HTML / CSS

Moussa Camara
Msa.camara@gmail.com
0650146351

HTML
Content
Structual
CSS
Style
Presentational
JS
Behavioral

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 DÉJA ?

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

En-tête

LOGO

TITRE

IMAGE

PARAGRAPHE

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 (Mosaïc), ~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 : 2éme 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

```html
<!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.

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

HTML

AVANT

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.

HTML 5 - STEP 1 - STRUCTURE

- Exercice 1
  - Créer un fichier html
  - Découper les différentes parties
    - Header,Article,section et footer comme sur la page 15

step1-structure

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 (on inclut ici les langages basés sur XML comme SVG 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 avantage de css:
  - simplifies 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 ...

HTML5/CSS3

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:**

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.

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

Les ancres, des liens spéciaux:

```
<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.
  - <a href='mailto:moussa@camara.wf3'>envoie un mail </a>

Télécharger un fichier

- Pour télécharger un fichier, on a juste définir l'url dans notre href
  - <a href ='monfichier.pdf'>télécharger</a>

QUELQUES BALISES

Mettre en valeur du texte **sémantiquement**:

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

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

LISTES ORDONNÉES

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

```html
<!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>Raisin</li>
    </ol>
</body>
</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.

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

step3-listes

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:

Logo

```
<img src="images/logo.png" alt="Nuage" titre="Accueil" />
```

step4-images

QUELQUES ELEMENTS CSS 3

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>
    <img src="images/ape_3.png" alt="Singe avec un badge" />
    <figcaption>Obtenir des badges</figcaption>
</figure>
```

LES TABLEAUX

icon	text	text	full text	icon	full text
	:)	:-)	:smile:		:lol:
	:D	:-D	:grin:		:oops:
	:(	:-(	:sad:		:cry:
	:o	:-o	:eek:		:evil:
	8O	8-O	:shock:		:twisted:
	:?	:-?	:???:		:roll:
	8)	8-)	:cool:		:!:
	:x	:-x	:mad:		:?:
	:P	:-P	:razz:		:idea:
	:|	:-|	:neutral:		:arrow:
	;)	;-)	:wink:		:mrgreen:

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>
            <td>Titre</th>
            <td>Création</th>
            <td>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
	:)	:-)	:smile:		:lol:
	:D	:-D	:grin:		:oops:
	:(	:-(	:sad:		:cry:
	:o	:-o	:eek:		:evil:
	8O	8-O	:shock:		:twisted:
	:?	:-?	:??:		:roll:
	8)	8-)	:cool:		:!:
	:x	:-x	:mad:		:?:
	:P	:-P	:razz:		:idea:
	:|	:-|	:neutral:		:arrow:
	;)	;-)	:wink:		:mrgreen:

```
<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

- 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>Arséne</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;
}
```

step5-tables

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

```html
<!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>


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

</body>
</html>
```

L'audio:

Son

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

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

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

```html
<!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:

Vidéo Poster

```html
<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>
```

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

TEST DE VALIDITER 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 – EMPLACEMENTS POSSIBLES

Dans le code HTML	`<p style="..." ></p>`

Dans la balise `<head>`	`<head>`
	`  <style> … </style>`
	`</head>`

Dans une feuille de style	`<head>`
	`  <link rel="stylesheet"`
	`  href="style.css" />`
	`</head>`

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.
- Sélecteur de type
  - Lorsqu'on utilise *h2* pour changer le style de tous les titres de niveau 2, on utilise un sélecteur de type
- Sélecteurs d'attributs
  - Les deux attributs les plus couramment utilisés sont **class** et **id**

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:

	Balise générique de type "bloc"
	Balise générique de type "inline"

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).

Eléments block	Eléments 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;
```

Style de police

```
font-weight: normal, bold;
font-style: italic,
oblique;

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 (hérite 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")

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;

TP 2

- Exercice:
  - 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:
      1. J'aime le Html, Webforce, 25/03/2020, Ceci est le résumé de mon article1
      2. J'aime aussi le CSS, Moussa Camara, 01/04/2020, Ceci est mle résumé de l'article2
      3. HTML/CSS, Tim Berner, 01/01/1996, Language hyperText

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 maxium
  - min-height : Hauteur minimum
- width : Largeur
  - max-width : Largeur maxium
  - min-width : minimum

- Valeurs possibles:
  - auto
  - Valeurs en px
  - Pourcentage

LES MARGES

On a deux types de margins: 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)

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**)
  - Les pouces (**in**)
  - Les centimètres (**cm**)
  - Les millimètres (**mm**)
  - Les picas (**pc**)
  - Les points (**pt**)

Retenez donc que 1 in = 96 px, 1 cm = 37,8 px et 1 mm = 3,78 px.

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 : 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 définit par défaut.
- Ex : très rarement utilisée, cette unité est relative à la hauteur de la police actuelle en minuscule.
- Ch : cette unité est elle aussi peu utilisée, elle est relative à la largeur du caractère "0".

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 vh = 1 % de la hauteur du viewport et que 1 vw = 1 % de la largeur du viewport

LES UNITÉS EN POURCENTAGE

- Le pourcentage fait partie des unités relatives de façon générale puisqu'il s'adapte en fonction de son élément parent.

- Lorsque vous définissez une valeur en %, vous devez utiliser un nombre entier.

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

Unités relatives vw

Viewport Width:

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

Unités relatives vh

Viewport Width:

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

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

© KyleSchaeffer.com

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 <b> 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 , section>, <ul>, 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

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;
}
```

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.

<display-legacy>

CSS 2 utilise une syntaxe avec un seul mot-clé pour la propriété display et il faut donc des mots-clés distincts pour les variantes bloc/en ligne d'un même mode de disposition.

https://developer.mozilla.org/fr/docs/Web/CSS/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;

TP3 - NAVBAR - MENUS

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

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 nagivateurs.

- 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

```html
<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>
```

```css
.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;
}
```

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.
- 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="#ancre3">ancre3 </a>
<a href="#ancre4">ancre4 </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="#ancre3">ancre3 </a>
<a href="#ancre4">ancre4 </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 propriété top, left, right et bottom pour déplacer l'élément

```
<a href="#ancre3">ancre3 </a>
<a href="#ancre4">ancre4 </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;
}
```

TP

- Aprés le header, le contenu, nous allons maintenant nous attaquer au footer.

- Dans le footer, notre client voudrait:
  - L'adresse postale de son établissement
  - Liens vers les reseaux sociaux

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/

LES FORMULAIRES

Les formulaires HTML sont un des vecteurs principaux d'interaction entre un utilisateur et un site web ou une application. Ils permettent à l'utilisateur d'envoyer des données au site web.

L’attribut action précise l’adresse de la page qui va traiter les données.

Forme simplifiée:

```
<form method="post" action="traitement.php">
    <label>Votre pseudo</label> : <input type="text" />
</form>
```

method:

- ★ post
- ★ get

LES CHAMPS DE FORMULAIRE

input est un élément qui va permettre à l'utilisateur d'envoyer des données.

type va déterminer le type de données que l'utilisateur va pouvoir envoyer

Forme complète:

```
<fieldset>
    <legend>Mon 1er formulaire</legend>
    <form method="post" action="traitement.php">
        <label for="nom">Votre nom :</label>
        <input type="text" name="nom" id="nom" placeholder="
        Votre nom" size="50" maxlength="5" autofocus required/>
    </form>
</fieldset>
```

LES CHAMPS DE FORMULAIRE

Champs monolignes:

```
<input type="text" name="nom"/>
<input type="password" name="mdp"/>
<input type="email" name="couriel"/>
<input type="date" name="naissance"/>
<input type="number" name="nb_article"/>
<input type="range" name="nb_article"/>
<input type="search" name="recherche"/>
<input type="url" name="blog"/>
<input type="color" name="couleur_article"/>
```

CHAMPS MULTILIGNES

Champs multilignes:

```
<textarea name="contact" id="contact" rows="10" cols="50">
    Vous pouvez me contacter ici.
</textarea>
```

QUELQUES ELEMENTS CSS 3

Champs d'options:

```
<form>
    <p>Language que vous connaissez :</p>
    <input type="checkbox"/>HTLM
    <input type="checkbox"/>CSS
    <input type="checkbox"/>JavaScript
    <input type="checkbox" checked/>PHP
</form>
```

QUELQUES ELEMENTS

Champs d'options multiples:

```
<form method="post" action="traitement.php">
    <input type="checkbox" name="html" id="html" />
    <label for="html">HTML</label><br />
    <input type="checkbox" name="js" id="js" />
    <label for="js">JavaScript</label><br />
    <input type="checkbox" name="php" id="php" />
    <label for="php">PHP</label>
</form>
```

QUELQUES ELEMENTS

Champs d'option unique:

```
<form method="post" action="traitement.php">
    <p>Language que vous préférez :</p>
    <input type="radio"/>HTML
    <input type="radio"/>CSS
    <input type="radio"/>JavaScript
    <input type="radio" checked/>PHP
</form>
```

QUELQUES ELEMENTS

Bouton de soumission du formulaire:

```
<input type="submit" value="Soumettre" />
<input type="image" value="Soumettre" />
<input type="button" value="Soumettre" />
<input type="reset" value="Réinitialiser" />
```

TP

- Créer un formulaire de contact avec les champs:
  - Nom
  - Prénom
  - Pseudo
  - Mot de passe
  - Date de naissance
  - Adresse
    - Rue
    - Ville
    - Code postal
  - cv

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;
}
```

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 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
- ✓:first-child
- ✓:first-of-type
- ✓:fullscreen
- ✓:future
- ✓:focus
- ✓:hover
- ✓:indeterminate
- ✓:in-range
- ✓:invalid
- ✓:last-child:not()
- ✓:nth-child()
- ✓:nth-col()
- ✓:nth-last-child()
- ✓:nth-last-col()
- ✓:nth-last-of-type()
- ✓:nth-of-type()
- ✓:only-child
- ✓:only-of-type
- ✓:optional
- ✓:out-of-range
- ✓:read-only
- ✓:read-write
- ✓:required
- ✓:target
- ✓:visited

EXEMPLE

- Hover

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

- Focus

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

- Checked

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

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.

ANIMATIONS

- On va mettre un peu d’animation, le but étant de donner l’envie de cliquer aux utilisateurs.

- On va faire un slide sur les articles sur notre application

EXEMPLES

- ::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";
}
```

TP-ANIMATION - SLIDE

- Créer un fichier slide.html
- Ensuite faire un style.css

```
<section>
	<ul>
		<li><img src="images/famille.jpg"></li>

		<li><img src="images/homer.jpg"></li>

		<li><img src="images/simpson.jpg"></li>

		<li><img src="images/h-simson.jpg"></li>

		<li><img src="images/famille2.jpg"></li>
	</ul>
</section>
```

```
img{
	height:188px;width:250px;
}
section {
	width: 250px; /* taille de 3 images + padding 7 */
   overflow: hidden; /* pas de scrollbar horizontal */
}
section ul {
   animation: 20s normal none infinite slide;
   display: table;
   width: calc(250px * 5);}
section ul li {
   display: table-cell; /* simule les cellules des tables  */
}

@keyframes slide {
0%, 20% { transform: translateX(0px); }
20% { transform: translateX(-250px); }
40% { transform: translateX(-500px); }
60%{ transform: translateX(-750px); }
80% { transform: translateX(-1000px); }
100% { transform: translateX(0px); }
}
```

TP

- Faire un tableau pour afficher les offres du moment

	Pack 1	Pack 2	Pack 3
	199 € / mois	139 € / mois	59 € / mois
Projets	illimité	50	10
Utilisateurs	125	90	20
Stockage	illimité	illimité	2 GO
Base de données	oui	oui	oui
Gestion des droits	oui	non	non

TP – REFONTE DU SITE DE WF3

- Bonne nouvelle, notre réputation nous precede, la responsable a parlé de nous au PDG du boncoin Antoine Jouteau qui nous a sollicité pour faire un site vitrine pour vendre des voitures d'occasion.
- Le but etant de faire une page vitrine avec les voitures du moment, afficher la marque, le kilometrage et le prix.
- Avant d'acheter ou vendre son véhicule, on doit se connecter ou s'inscrire si on n'a pas de compte.

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
  - Horigontal
  - Vertical

LES FLEXBOX

Flex-Wrap

LES FLEXBOX

```
<div id="flexbox">
  <div>     test 1</div>
  <div>     test 2</div>
  <div>     test 3</div>
  <div>     test 4</div>
</div>
```

```
#flexbox{
  background-color:                red;
  width:200px;
  display: flex;
  flex-wrap: wrap;
  justify-content: space-around;

}

#flexbox div{
  width: 75px;
   border:1px solid;
   height: 80px;
}
```

LES FLEXBOX-TP

Seamlessly visualize quality

Collaboratively administrate empowered markets via plug-and-play networks.

Here's why

Completely Synergize

Dramatically engage seamlessly visualize quality intellectual capital without superior collaboration and idea-sharing.

Here's how

Dynamically Procrastinate

Completely synergize resource taxing relationships via premier niche markets.

Read more

RWD - RESPONSIVE WEBDESIGN

- Suite à notre premier livraison, le comité de pilotage nous a fait leur premier retour:
  - Consultation des voitures sur mobile, tablette et desktop

RWD - RESPONSIVE WEBDESIGN

- **Les requêtes média (*media queries*)** permettent de modifier l'apparence d'un site ou d'une application en fonction du type d'appareil (impression ou écran par exemple) et de ses caractéristiques (la résolution d'écran ou la largeur de la zone d'affichage (*viewport*) par exemple).

- Les requêtes média sont utilisées afin :
  - **D'appliquer** certains styles de façon conditionnelle grâce aux règles @ @media et @import.
  - **De cibler** certains média pour les éléments <style>, <link>, <source> et d'autres éléments HTML grâce à l'attribut media=

RWD - RESPONSIVE WEBDESIGN

- Valeurs possibles:
  - ✓ screen
    - ✓ Écrans
  - ✓ handheld
    - ✓ Périphériques mobiles ou de petite taille
  - ✓ print
    - ✓ Impression
  - ✓ projection
    - ✓ Projecteurs (ou présentations avec slides)
  - ✓ tty
    - ✓ Terminal/police à pas fixe
  - ✓ tv
    - ✓ Téléviseur
  - ✓ all
    - ✓ Tous les précédents

RWD - RESPONSIVE WEBDESIGN

- Exemples suivants ciblent les écrans de largeur inférieure à 640 pixels grâce à la règle max-width associée à la valeur 640px.

```
<!DOCTYPE html>
<html>
<head>
    <title>Revision html</title>
    <link rel="stylesheet" type="text/css" media="screen and (max-width:500px)" href="css/design.css">
    <meta charset="utf-8">
</head>
<body>
```

```
@media screen and (min-device-width: 200px) and (max-device-width: 640px) {
    Je mets mon code ici
    }
```

RWD - RESPONSIVE WEBDESIGN

Avant d'aller plus loin, on va voir les différentes tailles de device:

- mobile
- tablette
- Desktop

https://media-queries.aliasdmc.fr/media-queries-live.php

RWD - DESKTOP

- <u>Ecran d'ordinateur (que l'on appelle « Desktop »)</u>
  - On distingue 2 sortes d'écran différents: les écrans « normaux » (15, 17 pouces) et les grands écrans (21 pouce et +).
  - Il faut la taille de la fenêtre et non de l'écran.

    - 1024px
    - 1600px

RWD - TABLETTE

- <u>Tablette</u>
  - Pour les tablettes on a les tailles suivantes (généralement):
    - 1024px de largeur (format paysage)
    - 740px de largeur (format portrait)

RWD - MOBILE

- Plus compliqué à gérer depuis l'arrivée des smartphones. On peut retrouver les plus utilisés:
  - 480px (format portrait et paysage)
  - 320px (format portrait)

CORRECTION TP

- Correction TP:
  - Adapter notre application à toutes les devices
    - Mobile first
    - Tablette
    - Desktop

DES QUESTIONS ??

FIN

Ask

Google