

HTML5 / CSS3



Site internet





- Introduction à cette formation

- Votre formateur ...

Et Vous



- Le matériel

- L'éditeur de texte Brackets orienté développement Web. Navigateurs Chrome et Firefox

- L'organisation – horaires

- Formation de 4.5 jours

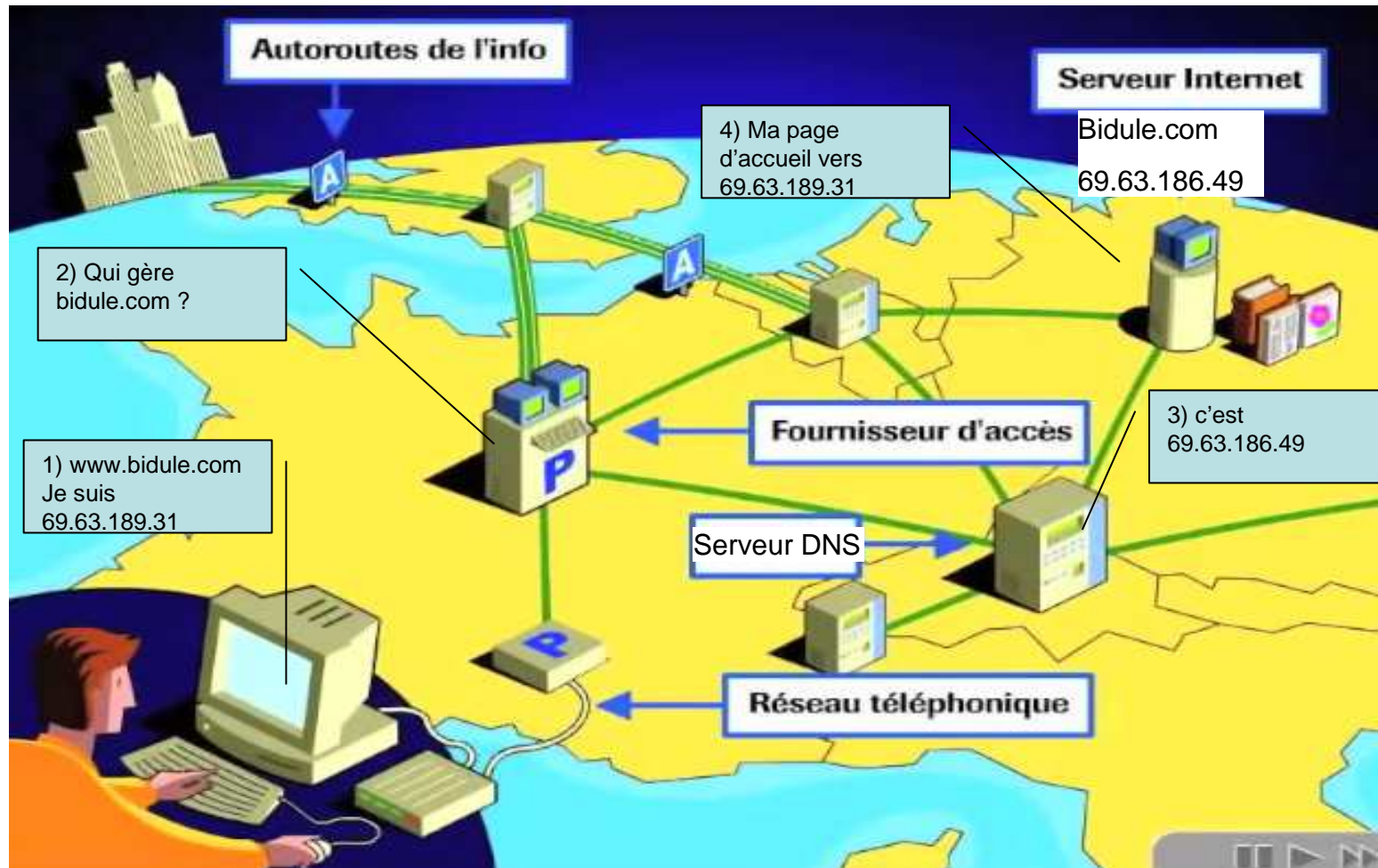
- La forme :

- Un mélange de concepts avec application directe par un exemple simple
 - Des exercices

- Les liens utiles

- <http://www.w3schools.com/>
- <https://www.w3.org/TR/html5/>
- <https://www.w3.org/TR/css-syntax-3/>
- <https://www.futura-sciences.com/tech/videos/kezako-fonctionne-internet-5293/>
- <https://openclassrooms.com/courses/apprenez-a-creer-votre-site-web-avec-html5-et-css3>

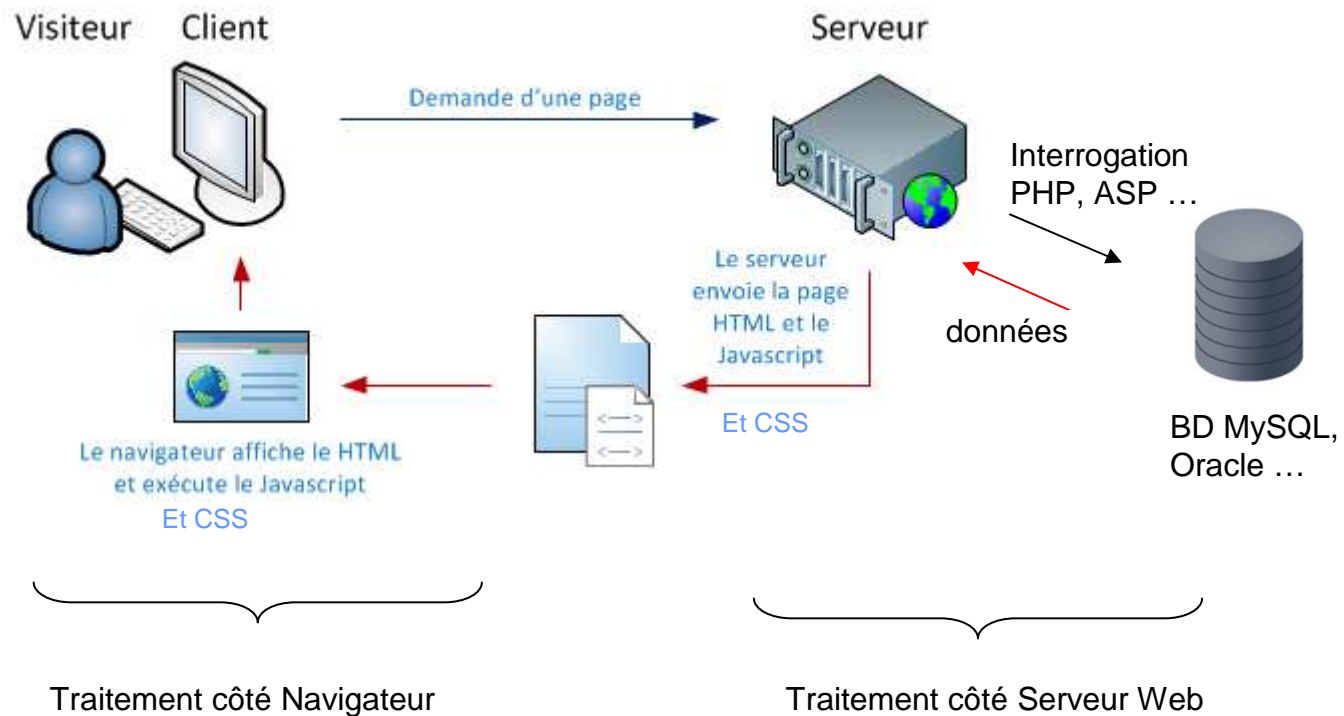
- Généralités sur le web
- Notre environnement de travail
- HTML
- CSS



- La communication client/serveur (navigateur/serveur du site Web) se fait en TCP/IP
- Protocoles utilisés
 - http, HyperText Transfert Protocol
 - https, crypté
 - ftp pour télécharger de gros fichiers
 - Imap, smtp pour les e-mails
 - ...
- Pour l'affichage d'une page web, deux acteurs principaux
 - Le serveur Web qui fournit les données
 - Le navigateur qui sait les interpréter et afficher



- Traitements serveur et client



- Nous allons séparer notre approche en 2 étapes :
 - Côté navigateur, apprendre le HTML, CSS : utilisation de l'éditeur de texte et simulateur, application Brackets



<http://www.01net.com/telecharger/windows/Programmation/javascript/fiches/125215.html>

<http://www.01net.com/telecharger/mac/Programmation/fiches/125218.html> pour Apple



Brackets utilise le navigateur Chrome pour visualiser les code. Le télécharger si non présent sur le poste.

Télécharger aussi FireFox pour disposer d'un second navigateur.

<https://www.mozilla.org/en-US/firefox/all/#fr>

- Si nous souhaitons simuler notre site en production, il faut un serveur Web. L' environnement XAMPP contient un serveur Web Apache/Tomcat, un interpréteur php, un gestionnaire de base MySQL :

<https://www.apachefriends.org/fr/download.html>

- Si erreur **XAMPP - Port 80 in use by "Unable to open process" with PID 4** dans fenêtre cmd 'Net stop was /y' puis désactiver service WWW-Publishing et Web Deployment Agent Service

- HTML (Hyper text Markup Language) 1991
Autrefois utilisé pour le contenu, la structure et la présentation.
Le HTML (ou XHTML plus rigoureux) est aujourd'hui destiné à représenter le **contenu** et la **structure des pages** : titres, sous-titres, paragraphes, images, textes ...
- CSS (Cascading Style Sheets) 1995
Agit sur la **présentation** des éléments HTML : couleur, taille, police, position dans la page ...
- Javascript 1995
Langage qui est reconnu et interprété par les navigateurs.
Permet de créer des réactions locales à des événements opérateur, réaliser des vérifications locales, demander des mises à jour au serveur ... (non abordé dans ce cours)

- AJAX 1995
Asynchronous JavaScript and XML
Code Javascript qui permet des échanges entre le serveur et client
- JQuery 2006
Bibliothèque en Javascript qui intègre des commandes AJAX

- Charger une page quelconque dans le navigateur (Firefox ou Chrome) puis examiner le code source de la page. On y voit facilement des séquences :
 - De html
 - De référence à des fichiers css, séquence `type="text/css"`
 - De référence à des fichiers javascript `type="text/javascript"`



- Le HTML est un langage à balises qui permet de structurer les pages Web

- Balise :

`<bal> ... </bal>`

balise d'ouverture et fermeture

`<bal/>`

ouverture et fermeture (balise orpheline)

- Balise avec attribut :

`<bal attr="valeur"> ... </bal>`

`<bal attr="valeur" />`

ex :

`<div id="piedPage"> ... </div>`

`<body class="xpInfos" id="xp-infos3">`

- Structure d'une page

```
<!DOCTYPE html>
```

Indique la nature du document
attr lang et dir existent

```
<html>
```

```
  <head>
```

```
    <title>Titre de la page</title>
```

```
    <link rel="stylesheet" type="text/css" href="styleWeb1.css" media="screen"/>
```

```
    <link rel="script" type="text/javascript" href="code.js"/>
```

```
    <script type="text/javascript">
```

```
      <!-- Scripts JavaScript -->
```

```
    </script>
```

```
    <style type="text/css">
```

```
      <!-- Styles CSS -->
```

```
    </style>
```

```
    <meta name="Keywords" content="HTML 5, CSS 3, Web" />
```

```
  </head>
```

```
  <body>
```

```
    <!-- le corps de la page -->
```

```
  </body>
```

```
</html>
```

- Structure HTML

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Titre de la page</title>
5  </head>
6  <body>
7    <!-- ceci est un commentaire -->
8    <h1>Corps de la page</h1>
9    Texte de la page
10 </body>
11 </html>
12
```



- Exercice 1
Prise en main de Brackets



- Organiser le texte

- `<p>` créer un paragraphe
- `<h1>` à `<h6>` créer un niveau de titre (taille couleur .. avec CSS)
- `` gras (bold)
- `` élément important (traduit en gras)
- `<i>` italique
- `<mark>` surligné
- `
` saut ligne
- `<hr />` ligne de séparation
- `<q>` citations (mise entre quotes)

Explorer ces items sur le site

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



- Les listes :
 - Listes ordonnées
 - `` `` la liste
 - `` `` un élément de la liste

```
<ol>
  <li> élément 1</li>
  <li> élément 2</li>
  <li> élément 3</li>
  <li> élément 4</li>
</ol>
```

1. élément 1
2. élément 2
3. élément 3
4. élément 4

- Listes non ordonnée
 - `` `` la liste
 - `` `` un élément de la liste

```
<ul>
  <li> élément 1</li>
  <li> élément 2</li>
  <li> élément 3</li>
  <li> élément 4</li>
</ul>
```

- élément 1
- élément 2
- élément 3
- élément 4


– Liste de définition : remplace la puce par un terme dt

- `<dl> </dl>` crée la liste de définition
- `<dt> </dt>` le terme d'un élément de la liste
- `<dd> </dd>` la définition du terme

```
<dl>
  <dt>Terme 1</dt>
  <dd>Définition du terme 1</dd>
  <dt>Terme 2</dt>
  <dd>Définition du terme 2</dd>
  <dt>
  <dd>Définition du terme 3</dd>
</dl>
```

Terme 1
Définition du terme 1

Terme 2
Définition du terme 2



Définition du terme 3

– Listes imbriquées

```
<!-- Liste imbriquée -->
<ol>
  <li>Elément 1</li>
  <li>
    <ul>
      <li>Elément 2.1</li>
      <li>Elément 2.2</li>
    </ul>
  </li>
</ol>
```

1. Elément 1
2.
 - Elément 2.1
 - Elément 2.2

- Les liens.

Essentiels, ils permettent la navigation sur le Web :

`Texte sélectionnable`

- Liens d'un site vers un autre

`Texte sélectionnable`

- Liens entre pages du même site

`Texte sélectionnable`



URL relative

- Créer les répertoires c:\Web\pages et c:\Web\images
- Créer le fichier c:\Web\pages\page1.html
- Editer c:\Web\pages\page1.html avec Brackets



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Titre de la page</title>
5   </head>
6   <body>
7     <!-- ceci est un commentaire -->
8     <h1>Corps de la page page1</h1>
9     <p>
10       <a href="../index.html">Aller à index</a>
11     </p>
12
13   </body>
14 </html>
```



- Dans c:\Web\index.html, ajouter

```
<p>
  <a href="pages/page1.html">Aller à la page 1</a>
</p>
```

- Liens vers d'autres parties de la même page

2 étapes :

- Poser des balises avec l'attribut `id= " balise "`
- Utiliser ces balises dans le tag ``

- Dans index.html ajouter les séquences :

en début

```
<a href="#lienPage1">Vers lien page 1</a>
```

en fin

```
<p id="lienPage1">  
  <a href="pages/page1.html">Aller à la page 1</a>  
</p>  
<a href="#">Top</a>
```

- Le cas `` crée un lien en début de page, sans poser de balise

- Liens vers d'autres parties d'autres pages
dans c:\Web\pages\page1.html ajouter

```
<p>  
  <a href="../../../index.html#lienPage1">Vers fin de page 1</a>  
</p>
```



- Liens avec ouverture d'une autre fenêtre
dans le tag <a> ajouter l'attribut target= " _blank "
Modifier page1.html

```
<p>  
  <a href="../../../index.html#lienPage1" target="_blank">Vers fin de page 1</a>  
</p>
```

- Liens pour l'envoi de mail
`texte sélectionnable `

Modifier page1.html

```
<p>  
  <a href="mailto:albert.einstein@e.mc2">Ecrire à Albert</a>  
</p>
```

- Lien mail plus complet
`
 Texte_ou_image_à_cliquer`

- Choisir leur format

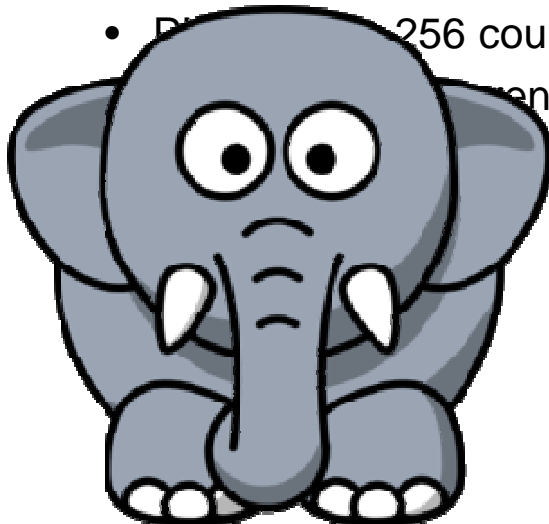
- Le JPEG

- Extension .jpg et .jpeg
 - Efficace pour les photos, optimise le poids (taille en Ko)
 - 16 millions de couleurs



- Le PNG

- Format récent
 - Concurrent gratuit de GIF, quand le GIF était payant
 - Efficace pour ce qui n'est pas photo
 - PNG 256 couleurs, PNG 24 bits : 16 millions de couleurs



– Le GIF

- Format ancien encore très utilisé
- Préferer le PNG : plus léger et transparence
- GIF limité à 256 couleurs
- Notion de transparence
- GIF animé



– Le SVG

- Fait partie du HTML5, images vectorielles
Regarder des exemples dans un navigateur



– Les autres formats

- À oublier

- Choisir leur taille et leur poids
Les images utilisées sur le site Web doivent être enregistrées avec les dimensions qu'elles auront sur le site.
Avoir des images trop grandes nuit au chargement de la page
=> il faut retailler les images
ex : site <http://resizeimage.net/>

La résolution conseillée est 72 ppp (pixels par pouce)

- Les images :
 - Sont soumises à droit d'auteur
 - Doivent être pertinentes
 - Correspondre à la charte graphique du site
 - Transmettre des informations

- Insérer une image

``

ex :

```

```

Info bulle : attribut title

```

```

- Il est important que les URL :
 - Ne comportent pas d'espace, les remplacer par ex par '_'
 - Ne comportent pas d'accents
- Afficher une vignette cliquable

```
<p>  
  Vous souhaitez voir l'image dans sa taille d'origine ? Cliquez dessus !<br />  
  <a href="../../../images/pumpkin-1789458_640_large.jpg" target="_blank"></a>  
</p>
```



- Définir la taille de l'image : attributs `height` et `width`

Améliore la performance : permet au navigateur d'établir le rendu de la page et de garder la place pour l'image, pendant qu'elle se charge.

- Les figures
Utile quand une image illustre un texte, lui donne un sens.

```
<figure>
  
  <figcaption>Eléphant en png</figcaption>
</figure>
```

- `<table></table>` balise pour tableau qui se construit ligne à ligne
- `<tr> </tr>` (table row) définition d'une ligne, contient des `<td>`
- `<td> </td>` (table data) définition d'une cellule
- Ouvrir <pages/basic-table-structure.html> avec brackets
- `<th>` pour ajout de tête de colonne et ligne
- Fusion de plusieurs cellules : attributs `rowspan` et `colspan`
Ouvrir <pages/spanning-rows.html>

- L'attribut **id** : **id**entifie tout élément html. La valeur doit être **unique** dans la page html

`<p id= "chanteur" > ... </p>`

- L'attribut **class** : permet de **classer** des éléments html dans des catégories différentes. La valeur peut être utilisée **plusieurs fois** dans la page html

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

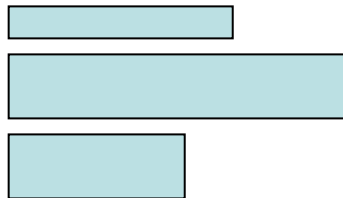
Un élément peut être associé à plusieurs classes

`<p class= "important surligne" > ... </p>`

- Ces 2 notions sont la base de l'emploi du CSS, du code Javascript
...

- Structure de document :

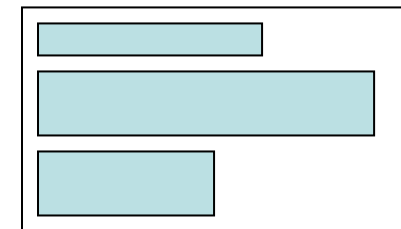
- Éléments de **niveau bloc** : saut de ligne d'un élément à l'autre
ex : `<h1>`, `<p>`, ``, ``



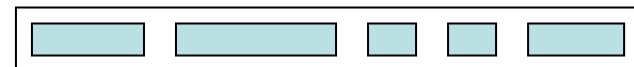
- Éléments de **niveau ligne** : pas de saut de ligne
ex : ``, ``, `<i>`, `<a>` ..



- Conteneur `<div>` : groupe d'éléments de niveau bloc



- Conteneur `` : groupe d'éléments de niveau ligne



Ces conteneurs sont utiles pour y définir des attributs id et class sur ces ensembles d'éléments

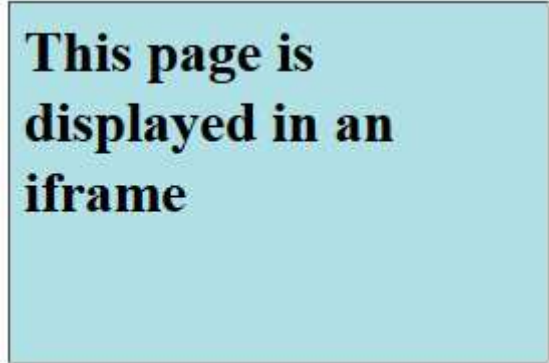
- Cadres

<iframe> définit une région de la page pour contenir une autre page

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

<iframe src="demo_iframe.htm" height="200" width="300"></iframe>

</body>
</html>
```

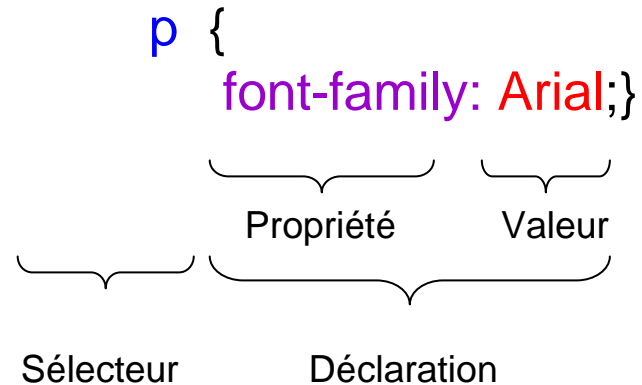


**This page is
displayed in an
iframe**

- Rappel :
 - Le HTML définit le **contenu** et la **structure**
 - Le CSS définit la **présentation**

www.csszengarden.com
Site de démonstration qui permet de changer de CSS à HTML constant
- Le CSS est plus récent que le HTML, pour vérifier si un navigateur implémente une méthode particulière : site www.caniuse.com
- Ce cours est basé sur CSS3

- Instruction CSS



- La/les déclarations **affectent** le/les sélecteurs
autre ex :

```
h1, h2, h3 {  
    font-family: Arial;  
    color: yellow;  
}
```

Signifie que l'on souhaite changer la fonte et la couleur pour les balises html h1, h2, h3

- Où définir le code CSS ?
 - Dans un fichier .css externe au .html : feuille de style
- Le .html fait référence à ce fichier dans le header <head> par la balise <link> :

```
<head>
  <title>Utilisation de CSS externe</title>
  <link href="css/styles.css" type="text/css" rel="stylesheet" />
</head>
```

contenu de styles.css :

```
body {
  font-family: arial;
  background-color: rgb(185,179,175);}
h1 {
  color: rgb(255,255,255);}
```

Le code CSS dans un fichier externe est une bonne pratique :

- Le même code CSS est utilisé par l'ensemble des html du site
- Possibilité d'indiquer un CSS propre à l'impression, propriété [media](#)

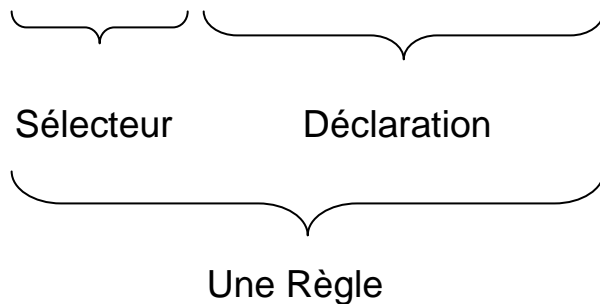
```
<link rel="stylesheet" type="text/css" media="screen" href="sans-serif.css">  
<link rel="stylesheet" type="text/css" media="print" href="serif.css">
```

- Code CSS dans le header du fichier html (peu recommandé):

```
<style type="text/css">  
  body {  
    font-family: arial;  
    background-color: rgb(185,179,175);}  
  h1 {  
    color: rgb(255,255,255);}  
</style>
```

Rappel :

```
p {  
    font-family: Arial;}
```



Un Sélecteur **cible** le/les éléments auxquels la/les déclarations s'appliquent

Regarder http://www.w3schools.com/cssreF/css_selectors.asp

Et tester <http://www.w3schools.com/cssreF/trysel.asp>

- CSS : Cascading Style Sheets
- Si plusieurs règles s'appliquent aux mêmes éléments, le « cascading » définit des priorités :
 - Dernière règle : pour 2 **sélecteurs identiques**, la **dernière règle** s'applique
 - Spécificité : les règles les **plus précises** sont **prioritaires** aux règles plus générales
ex :
 - h1 plus précis que *
 - p b plus précis que b
 - p#intro plus précis que #intro
 - **!important** annule les règles précédentes

```
p b {  
  color: blue !important;}  
p b {  
  color: violet;}
```



- Jeux sur <http://flukeout.github.io/>

- Quelques styles comme font-family, text-alignment sont hérités automatiquement par les éléments html enfants.
- La plupart n'héritent pas. Pour forcer l'héritage, la valeur de la propriété est mise à **inherit**

```
<!DOCTYPE html>
<html>
<head>
<style>
span {
  color: blue;
  border: 1px solid black;
}

.extra span {
  color: inherit;
}
</style>
</head>
<body>

<div>
  Here is <span>a span element</span> which is blue, as span elements are set to be.
</div>

<div class="extra" style="color:magenta">
  Here is <span>a span element</span> which is magenta, because it inherits from its parent.
</div>

<div style="color:red">
  Here is <span>a span element</span> which is blue, as span elements are set to be.
</div>

</body>
</html>
```

Here is a span element which is blue, as span elements are set to be.
Here is a span element which is green, because it inherits from its parent.
Here is a span element which is blue, as span elements are set to be.

Le Texte

- Sont abordés :
 - La police et la taille
 - Gras, italique, majuscule, souligné
 - Espace entre les lignes, les mots et les lettres

- Police : propriété `font-family`

Une police peut être soumise à des droits de licence.

Privilégier des polices courantes, disponibles sur le navigateur client

`font-family` permet de fournir une liste de fontes puis un nom générique de fonte

ex : `font-family: Georgia, Times, serif;`

ex : `font-family: "Courier New ", monospace;`

http://www.w3schools.com/cssreF/pr_font_font-family.asp

- Taille : propriété **font-size**
font-size : <taille> ;

<taille> s'exprime avec plusieurs unités :

- En px (pixels) : permet un contrôle précis de la place occupée
ex : **font-size**: 16px;
- pt (point) : utilisé à priori pour l'impression de la page
- % : en pourcentage de la taille de défaut du navigateur.
100% correspond à 16 px en principe . Ex 75% équivaut à 12px
Des définitions successives dans des éléments imbriqués s'accumulent.
Ex : 75% (12px) suivi d'un 75% donne un 9px
- em : 1 em équivaut à la largeur de la lettre m
ex : **font-size**: 1.3em;

- Taille suite
 - vw : 1vw est 1% de la largeur du viewport. Le viewport est la zone utilisateur visible du navigateur. Est donc sensible au type de périph (mobile, PC, tablette ...) . Utile pour des sites « responsives »
 - vh : 1vh est 1% de la hauteur du viewport
 - vmin : 1vmin est la plus petite valeur entre 1vw et 1vh
 - vmax : 1vmax est la plus grande valeur entre 1vw et 1vh

Le redimensionnement du navigateur fait changer la taille du texte.

- Import d'une police.
Si absente du navigateur coble, [@font-face](#) permet de charger la police

```
@font-face {  
    font-family: myFirstFont;  
    src: url(sansation_light.woff);  
}
```

- Gras

propriété **font-weight**

font-weight: normal|bold|bolder|lighter|*number*|initial|inherit;

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

```
em {  
  font-weight: bold;}
```

- Italique

propriété **font-style**

font-style: normal|italic|oblique|initial|inherit;

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

- Majuscule, minuscule

propriété **text-transform**

text-transform: none|capitalize|uppercase|lowercase|initial|inherit;

- Interligne
propriété `line-height`
`line-height`: normal|*number*|initial|inherit;
éviter de fixer number en pixels px
- Espaces entre lettres et mots
propriété `letter-spacing` , `word-spacing`
`letter-spacing`: normal|*length*|initial|inherit;
`word-spacing`: normal|*length*|initial|inherit;
- Alignement
propriété `text-align`
`text-align`: left|right|center|justify|initial|inherit;
- Alignement vertical
propriété `vertical-align`
`vertical-align`: baseline|*length*|sub|super|top|text-top|middle|bottom|text-bottom|initial|inherit;

- Ombre portée
propriété `text-shadow`
`text-shadow: h-shadow v-shadow blur-radius color|none|initial|inherit;`
- Style des liens
 - `a:link` - a normal, unvisited link
 - `a:visited` - a link the user has visited
 - `a:hover` - a link when the user mouses over it
 - `a:active` - a link the moment it is clicked

http://www.w3schools.com/css/css_link.asp

- propriété `color` et `background-color`

La valeur peut être exprimée sous plusieurs formes :

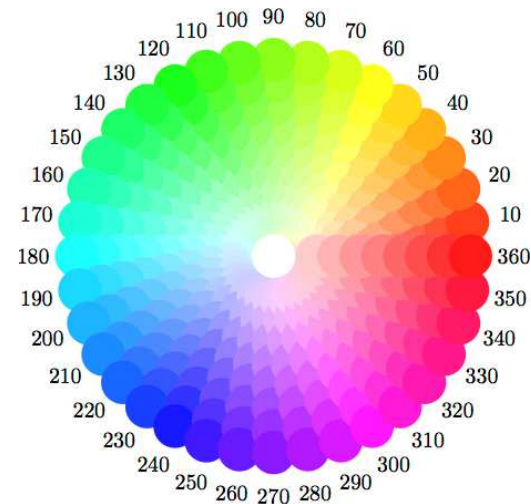
- Par le nom : `DarkCyan`
- Par les composantes rgb : `rgb(255, 0, 100)`
- Par les composantes rgb en hexa : `#ff004a`
- L'opacité peut être ajoutée : `rgba(255, 0, 100, 0.8)` 0.8 très opaque
- Spécifique à CSS3 : méthode hsl et hsla
h : hue (teinte) en degré
s : saturation en %
l : luminosité en %
a : alpha, opacité entre 0 et 1

`hsla(0, 100%, 80%, 0.6)`

<http://hslpicker.com/>

pour le choix d'une couleur

- Exercice 2



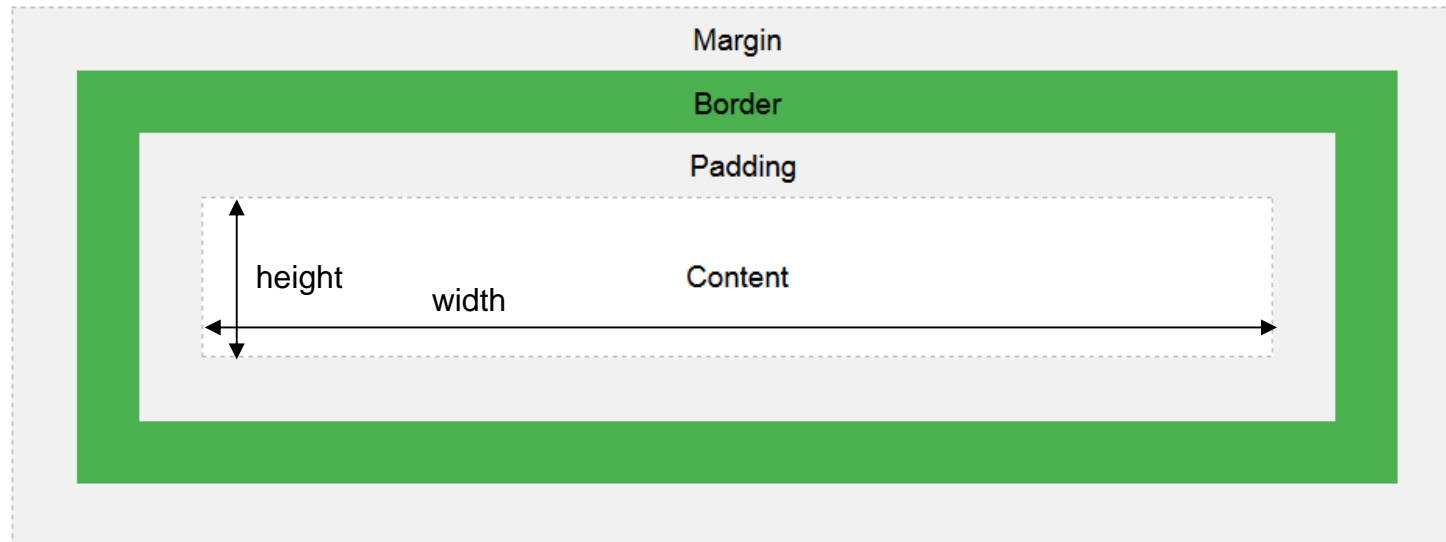
- CSS permet de gérer les éléments html comme s'ils étaient dans des boîtes.



Ce chapitre décrit :

- Le contrôle des dimensions des boîtes
- Bordures autour des boîtes
- Marges et espacement
- Affichage/masquage des boîtes

- Dimension d'une boîte

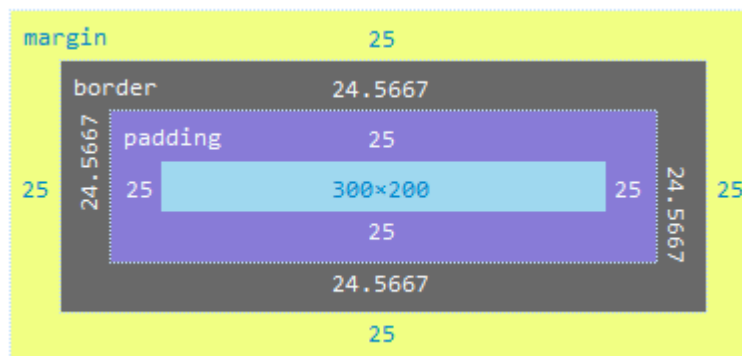
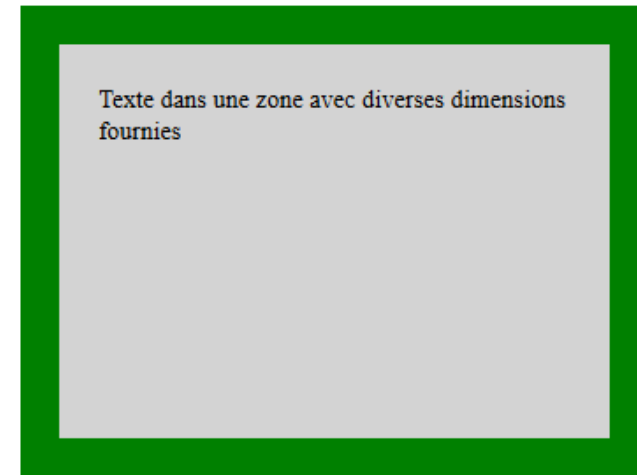


- content : endroit du texte et des images
- height, width : hauteur et largeur de content
- padding : zone vide autour de content. Transparente.
- border : bordure autour de content et padding
- margin : zone vide autour de border. Transparente.

- Ex :

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: lightgrey;
  width: 300px;
  height: 200px;
  border: 25px solid green;
  padding: 25px;
  margin: 25px;
}
</style>
</head>
<body>
  <h2>Démonstration des boîtes</h2>
  <div>Texte dans une zone avec diverses dimensions fournies</div>
</body>
</html>
```

Démonstration des boîtes



Utilisation du mode « examiner l'élément » de Firefox

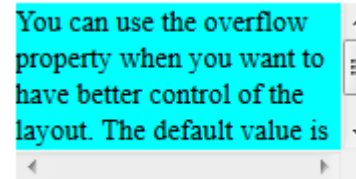
- Autres propriétés de dimensionnement :
 - width: 50%; l'élément occupera 50% de son conteneur
 - min-width, max-width
 - min-height, max-height

```
p {  
    width:50%;  
    min-width: 300px;  
}
```

- Gérer le dépassement

overflow: visible|hidden|scroll|auto|initial|inherit;

```
div.scroll {  
    background-color: #00FFFF;  
    width: 100px;  
    height: 100px;  
    overflow: scroll;  
}
```



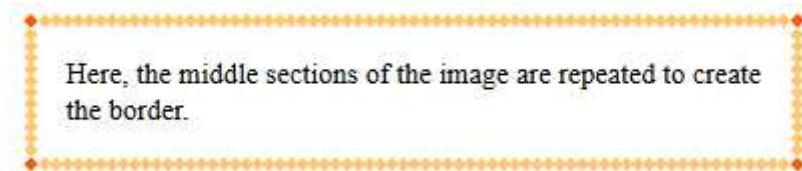
- Quelques propriétés des bordures
 - Généralité : à une propriété `<prop>` qui définit les 4 bords, il existe les propriétés:
 - `<prop>-top`, `<prop>-right`, `<prop>-bottom`, `<prop>-left`

ex : `padding-right` `margin-top`
 - **`border`**: *`border-width border-style border-color|initial|inherit`*;
avec
`border-style:none|hidden|dotted|dashed|solid|double|groove|ridge|inset|outset|initial|inherit`;
 - Les propriétés **`border-width`**, **`border-style`**, **`border-color`** existent séparément

`border-width: 5px, 1px, 4px, 3px;` pour épaisseur bords top, right, bottom, left
`border-width: 5px 1px;` 5px pour top bottom, 1px pour right left
`border-width: 2px;` 2px pour les 4 bords
`border-width-top: 2px;` 2px pour top
 - **Margin**
Règle l'espace entre boîtes. Le mot `auto` permet le centrage
ex : **`margin`**: 2cm auto 3cm auto;

- border-image

```
#borderimg1 {  
  border: 10px solid transparent;  
  padding: 15px;  
  -webkit-border-image: url(border.png) 30 round; /* Safari 3.1-5 */  
  -o-border-image: url(border.png) 30 round; /* Opera 11-12.1 */  
  border-image: url(border.png) 30 round;  
}
```



- box-shadow

`box-shadow: none|h-shadow v-shadow blur spread color |inset|initial|inherit;`

```
div#myDIV {  
  background-color:yellow;  
  width:200px;  
  height:100px;  
  box-shadow:10px 10px black;  
}  
  
div#myDIV {  
  background-color:yellow;  
  width:200px;  
  height:100px;  
  box-shadow:50px 50px 20px black;  
}
```



- border-radius

`border-radius: 1-4 length|% / 1-4 length|%|initial|inherit;`

`border-radius: 15px 50px 30px 5px;`



– Masquage

- `visibility: visible|hidden|collapse|initial|inherit;`
`visibility: hidden;` cache l'élément mais garde sa place
- `display: none;` cache l'élément et rend la place

- Tout élément html cliquable peut être présenté de façons différentes dans les situations :
 - Au survol :hover
 - Lors du clic (éphémère) :active
 - Lors de la sélection :focus
 - Lorsque le lien a déjà été sélectionné :visited
 - Lorsque le lien n'a jamais été sélectionné :link

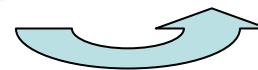
```
a:link {  
    color: green;  
}
```

Mouse over and click the link: w3schools.com

```
input[type=text]:focus {  
    width: 250px;  
}
```

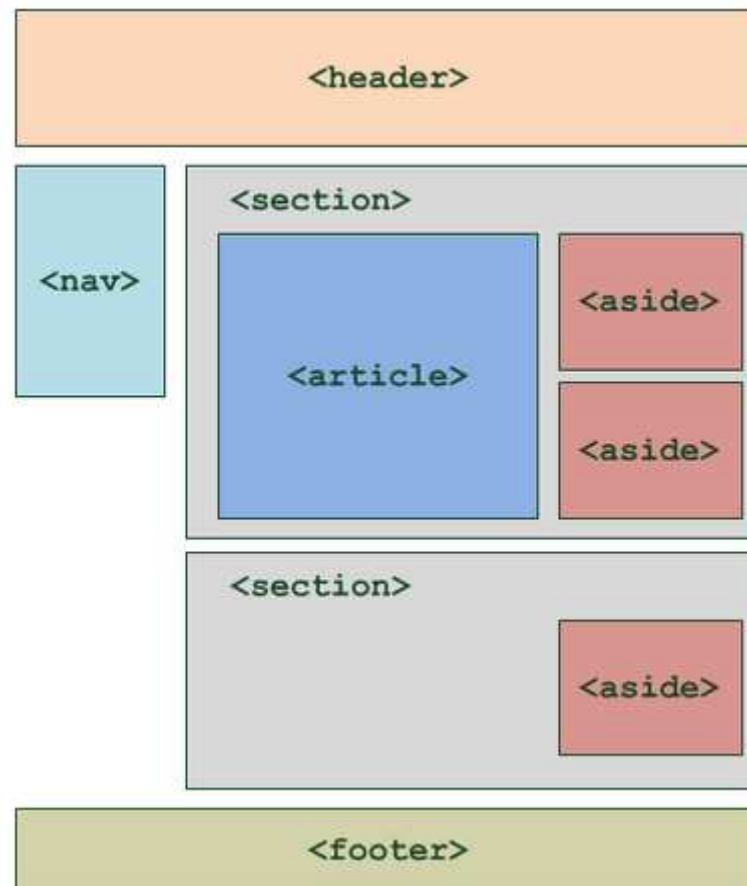
Search:

Search:



Clic dans la zone => focus

- Html5 a mis en place des balises (éléments) pour structurer la page



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Une page Web structurée</title>
  </head>
  <body>
    <header>
      <h1>Les cours HTML5 CSS3</h1>
      <h2>Commençons par HTML 5</h2>
    </header>
    <nav>
      <ul>
        <li><a href="#">Accueil</a></li>
        <li><a href="#">Blog</a></li>
        <li><a href="#">références</a></li>
      </ul>
    </nav>
    <section>
      <aside>
        <h1>Un petit historique</h1>
        <p>L'HyperText Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques. </p>
      </aside>
      <article>
        <h1>HTML paru dans la presse</h1>
        <p>Bla bla bla bla (texte de l'article)</p>
      </article>
    </section>
    <footer>
      <p>Copyright LECOURS - Tous droits réservés<br />
      <a href="#">Me contacter !</a></p>
    </footer>
  </body>
</html>
```

Les cours HTML5 CSS3

Commençons par HTML 5

- [Accueil](#)
- [Blog](#)
- [références](#)

Un petit historique

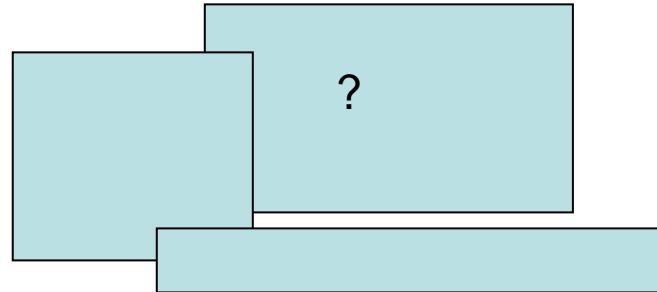
L'HyperText Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques.

HTML paru dans la presse

Bla bla bla bla (texte de l'article)

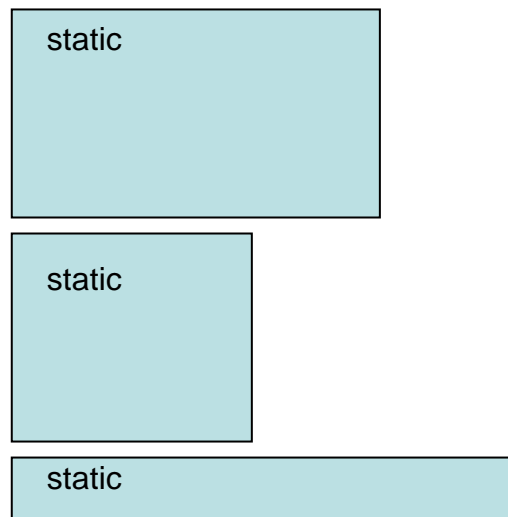
Copyright LECOURS - Tous droits réservés
[Me contacter !](#)

- Il s'agit de positionner des blocks les uns par rapport aux autres



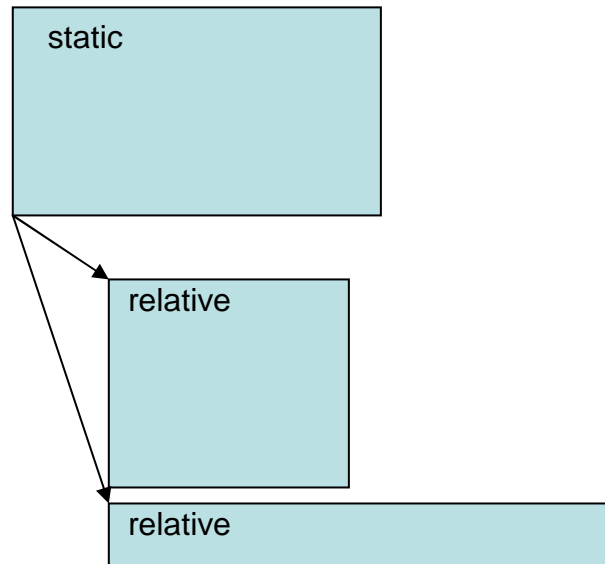
- Par défaut : `position: static;`
Les blocs s'affichent les uns en dessous des autres. Tient compte de l'ordre des blocs.

```
.bloc20 {  
  width: 20%;  
  border-style: solid;  
  position: static;  
}
```



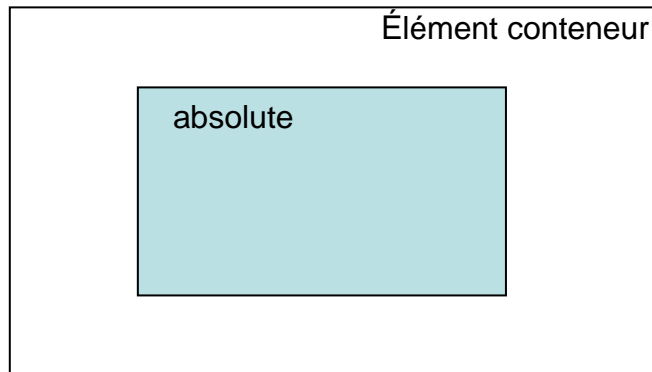
- position: relative;
Permet de déplacer un bloc relativement par rapport au bloc précédent de type static

```
.rel {  
  position: relative;  
  top: 10px;  
  left: 100px;  
}
```



- `position: absolute;`
Permet de placer un bloc précisément à un endroit par rapport à son conteneur non static, sans tenir compte de l'ordre des blocs

```
.abs {  
  position: absolute;  
  top: 10px;  
  left: 100px;  
}
```



- `position: fixed;`
Permet de placer un bloc précisément à un endroit, relativement à la fenêtre du navigateur (crée des zones insensibles au scroll)

- Propriété float

`float: none|left|right|initial|inherit;`

Place le bloc à l'extrême gauche ou droite de son conteneur.

`width:` doit être fourni

- Propriété clear

`clear: none|left|right|both|initial|inherit;`

Indique que dans un même conteneur, pour ce bloc, aucun bloc ne doit se trouver à gauche, à droite, des 2 côtés;

- Réaliser l'exercice 3



- Propriété display
display: inline;

fait passer des éléments html de type bloc en éléments de type ligne

display: block;

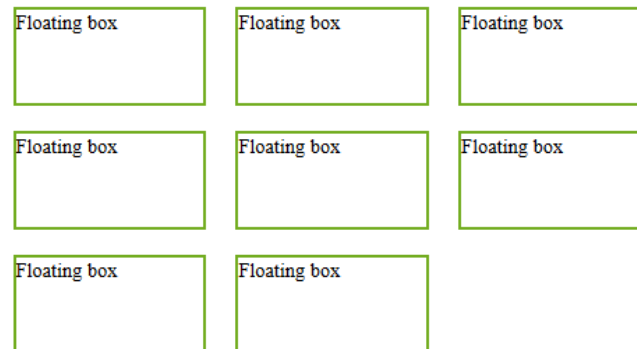
fait passer des éléments html de type ligne en éléments de type bloc

display:inline-block;

remplace l'usage de float: et clear: pour positionner des blocs à l'horizontale

```
.floating-box {  
    display: inline-block;  
    width: 150px;  
    height: 75px;  
    margin: 10px;  
    border: 3px solid #73AD21;  
}
```

The New Way - using inline-block

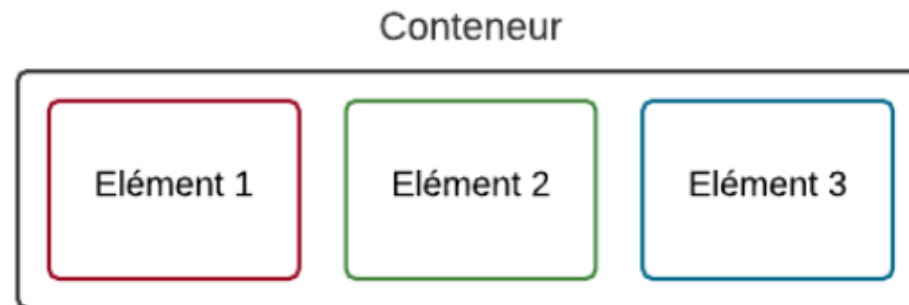


Another box, after the floating boxes...

- Exercice 4



- Historiquement les développeurs de site ont utilisé pour la mise en page :
 - La propriété float suivie de clear
 - La propriété display: inline-block; mieux gérée par les navigateurs
 - Sur des nouveaux sites il est avantageux d'utiliser Flexbox
- Flexbox
Flexbox réalise la mise en page :
 - crée la notion de conteneur
 - gère l'organisation des blocs dans ce conteneur



- Affichage par défaut

```
<body>
  <div id="conteneur">
    <div class="element">Elément 1</div>
    <div class="element">Elément 2</div>
    <div class="element">Elément 3</div>
  </div>
</body>
```



- Fonctionnalité flex :

```
#conteneur {
  display: flex;
  border: 2px solid red;
  margin: 10px;
}
```



- Propriétés associées à flex :
 - flex-direction: row|row-reverse|column|column-reverse|initial|inherit;
 - flex-wrap: nowrap|wrap|wrap-reverse|initial|inherit;
 - Aligner sur l'axe principal
justify-content: flex-start|flex-end|center|space-between|space-around|initial|inherit;
 - Aligner sur l'axe secondaire
align-items: stretch|center|flex-start|flex-end|baseline|initial|inherit;
 - Déterminer l'ordre d'affichage d'un bloc order:
 - Entraînement sur <https://flexboxfroggy.com/#fr>
Au moins jusqu'à l'item 15/24



- Donner une proportion aux éléments : propriété flex:

```
.conteneur {  
  display: flex;  
  border: 2px solid red;  
  margin: 10px;  
}  
  
.element20 {  
  flex: 0.2;  
  color: green;  
  border: 2px solid;  
  padding: 10px;  
  margin: 2px;  
}  
  
.element60 {  
  flex: 0.6;  
  color: green;  
  border: 2px solid;  
  padding: 10px;  
  margin: 2px;  
}
```

CSS

html

```
<div class="conteneur">  
  <div class="element20">Elément 1</div>  
  <div class="element60">Elément 2</div>  
  <div class="element20">Elément 3</div>  
</div>
```

Elément 1

Elément 2

Elément 3

Exercice 5



- Permet de saisir des données dans le navigateur et de les envoyer au serveur. Elles seront traitées côté serveur, code PHP ...

Inscription des élèves

Nom

Prénom

Classe

Sexe Homme ☒ Femme ☐

Option Langue ☐ Dessin ☐ Musique ☐ Scientifique ☐

Journée

Envoie les données au serveur qui réalise le traitement et crée si nécessaire une nouvelle page

- Chaque élément de saisie s'appelle un contrôle de formulaire
- Chaque contrôle a un attribut **name**. Cette valeur est connue du serveur car lui permet d'identifier les données entre elles.

Option=Dessin

↑ ↑
name valeur saisie

- Code html de base : élément <form>

```
<body>
  <p>Texte avant le formulaire</p>

  <form>
    <p>Texte à l'intérieur du formulaire</p>
  </form>

  <p>Texte après le formulaire</p>
</body>
```

- Attribut action
Fournit l'URL de la page qui recevra les données
- Attribut method
Indique la méthode d'envoi vers le serveur :
 - get : les valeurs saisies sont ajoutées en fin d'URL action
 - Pour petit formulaire
 - Vulnérable
 - post : les valeurs saisies sont envoyées dans les en-têtes HTTP
 - Gros volume de données
 - Moins visible mais vulnérable

```
<form method="post" action="http://www.xxx.fr/traitement.php">
```

- Liste des principaux contrôles

- Saisie de texte

```
Last name:<br>  
<input type="text" name="lastname">
```

Last name:

- Saisie de mot de passe

```
User password:<br>  
<input type="password" name="psw">
```

User password:

- Bouton submit

```
<input type="submit" value="Submit">
```

Submit

- Bouton reset (remet les valeurs par défaut)

```
<input type="reset">
```

- bouton radio

```
<form>  
  <input type="radio" name="gender" value="male" checked> Male<br>  
  <input type="radio" name="gender" value="female"> Female<br>  
  <input type="radio" name="gender" value="other"> Other  
</form>
```

☒ Male
☐ Female
☐ Other

- Check box

```
<input type="checkbox" name="vehicle1" value="Bike"> I have a bike<br>  
<input type="checkbox" name="vehicle2" value="Car"> I have a car
```

☐ I have a bike
☐ I have a car

-

Exercice 6 : créer un formulaire.



- Avant HTML5 le player Flash était le plus employé pour l'audio et l'animation
- HTML5 : balises <audio> et <video>
- Les formats audio :
 - mp3 : ancien et le plus compatible
 - AAC : format Apple
 - OCG : du monde Linux, libre et sans droit
 - WAV : à oublier à cause de sa taille
- Les formats vidéo : il faut considérer 3 sous-parties :
 - Le format conteneur : ex .avi .mp4 ..
 - Le codec audio : mp3, AAC, OCG, ..
 - Le codec vidéo : compression de l'image, pas forcément libre (et gratuit)
 - H.264 : puissant et très utilisé. Non libre de droits.
 - Ogg Thera, moins puissant que H.264 mais libre
 - WebM, proposé par Google, concurrent gratuit à H.264
- Site Caniuse.com pour compatibilité avec les navigateurs

- L'outil gratuit Miro Video Converter permet de convertir des vidéos dans le format souhaité.

- Balise <audio>

```
<audio controls>  
  <source src="horse.ogg" type="audio/ogg">  
  <source src="horse.mp3" type="audio/mpeg">  
Your browser does not support the audio element.  
</audio>
```

- Balise <video>

```
<video width="320" height="240" controls>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogg" type="video/ogg">  
Your browser does not support the video tag.  
</video>
```

- Le terme « responsive » signifie que le contenu et la disposition de la page s'adapte en fonction de la taille de l'écran qui l'expose :
 - Cacher des éléments secondaires sur un petit écran
 - Changer la disposition d'éléments selon la taille de l'écran ou son orientation portrait/paysage
 - ...
- Utiliser la puissance des « Media Queries »
Puisque c'est le CSS qui détermine la forme, des évolutions en CSS3 ont été faites sous formes de règles (pas des nouvelles propriétés)
- Il existe 2 façons d'appliquer les Media queries :
 - Un fichier css spécifique pour chaque contexte

```
<head>
  <meta charset="utf-8" />
  <link rel="stylesheet" href="style.css" />  <!-- Pour tout le monde -->
  <link rel="stylesheet" href="petite_resolution.css" />  <!-- Pour ceux qui ont une résolution inférieure à 1280px -->
  <link rel="stylesheet" media="screen and (max-width: 1280px)" href="petite_resolution.css" />
  <title>Media queries</title>
</head>
```

- ...appliquer les Media queries – suite
 - En utilisant des règles dans le même fichier CSS
 - Voir http://www.w3schools.com/cssref/css3_pr_mediaquery.asp
- Compléter le TP « site du vélo électrique » pour le rendre responsive

- Javascript
Ce langage interprété et géré côté navigateur rend plus dynamique les pages, sans solliciter le serveur.

Fait partie d'une autre formation
- HTML5 <canvas> permet de faire du dessin dans une page
- HTML5 <SVG> permet de faire du dessin vectoriel.