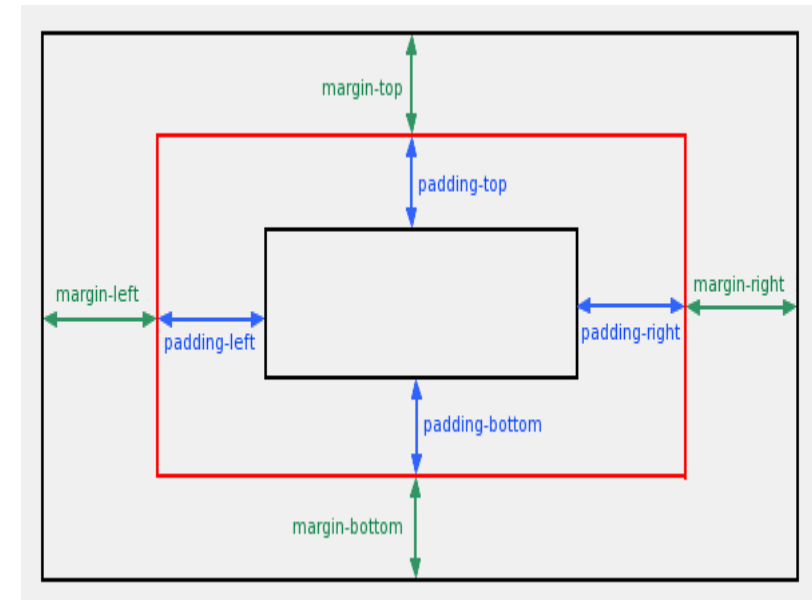
- width (largeur) et height (hauteur) ne sont applicables QUE sur des éléments de bloc
- **height** : Hauteur
 - line-height : Hauteur d'une ligne
 - max-height : Hauteur maximum
 - min-height : Hauteur minimum
- **width** : Largeur
 - max-width : Largeur maximum
 - min-width : minimum
- **Valeurs possibles:**
 - auto
 - Valeurs en px
 - Pourcentage

LES MARGES



On a deux types de marges: **padding** et **marge**

- **Padding** (rembourrage): padding pour les marges intérieures
 - La propriété padding est une propriété raccourcie qui permet de définir les différents écarts de remplissage sur les quatre côtés d'un élément. Elle synthétise **padding-top**, **padding-right**, **padding-bottom**, **padding-left**.
- **Margin** (marge): margin, pour les marges extérieures
 - La propriété margin définit la taille des marges sur les quatre côtés de l'élément. C'est une propriété raccourcie qui permet de manipuler les autres propriétés de marges : **margin-top**, **margin-right**, **margin-bottom** et **margin-left** (dans l'ordre : haut, droite, bas, gauche)



STYLE ELEMENT



MARGIN/PADDING :

margin/padding: 10px 9px 8px 7px; 10px en haut - 9px à droite - 8px en bas - 7px à gauche

margin/padding: 10px 7px; 10px en haut et en bas - 7px à gauche et à droite

margin/padding: 10px 9px 8px; 10px en haut - 9px à droite - 8px en bas.

margin/padding: 10px; 10px tout autour (haut, droite, bas, gauche).

padding-top/margin-top: 10px; top en haut

padding-right/margin-right: 10px; right à droite

padding-bottom/margin-bottom: 10px; bottom en bas

padding-left/margin-left: 10px; left à gauche

STYLE ELEMENT



Plusieurs moyens de styliser un élément :

par son id: #elem

par sa classe: .elem

par sa balise: balise



Règle d'écriture :

div.zone{} veut dire "va me chercher la div qui possède la classe 'zone' ».

div .zone{} veut dire "va me chercher la classe zone qui se trouve A L'INTERIEUR de la div (c'est l'espace qui change la signification).

.maclasse, .maclasse2{} : il y a des virgules, on applique le style sur plusieurs éléments.

CSS3 - LES UNITES



- Il existe plusieurs types d'unité, et chacun influence à sa manière le design de votre site. Nous allons faire un point sur les unités de mesure en CSS pour que vous puissiez les utiliser correctement.
 - Les unités absolues
 - Les unités relatives au texte
 - Les unités relatives au viewport
 - Les unités en pourcentage

<https://www.w3.org/Style/Examples/007/units.fr.html>

LES UNITÉS ABSOLUES



- Les unités absolues sont les suivantes :
 - Les pixels (**px**) ou Picture Element.
 - Les pouces (**in**) ou Inch
 - Les centimètres (**cm**)
 - Les millimètres (**mm**)
 - Les picas (**pc**)
 - Les points (**pt**)

Retenez donc que:

1 in = 96 px

1 cm = 37,8 px

1 mm = 3,78 px

1 pc = 16px

1 pt = 1.333.

LES UNITÉS RELATIVES AU TEXTE



À l'opposé des unités absolues, il existe des unités qui font varier la taille d'un élément en fonction de la taille de la police ou de l'élément parent.

Ces unités dites relatives au texte sont de plus en plus utilisées pour harmoniser les éléments d'un design. On retrouve :

- **Em** : appelée « cadratin » en typographie française, elle est proportionnelle à la taille de la police de l'élément parent ou du document. Par défaut, 1 em = 16 px si aucune taille de police n'est définie.
- **Rem(root em)** : l'unité rem fait toujours référence à la taille de la police de l'élément racine. En d'autres termes, elle dépend du font-size défini par défaut.
- **Ex** : très rarement utilisée, cette unité est relative à la hauteur de la police actuelle en minuscule.

LES UNITÉS EN POURCENTAGE



- Le pourcentage fait partie des unités relatives de façon générale puisqu'il s'adapte en fonction de la taille de son élément parent.
- Lorsque vous définissez une valeur en %, vous devez utiliser un nombre entier.

	Recommandé	Usage occasionnel	Non recommandé
Écran	em, px, %	ex	pt, cm, mm, in, pc
Imprimante	em, cm, mm, in, pt, pc, %	px, ex	

UNITÉS



Unités relatives - %

Exemple des pourcentage:

Coté HTML:

```
<div class="unites-relative">
  Je suis le parent
  <div class="unite-
pourcentage">
    Je suis l'enfant en
pourcentage
  </div>
</div>
```

Coté css:

```
.unites-relative{
  font-size: 20px;
}
.unite-pourcentage{
  font-size: 50%;
}
```

Unités relatives - em

Exemple d' em

Coté HTML:

```
<div class="unites-relative">
  Je suis le parent
  <div class="unite-em">
    Je suis l'enfant en em
  </div>
</div>
```

Coté css:

```
.unites-relative{
  font-size: 10em;
}
.unite-em{
  font-size: 50%;
}
```

Unités relatives - rem

Exemple de rem

Coté HTML:

```
<div class="unites-relative">
  Je suis le parent
  <div class="unite-rem">
    Je suis l'enfant en rem
  </div>
</div>
```

Coté css:

```
html{
  font-size: 20px;
}
.unite-rem{
  font-size: 1rem;
}
```

UNITÉS

- **Le pixel**

- Le pixel, comme vu en introduction de cet article, est une unité relative à la résolution et à la taille de l'écran sur lequel il est affiché.
- Cette unité considérée comme « magique » permet un affichage net et visible, même si petite, sur n'importe quel support.

- **Le pourcentage**

- Cette unité notée « % » est relative à son élément parent. Un élément avec une taille de « 50% » et qui a un élément parent avec une taille de 50px aura alors une taille de 25px.
- Le pourcentage sera à utiliser à chaque fois que vous souhaitez définir une taille en fonction de son élément parent.

UNITÉS

- **Le em**

- Le « em » correspond à la taille de la police de caractère de l'élément en cours et si cette taille n'est pas redéfinie alors cette taille correspondra à la taille de la police de l'élément parent. Par exemple, si la taille de la police est définie à 20px alors 1em sera égale à 20px.

- **Le ex**

- L'unité « ex » est une unité rarement utilisée qui permet d'exprimer des tailles relatives à l'x-height de la police. Le x-height est la hauteur des plus petites lettres (minuscules) d'une police.

- **Le rem**

- L'unité « rem » signifiant « root em » reprend le principe de l'unité « em », c'est-à-dire correspondre à la taille de la police d'un élément et cet élément est l'élément racine du document (root).

LES UNITÉS RELATIVES AU VIEWPORT



D'autres unités permettent aux éléments de s'adapter à la taille du viewport, c'est-à-dire à la taille de la fenêtre du navigateur.

Ces unités relatives au viewport sont essentielles pour mettre en place un design responsive, et on retrouve :

- La hauteur du viewport (vh)
- La largeur du viewport (vw)
- Le viewport minimum (vmin)
- Le viewport maximum (vmax)

Retenez que $1 \text{ vh} = 1 \%$ de la hauteur du viewport et que $1 \text{ vw} = 1 \%$ de la largeur du viewport

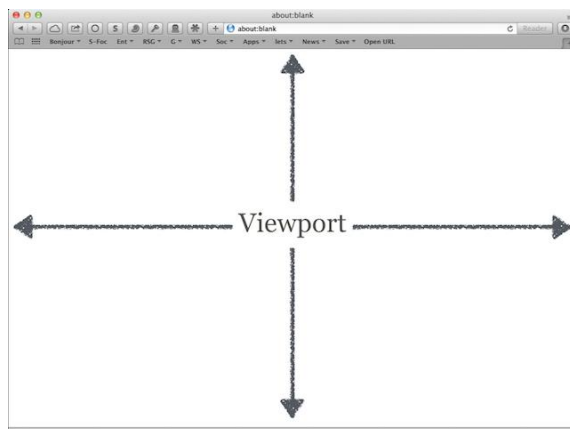
UNITÉS



Unités relatives vw

Viewport Width:

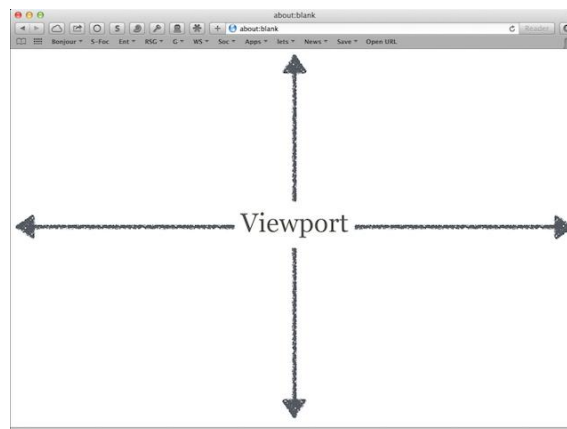
Il s'adapte à la taille de notre fenêtre de navigation



Unités relatives vh

Viewport Height:

Il s'adapte à la hauteur de notre fenêtre de navigation



Unité absolue - px

Unité qui ne change pas et n'hérite d'aucun autre élément

	<code>body { font-size: 100%; }</code>	<code>body { font-size: 120%; }</code>
<code>font-size: 1em</code>	The quick brown fox	The quick brown
<code>font-size: 12pt</code>	The quick brown fox	The quick brown fox
<code>font-size: 16px</code>	The quick brown fox	The quick brown fox
<code>font-size: 100%</code>	The quick brown fox	The quick brown

UNITÉS

- Les unités « vw » et « vh » sont similaires car elles correspondent au viewport de la fenêtre.
 - « vw » correspondra à la largeur du viewport alors que « vh » se réfèrera à la hauteur du viewport.
 - « vw » correspond ainsi à 1/100 e ou 1% de la largeur de la fenêtre et de même pour « vh » qui équivaut à 1/100 e ou 1% de la hauteur de la fenêtre.
- Grâce à ces deux unités, vous allez pouvoir définir des tailles relatives à la taille du viewport de votre support d'affichage et ainsi créer facilement des éléments responsifs ou par exemple un élément qui prendra toute la largeur de l'écran.

RESUMONS

- **PX, EM et REM** sont principalement utilisés pour le dimensionnement des polices
- **%, VW et VH** sont principalement utilisés pour les marges, le remplissage, l'espacement et les largeurs / hauteurs.
- les unités relatives sont souvent plus appréciées pour leur côté « adaptable » et les unités « rem », « vh » et « vw » sont de plus en plus utilisées pour leur simplicité et efficacité de gestion du multi-support.



LES DISPLAYS

- La propriété display définit le type d'affichage utilisée pour le rendu d'un élément (de bloc ou en ligne) et la disposition utilisée pour ses éléments fils : grille ou boîtes flexibles.
- Chaque élément d'une page web est une boîte rectangulaire. La propriété display détermine la façon dont cette boîte rectangulaire va se comporter.

DISPLAY:BLOCK



- ``, ``, ou `` sont de type inline par défaut on peut les forcer à prendre le type block.
- Pour transformer les liens inline en éléments de bloc

```
Span,b,a{  
    display: block;  
}
```

[ancre1](#) [ancre2](#) [ancre3](#) [ancre4](#)

Mon titre

Mon autre titre

[ancre1](#)
[ancre2](#)
[ancre3](#)
[ancre4](#)

Mon titre

Mon autre titre

DISPLAY:INLINE



De nombreux éléments sont réglés sur block par le navigateur lui-même. Ce sont en général des éléments conteneurs, comme <div>, <section>, , mais aussi <p> et <h1>.

```
P,h1,section {  
    display: inline;  
}
```

Confère un à un élément toutes les propriétés d'un élément en ligne, sans retour à la ligne

[ancré1](#) [ancré2](#) [ancré3](#) [ancré4](#)

Mon titre

Mon autre titre

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

test test

[ancré1](#) [ancré2](#) [ancré3](#) [ancré4](#)

Mon titre

Mon autre titre

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. test test

DISPLAY: INLINE-BLOCK



- display: inline-block sur une liste de liens permet par exemple de créer une navigation horizontale.
- La différence avec le type inline est que vous pouvez lui donner une hauteur et une largeur, qui sera respectée.

```
a {  
    display: inline-block;  
}
```

[ancré1](#) [ancré2](#) [ancré3](#) [ancré4](#)

Mon titre

Mon autre titre

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

test test

DISPLAY (MISE EN PAGE)



La propriété **display** définit le type d'affichage utilisée pour le rendu d'un élément (de bloc ou en ligne) et la disposition utilisée pour ses éléments fils : **grille** ou **boîtes flexibles**.

La propriété display est définie à l'aide de mots-clés. Ces valeurs sont rangées selon **six catégories**.

<display-outside>

Ces mots-clés définissent le type d'affichage extérieur de l'élément. Autrement dit, ils définissent comment l'élément participe au flux.

<display-inside>

Ces mots-clés définissent le type d'affichage intérieur de l'élément qui définit le contexte de formatage qui organisera le contenu de l'élément (si celui-ci n'est pas un élément remplacé).

<display-listitem>

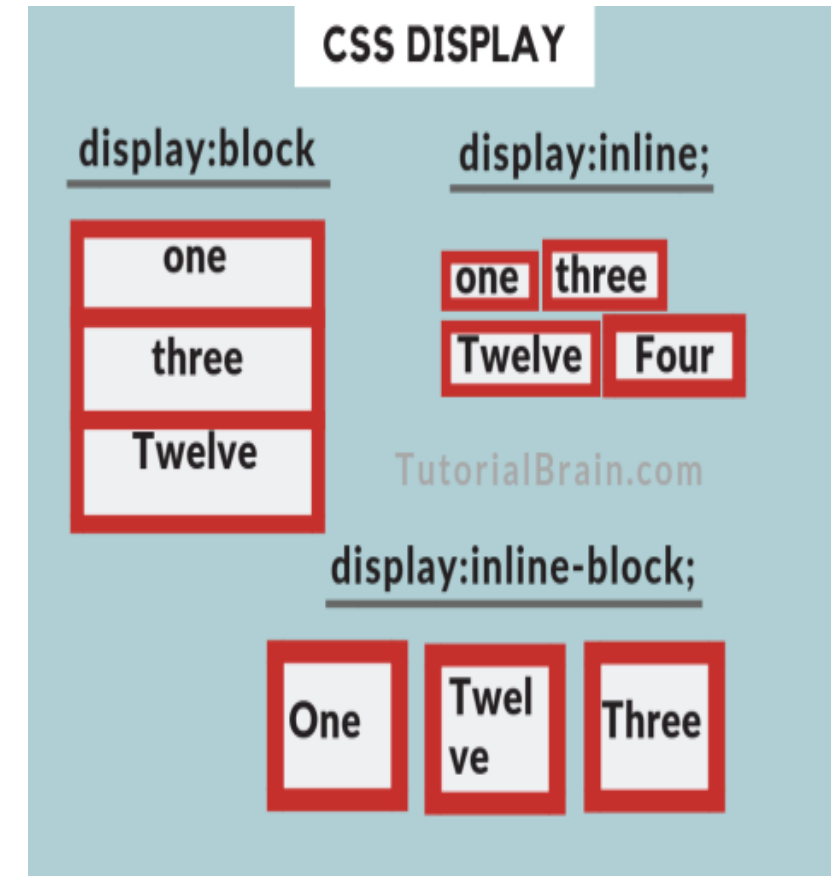
L'élément génère une boîte de bloc pour le contenu et une boîte en ligne séparée pour l'élément de liste.

<display-internal>

Certains modes de dispositions, tels que table et ruby possèdent une structure interne complexe avec différents rôles possibles pour les éléments descendants. Cette section définit ces rôles internes qui s'appliquent uniquement pour un mode de disposition donné.

<display-box>

Ces valeurs définissent si un élément génère une boîte ou non.



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

AUTRES TYPE DE DISPLAY

/ Valeurs de type
<display-outside> */*

- display: block;
- display: inline;
- display: run-in;

/ Valeurs de type
<display-inside> */*

- display: flow;
- display: flow-root;
- display: table;
- display: flex;
- display: grid;
- display: ruby;

/ Combinaison de
valeurs */*

/ <display-outside>
et <display-inside> */*

- display: block flow;

- display: inline table;
- display: flex run-in;

/ Valeurs de type
<display-listitem> */*

- display: list-item;
- display: list-item block;
- display: list-item inline;
- display: list-item flow;
- display: list-item flow-root;
- display: list-item block flow;
- display: list-item block flow-root;
- display: flow list-item block;

/ Valeurs de type
<display-internal> */*

- display: table-row-group;
- display: table-header-group;
- display: table-footer-group;
- display: table-row;
- display: table-cell;
- display: table-column-group;
- display: table-column;
- display: table-caption;
- display: ruby-base;
- display: ruby-text;
- display: ruby-base-container;
- display: ruby-text-container;

/ Valeurs de type
<display-box> */*

- display: contents;
- display: none;

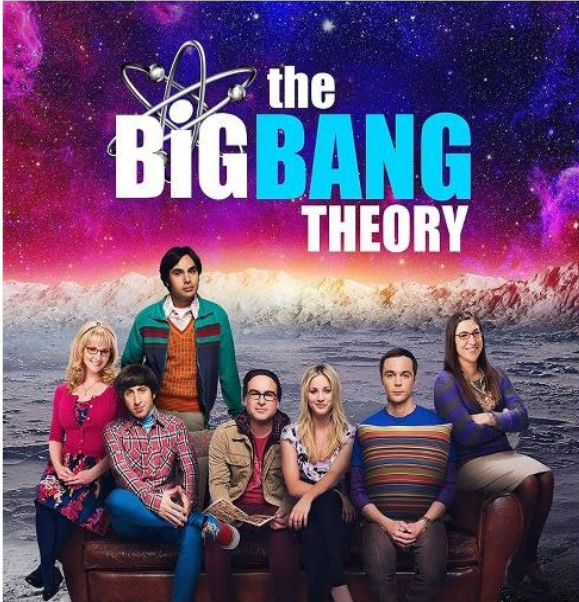
/ Valeurs de type
<display-legacy> */*

- display: inline-block;
- display: inline-table;
- display: inline-flex;
- display: inline-grid;

/ Valeurs globales */*

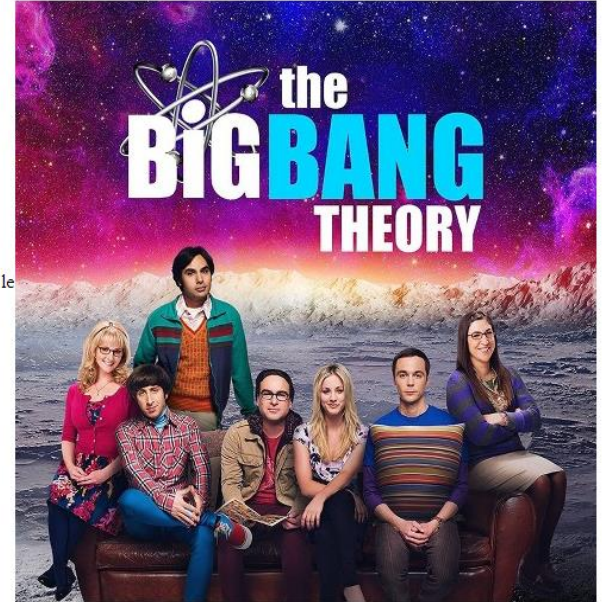
- display: inherit;
- display: initial;
- display: unset;

DISPLAY TABLE



Le Lorem Ipsum est simplement du faux texte employé dans la composition et la mise en page avant impression. Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500, quand un peintre anonyme assembla ensemble des morceaux de texte pour réaliser un livre spécimen de polices de texte.

Le Lorem Ipsum est simplement du faux texte employé dans la composition et la mise en page avant impression. Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500, quand un peintre anonyme assembla ensemble des morceaux de texte pour réaliser un livre spécimen de polices de texte.



<p>

Le Lorem Ipsum est simplement du faux texte employé dans la composition et la mise en page avant impression. Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500, quand un peintre anonyme assembla ensemble des morceaux de texte pour réaliser un livre spécimen de polices de texte.

Le Lorem Ipsum est simplement du faux texte employé dans la composition et la mise en page avant impression. Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500, quand un peintre anonyme assembla ensemble des morceaux de texte pour réaliser un livre spécimen de polices de texte.

</p>

```
p{ display:table; }
span{
  display: table-cell;
  vertical-align:middle;
}
```

LISTE DES PROPRIÉTÉS



- ✓ color
- ✓ text-decoration
- ✓ width
- ✓ background
- ✓ opacity
- ✓ text-align
- ✓ min-width
- ✓ background-color
- ✓ font
- ✓ text-indent
- ✓ max-width
- ✓ background-image
- ✓ font-face
- ✓ letter-spacing
- ✓ height
- ✓ background-repeat
- ✓ font-family
- ✓ vertical-align
- ✓ min-height
- ✓ background-attachment
- ✓ font-size
- ✓ border max-
- ✓ height
- ✓ background-position
- ✓ font-style
- ✓ border-width
- ✓ overflow
- ✓ border-collapse
- ✓ font-variant
- ✓ border-style
- ✓ visibility
- ✓ border-spacing
- ✓ font-weight
- ✓ border-color
- ✓ display
- ✓ empty-cells
- ✓ position
- ✓ border-top
- ✓ z-index
- ✓ text-shadow
- ✓ top
- ✓ border-right
- ✓ float
- ✓ box-shadow
- ✓ bottom
- ✓ border-bottom
- ✓ clear
- ✓ transition
- ✓ left
- ✓ border-left
- ✓ margin
- ✓ transform
- ✓ right
- ✓ border-radius
- ✓ Padding
- ✓ hover

<https://www.code-couleur.com/>

TP3 - NAVBAR - MENUS



- Créer un des 3 menus de navigation vers les pages:
 - Accueil
 - A propos de nous
 - Nous contacter
 - L'équipe

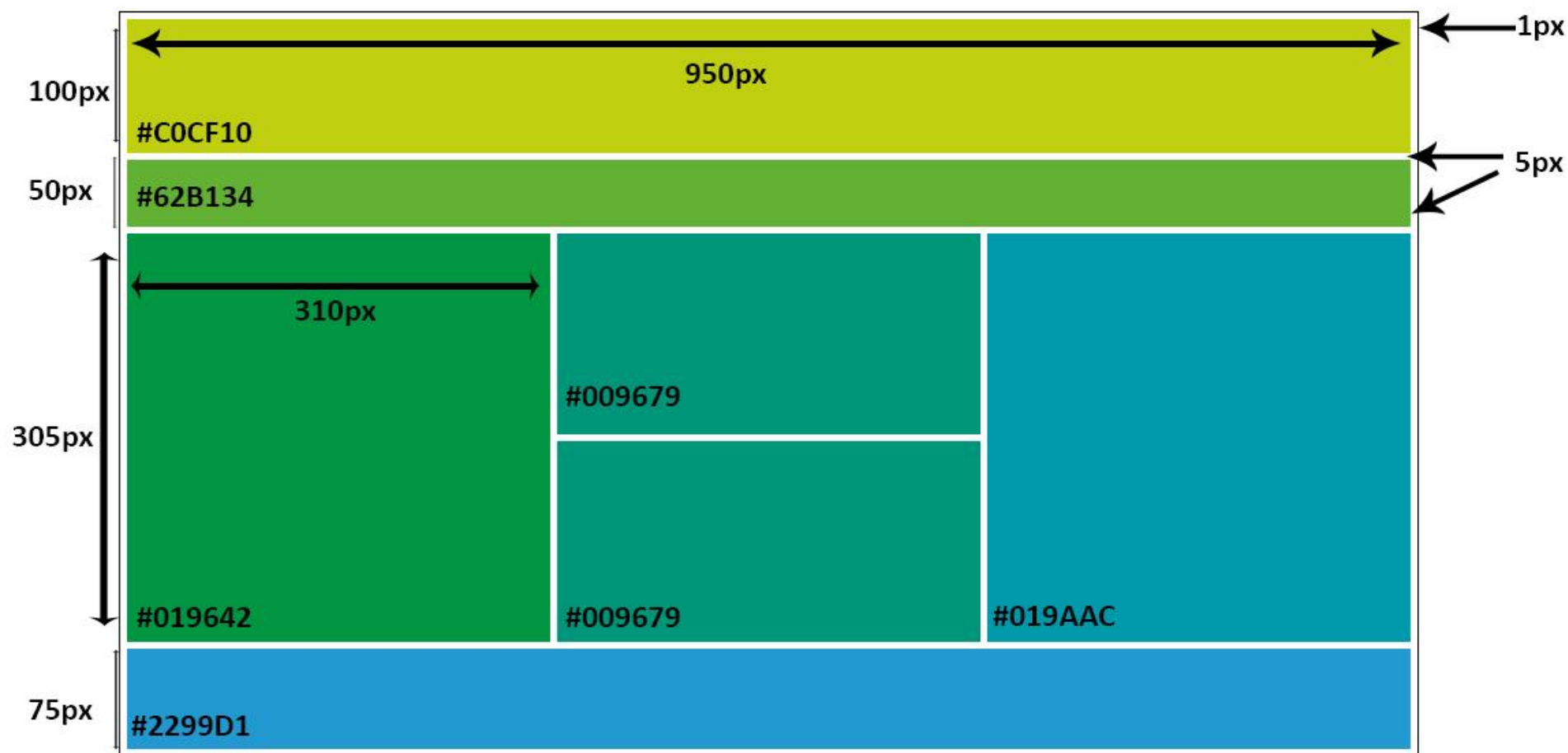
Accueil

A propos de nous

Nous contacter

L'equipe

HTML 5 - EXO 1 – ZONING 1



CORRECTION EXO – ZONING 1

EN LIVE !

BACKGROUND DEGRADE – LINEAR-GRADIENT



- Un dégradé linéaire se fait grâce au **linear-gradient**.
- Les dégradés linéaires peuvent aller du **haut vers le bas, de bas en haut, de gauche à droite, de droite à gauche en encore en diagonale**.
- Vous pouvez placer autant de couleurs que vous voulez dans votre dégradé. Il suffit d'ajouter des couleurs à la suite
- Exemple:
 - De haut en bas:
 - De bas en haut:
 - De gauche à droite:
 - `background: linear-gradient(to left, #333, #333 50%, #eee 75%, #333 75%);`
 - De droite à gauche:
 - En diagonale:

BACKGROUND DEGRADE – RADIAL-GRADIENT



Un dégradé de type radial est un dégradé créé à partir d'un point central. La valeur commence par radial-gradient (au lieu de linear-gradient).

Il y a deux formes de dégradé radiaux :

- circulaire : on utilise le mot-clé **circle** . On observe un véritable cercle dans le dégradé.
- elliptique : on utilise le mot-clé **ellipse** . On observe un cercle "aplati" si la zone du dégradé est aplatie.

```
background: radial-gradient(circle at center, red 5%, blue 5%, green 75%, yellow 10%);
```

```
background: radial-gradient(ellipse at center, red 5%, blue 5%, green 75%, yellow 10%);
```

BACKGROUND IMAGE



- On a vu comment inserer une couleur de fond maintenant on va voir comment mettre des images.
- On va commencer par créer notre fichier .css

```
body{  
    width: 100%;  
    background: url('animal.jpg');  
    background-repeat: no-repeat;  
    background-position: 500px 10px;  
}
```

- On peut aussi ajouter plusieurs image de fond

```
body{  
    width: 100%;  
    background:  
        url('animal.jpg') fixed no-repeat 200px 0,  
        url('html.jpg') fixed no-repeat top right;  
}
```


BACKGROUND IMAGE



- Background-repeat:
 - background-repeat: **no-repeat** | **repeat** | **repeat-x** | **repeat-y**
- Background-position: **left** | **right** | **center** | **top** | **bottom**
- Background-size: définit la taille des images d'arrière-plan pour l'élément
 - Valeurs possibles:
 - **Dimensions** : par exemple "px" ou "%" : précise la hauteur ou la largeur de l'image.
 - "**cover**" : forcera à couvrir toute la surface sans déformer l'image. Quitte à la rogner.
 - "**contain**" : forcera l'image à ne pas dépasser de l'élément sans la déformer.

```
body{  
  width: 100%;  
  background: url('animal.jpg');  
  background-repeat: no-repeat;  
  background-position: 50px 10px;  
  background-size: cover;  
}
```

LES FLOATS



- La propriété **float** indique qu'un élément doit être retiré du flux normal et doit être placé sur le côté droit ou sur le côté gauche de son conteneur.
- **float** implique l'utilisation d'une disposition en bloc
- La propriété CSS float peut prendre les valeurs de :
 - **none**, l'élément ne flotte pas, remet le display à sa valeur initiale et re-rentre dans le flux.
 - **left** : l'élément est considéré comme flottant, l'affichage de l'élément à droite. Son comportement est de type block et sort du flux.
 - **right** : l'élément est considéré comme flottant, l'affichage de l'élément à gauche. Son comportement est de type block et sort du flux.

CLEAR



- Comme décrit précédemment, float permet de sortir un élément du flux, on peut avoir un comportement inattendu avec certains navigateurs.
- La propriété CSS clear va nous permettre d'empêcher un élément de se positionner à côté d'un élément flottant.
- Pour éviter le mauvais placement du contenu, on utilise la propriété **clear** qui permet à un élément de cesser le contournement des éléments flottants. Il se positionne alors sous les éléments flottants précédents comme si ces derniers étaient restés dans le flux.
 - Valeurs possibles :
 - **clear: left** permet d'empêcher le contournement des blocs flottants à gauche
 - **clear: right** permet d'empêcher le contournement des blocs flottants à droite
 - **clear : both** permet d'empêcher le contournement des blocs flottants à gauche et à droite

CLEAR



```
<p class="lorem">
  Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
  tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
  quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
  consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
  cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
  proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
</p>
<p class="test">test test </p>
<p class="wf">
  Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
  tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
  quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
  consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
  cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
  proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
</p>
<p class="mouski">mouski</p>
</body>
</html>
```

```
.test{
  float: left;
  background: red;
}

.wf{
  float: right;
  width: 50%;
}

.mouski{
  clear: left;
}
```

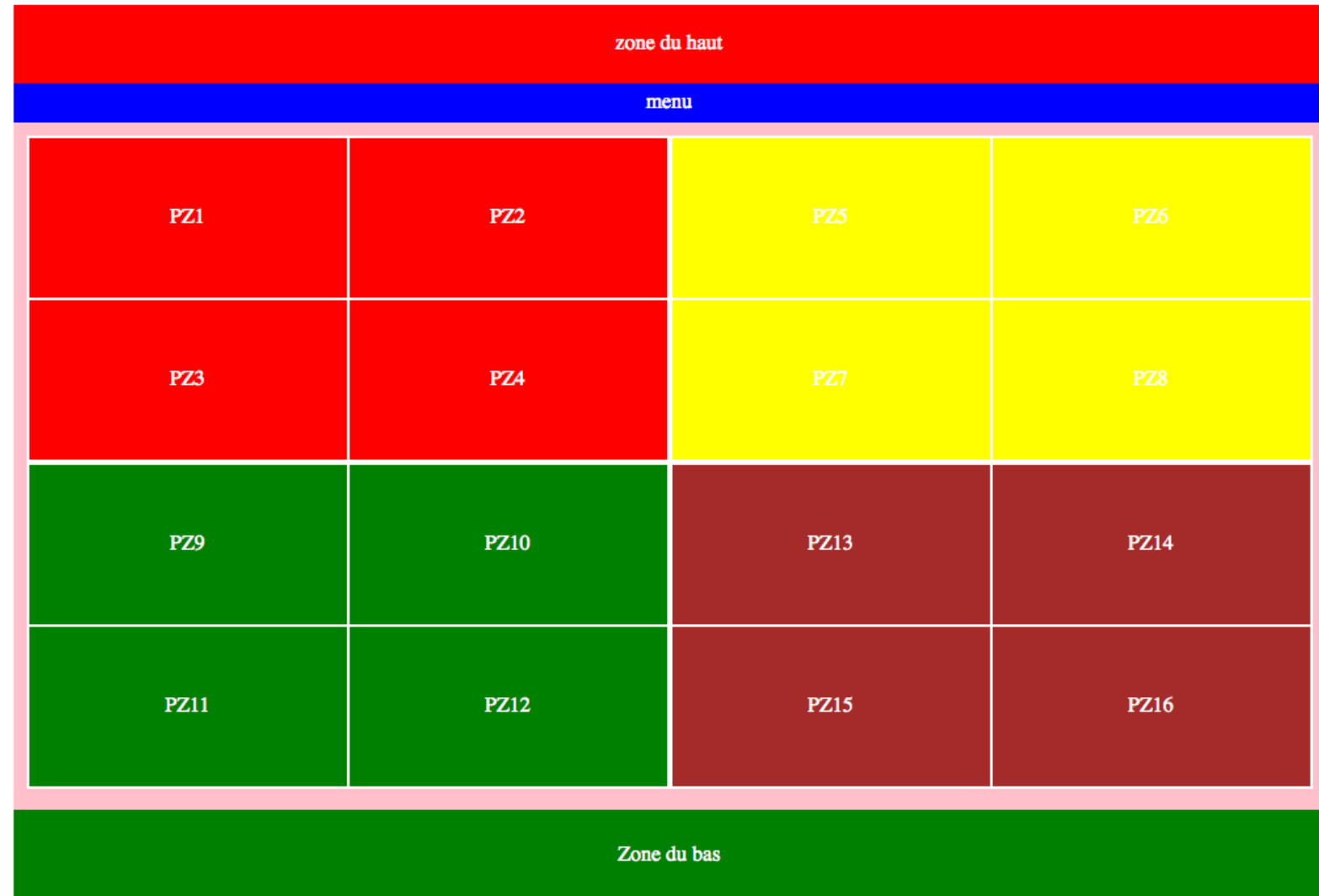
OVERFLOW



- Le souci avec les floats, on peut avoir un dépassement de contenu.
- Ajouter `overflow:hidden` au parent des flottants permet de créer un « contexte de formatage de bloc » et résoudre notre problème

```
div{  
    background: red;  
}  
img{  
    float: left;  
}  
section{  
    overflow: hidden;  
}
```

EXERCICE 4 – ZONING



CORRECTION EXO 4

EN LIVE !

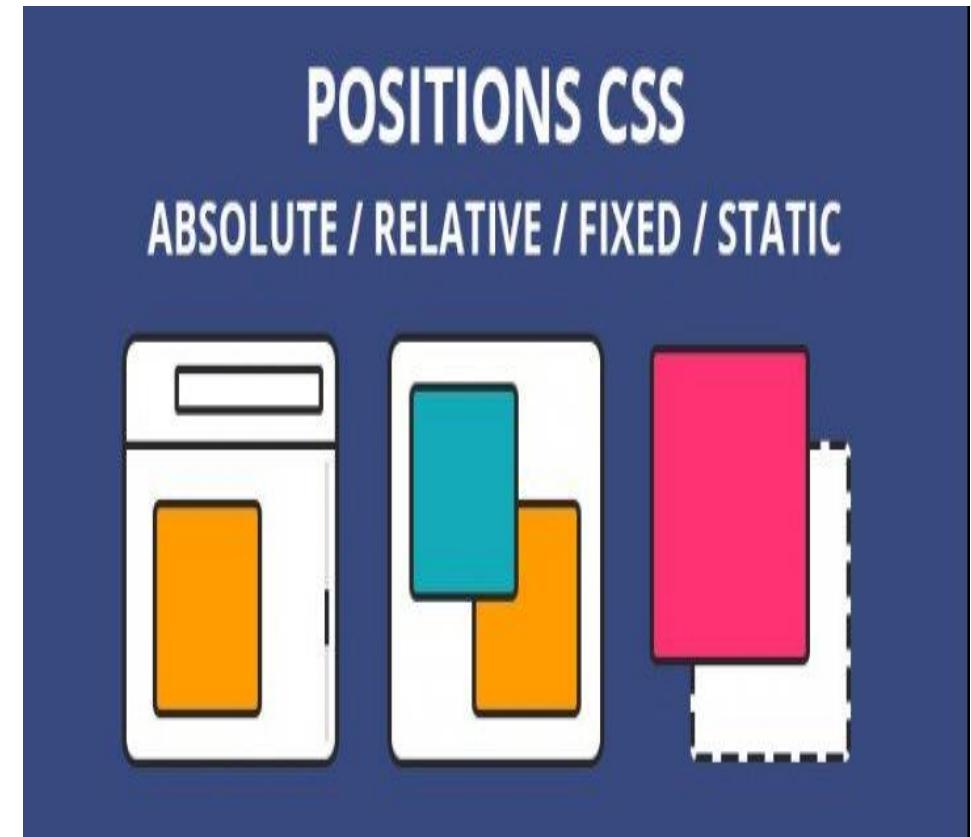


SERIE D'EXERCICES

LES POSITIONNEMENTS



- La propriété **position** définit la façon dont un élément est positionné dans un document.
- Un élément positionné est un élément dont la propriété de position calculée est **relative**, **absolute**, **fixed** ou **static**, **sticky**.
- Les propriétés **top**, **right**, **bottom** et **left** déterminent l'emplacement final de l'élément positionné.



POSITION FIXED



- Un élément en position: **fixed**; ne bougera pas, même si on fait défiler la page. Il sort du flux et se positionne au dessus des autres éléments.
- l'élément reste visible si vous descendez ou remontez le long d'une page ;

```
<a href="#ancree3">ancree3 </a>  
<a href="#ancree4">ancree4 </a>  
<h1 id="titre-page">Mon titre </h1>  
<h2 class="titre"> Mon autre titre</h2>
```

```
.titre{  
    position: fixed;  
    top: 0;  
    left: 0;  
}
```

POSITION STATIC



- static est la valeur par défaut de tous les éléments
- Un élément avec position: static; n'est positionné d'aucune manière spéciale donc ne change pas la position de base d'un élément ;

```
<a href="#ancree3">ancree3 </a>  
<a href="#ancree4">ancree4 </a>  
<h1 id="titre-page">Mon titre </h1>  
<h2 class="titre"> Mon autre titre</h2>
```

```
.titre{  
    position: static;  
    top: 0;  
    left: 0;  
}
```

POSITION RELATIVE



- **Relative** : permet de replacer un élément relativement par rapport à son positionnement par défaut ;
- On peut utiliser les propriétés **top**, **left**, **right** et **bottom** pour déplacer l'élément

```
<a href="#ancree3">ancree3 </a>  
<a href="#ancree4">ancree4 </a>  
<h1 id="titre-page">Mon titre </h1>  
<h2 class="titre"> Mon autre titre</h2>
```

```
.titre{  
    position: relative;  
    top: 50px;  
    left: 100px;  
}
```


POSITION ABSOLUTE



- **Absolute** : permet de positionner un élément n'importe où dans la page, par rapport à son élément parent direct. Il sort du flux (les autres éléments se réorganisent comme s'il n'était pas là)
- Par défaut, si aucun ancêtre n'est positionné (et qu'on a modifié top, left, right ou bottom) il se positionne par rapport à l'élément racine `<html>`.

```
<section>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
    tempor incididunt ut labore et dolore magna aliqua.
  </p>

  <p class="absolute">
    Lorem ipsum dolor sit amet, consectetur adipisicing.
  </p>
</section>
```

```
section{
  position: relative;
  background-color: #00ff00;
  padding: 5%;
}
.absolute{
  position: absolute;
  background-color: #ffff00;
}
```

POSITION STICKY

- La position STICKY est calculée en fonction du flux normal du document
- C'est le fait de pouvoir maintenir la position d'un objet à un endroit précis dès que celui-ci devient hors de portée

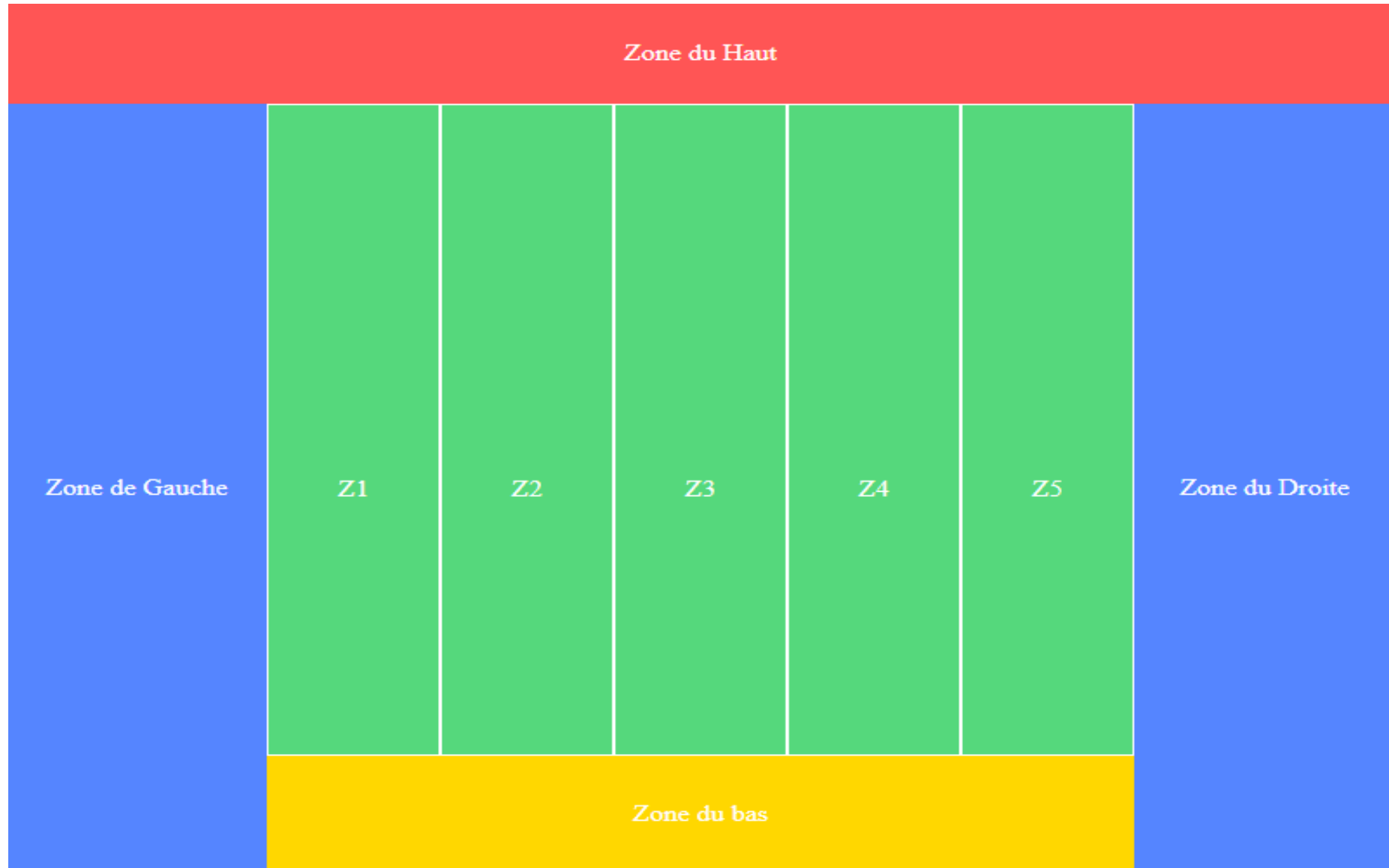
```
strong{  
  position: sticky;  
  top: 0;  
}
```

```
<div class="container">  
  <p> Duis aute irure dolor in repreh  
enderit in voluptate velit esse  
    cillum dolore eu fugiat nulla paria  
tur. Excepteur sint occaecat cupidatat non  
    proident, sunt in culpa qui officia  
deserunt mollit anim id est laborum.</p>  
  <strong>test</strong>  
  
  <p>Lorem ipsum dolor sit amet
```

RÉSUMONS

- Les positions sont pratiques dans certains cas.
- NB: NE LES UTILISER QUE POUR CES DEMANCES SPECIFIQUES (Navbar, effet de style,)
- **TOUJOURS PRIVILEGER LES FLEBOX POUR LES MISES EN FORME**

EXERCICE: REPRODUIRE LA MAQUETTE



DESIGN- LA BEAUTÉ DE LA CONCEPTION CSS



- Un peu de style ne fait pas jamais de mal ;)
- On a vu les palettes de couleur proposée, on peut aller plus loin avec des styles en degrade.
- La majorité des navigateurs suivent des standards qui ont été établis et implémentent les propriétés de la même façon mais on peut toujours des visiteurs de la vielle école et utilisent toujours la vielle technologie. Afin d'éviter quelconques soucis d'affichage, on a les **préfixes vendeurs**.

LES PREFIXES VENDEURS



Les préfixes permettent aux navigateurs d'interpréter **certaines propriétés** (plus récentes). Pour rendre vos CSS compatibles avec la plupart des navigateurs, il faudra préfixer votre code.

Les préfixes vendeurs sont les suivants :

- Pour Chrome et Safari : -webkit-
- Pour Mozilla : -moz-
- Pour Internet Explorer : -ms-
- Pour Opera : -o

```
.container
{
    margin: 50px;
    -webkit-border-radius: 5px;
    -moz-border-radius: 5px;
    -o-border-radius: 5px;
    -ms-border-radius: 5px;
}
```


LES PSEUDO-CLASSES



Une pseudo-classe est un mot clé préfixé par deux points (:) qui s'ajoute à un sélecteur CSS pour appliquer un style à un élément dans un cas particulier.

L'utilisation des pseudo-classes est très populaire sur les liens hypertextes qui changent de décor selon si ils sont nouvellement affichés, déjà visités ou survolés... Or, leur utilisation s'étend aux autres objets HTML tel que les images, les conteneurs, les listes...

QUELQUES PSEUDO CLASSES



✓:active

✓:any-link

✓:blank

✓:checked

✓:disabled

✓:drop

✓:empty

✓:enabled

✓:first-child

✓:first-of-type

✓:focus

✓:hover

✓:invalid

✓:last-child

✓:nth-child()

✓:nth-col()

✓:nth-last-child()

✓:nth-last-col()

✓:nth-last-of-type()

✓:nth-of-type()

✓:only-child

✓:only-of-type

✓:out-of-range

✓:read-only

✓:read-write

✓:required

✓:visited

EXAMPLE



- Hover

```
.container:hover{
    background: blue;
}
```
- Focus

```
input:focus{
    background:red;
}
```
- Checked

```
input:checked{
    height: 50px;
}
```

LES PARENTS

```
<article>
  <p>Je suis le premier paragraphe</p>
  <p>Je suis le deuxième paragraphe</p>
</article>
<article>
  <p>Je suis le premier paragraphe</p>
  <p>Je suis le deuxième paragraphe</p>
</article>
<article>
  <p>Je suis le premier paragraphe</p>
  <p>Je suis le deuxième paragraphe</p>
</article>
<article>
  <p>Je suis le premier paragraphe</p>
  <p>Je suis le deuxième paragraphe</p>
</article>
```

- Le premier parent:
- `article:first-of-type(2n+1){`
- `background: red;`
- `}`
- Le dernier parent:
- `article:last-of-type(2n+1){`
- `background: red;`
- `}`
- Les pairs:
- `article:nth-of-type(2n){`
- `background: red;`
- `}`
- Les impairs:
- `article:nth-of-type(2n+1){`
- `background: red;`
- `}`

UTILISATION DES PSEUDO-CLASSES :EVEN ET :ODD

- Autre option: ODD
- impair : L'utilisation d'une pseudo-classe impaire dans tout élément de liste qui n'affectera que la liste de numéros d'index impairs.

```
• p:nth-child(odd){  
•     color: blue;  
•     background-color: blue;  
•     padding: 5px;  
• }
```

- Autre option: EVEN
- even : L'utilisation d'une pseudo-classe paire dans n'importe quel élément de liste qui n'affectera que la liste de numéros d'index pair.

```
• p:nth-child(even){  
•     color: blue;  
•     background-color: blue;  
•     padding: 5px;  
• }
```

LES ENFANTS -- FIRST-CHILD

- HTML

- `<article>`
- `<p>Je suis le premier paragraphe</p>`
- `<p>Je suis le deuxième paragraphe</p>`
- `</article>`

CSS

```
p:first-child {  
    color: red;  
    background-color: black;  
    padding: 5px;  
}
```


LAST-CHILD

- HTML
- `<article>`
- `<p>Je suis le premier paragraphe</p>`
- `<p>Je suis le deuxième paragraphe</p>`
- `</article>`

CSS

```
p:last-child {  
    color: red;  
    background-color: black;  
    padding: 5px;  
}
```

CHILD

HTML

```
<article>
  <p>Je suis le premier paragraphe</p>
  <p>Je suis le deuxième paragraphe</p>
  <p>Je suis le troisième paragraphe</p>
  <p>Je suis le quatrième paragraphe</p>
</article>
```

CSS

- `p:nth-child(2n+1){`
- `color: blue;`
- `background-color: blue;`
- `padding: 5px;`
- `}`

PSEUDO-ELEMENTS



Un **pseudo-élément** est un mot-clé ajouté à un sélecteur qui permet de mettre en forme certaines parties de l'élément ciblé par la règle.

- Valeurs possibles:

- **::first-letter** ;
- **::first-line** ;
- **::before** ;
- **::after**;

::first-letter va nous permettre de sélectionner uniquement la première lettre du contenu d'éléments HTML

::first-line va nous permettre de sélectionner uniquement la première ligne du contenu d'éléments HTML

::before et **::after** vont nous permettre respectivement de cibler l'emplacement avant et après le contenu d'un élément HTML.

EXAMPLES



- `::first-letter`
`.container::first-letter{`
`color: red;`
`}`
- `::first-line`
`.container::first-line{`
`color: red;`
`}`
- `::before`
`p::before{`
`content: "Ceci est mon before";`
`}`
- `::after`
`p::after{`
`content: "Ceci est mon after";`
`}`

FIRST-LETTER

- HTML

<p>

Lorem ipsum dolor sit, amet consectetur adipiscing elit. Quærat earum, aliquid alias minima ipsa enim doloribus ratione in, nihil voluptatibus numquam sapiente architecto voluptate. Repellendus consectetur inventore ullam doloremque error!

Deleniti quam numquam beatae, amet rerum exercitationem iure qui nulla sed maxime repudiandae dicta obcaecati cum. Est non quisquam consequatur fugiat magnam nemo beatae, repellendus corporis, aliquid qui praesentium? Distinctio.

</p>

- CSS

```
p::first-letter {  
  color: red;  
  font-size: 130%;  
}
```

Lorem ipsum dolor sit, amet consectetur adipiscing elit. Quærat earum, aliquid alias minima ipsa enim doloribus ratione in, nihil voluptatibus numquam sapiente architecto voluptate.

FIRST-LINE

- HTML

<p>

Lorem ipsum dolor sit, amet consectetur adipiscing elit. Quærat earum, aliquid alias minima ipsa enim doloribus ratione in, nihil voluptatibus numquam sapiente architecto voluptate. Repellendus consectetur inventore ullam doloremque error!

Deleniti quam numquam beatae, amet rerum exercitationem iure qui nulla sed maxime repudiandae dicta obcaecati cum. Est non quisquam consequatur fugiat magnam nemo beatae, repellendus corporis, aliquid qui praesentium? Distinctio.

</p>

- CSS

```
p::first-line {  
    color: red;  
    font-size: 130%;  
}
```

Lorem ipsum dolor sit, amet consectetur adipiscing elit. Quærat earum, aliquid alias minima ipsa enim doloribus ratione in, nihil voluptatibus numquam sapiente architecto voluptate. Repellendus consectetur inventore ullam doloremque error! Deleniti quam numquam beatae, amet rerum exercitationem iure qui nulla sed maxime repudiandae dicta obcaecati cum. Est non quisquam consequatur fugiat magnam nemo beatae, repellendus corporis, aliquid qui praesentium? Distinctio. Expedita obcaecati



TP – PSEUDO-CLASSE

- Voir le pdf

LES ICONES

- Les icônes sont de plus en plus présentes sur les sites web car elles permettent de véhiculer facilement et rapidement des idées ou concepts. De nombreuses librairies d'icônes ont vu le jour suite à cette popularisation des icônes sur le web comme « Font Awesome » qui propose à l'heure actuellement plus de 1600 icônes gratuites et pas moins de 7800 icônes accessibles via leur formule payante « Standard ».
- Font Awesome propose un « kit » qui va vous permettre d'utiliser la librairie sur votre site, sans télécharger toute la librairie, en indiquant seulement une ligne « script » dans le code de votre site.

- `<link rel="stylesheet" href="font/css/all.min.css">`

QUELQUES LIENS UTILES

- <http://www.fileformat.info/info/emoji/list.htm>
- <https://fontawesome.com/v5.15/how-to-use/on-the-web/setup/hosting-font-awesome-yourself>
- [iconfinder](#)

MENU DEROULANT HORIZONTAL

- Un menu déroulant en HTML et en CSS est un peu plus complexe que de créer un menu simple à cause notamment des questions de positionnement qu'il va falloir régler.
- L'idée va donc être de créer des sous-menus. Chaque sous-menu va être lié à un onglet du menu principal et ne va devoir apparaître que lorsqu'un utilisateur passe sa souris sur l'onglet en question.



MENU DEROULANT VERTICAL

```
<nav>
  <ul class="menu-deroulant">
    <li><i class="fas fa-house-
user"></i><a href="#">Accueil</a></li>
    <li><i class="fas fa-
allergies"></i><a href="propos.html">A propos</a></li>
    <li><i class="fas fa-file-
signature"></i><a href="contact.html">Nous contacter</a></li>
    <li><i class="fas fa-american-sign-language-
interpreting"></i><a href="equipe.html">L'équipe</a></li>
    <li><i class="far fa-hand-point-right"></i>
      <a href="">Formations</a>
      <ul>
        <li><a href="">Accueil</a></li>
        <li><a href="">HTML/CSS</a></li>
        <li><a href="">JAVA</a> </li>
      </ul>
    </li>
  </ul>
</nav>
```

```
nav li {
  position: relative;
  list-style:none;
  margin:0px;
  padding: 10px;
}
nav .menu-deroulant ul {position: absolute; top: 0;}
nav ul {margin: 0; padding: 0; width: 150px; }
nav .menu-deroulant ul {left: 150px; display:none}
nav li a {text-decoration: none; color: #000; padding: 4px 0 4px;}

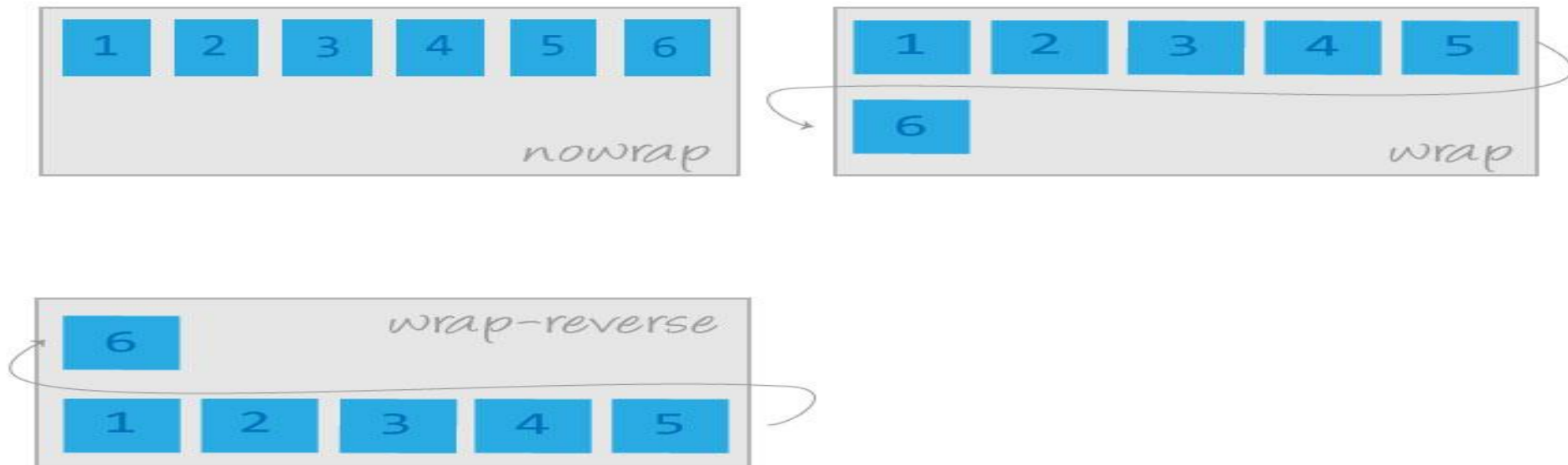
nav ul li:hover ul {
  display:block;
  background-color: lightcoral;
}
nav ul li a {
  color: white;
  text-decoration: none;
}
nav ul li:hover{
  background: #E7272F;
  border-radius: 15px;
}
```



LES FLEXBOX

- Flexbox est une méthode de mise en page selon un axe principal, permettant de disposer des éléments en ligne ou en colonne. Les éléments se dilatent ou se rétractent pour occuper l'espace disponible.
- Imaginons que nous avons un conteneur dans lequel nous voulons ranger des objets.
- Pour que tout rentre, nous devons choisir parfois des alignements différents
 - Horizontal
 - Vertical

LES FLEXBOX



Flex-Wrap