

Le sélecteurs avancées (ou complexes)

- En CSS, il existe des sélecteurs avancées qui permettent de cibler des éléments du document HTML :
 - Le sélecteur universel
 - Les sélecteurs d'attribut
 - Les combinateurs:
 - Les sélecteurs de voisin direct
 - Les sélecteurs de voisins
 - Les sélecteurs d'éléments fils
 - Les sélecteurs d'éléments descendants

Le sélecteur universel

- Ce sélecteur permet de cibler **tous** les éléments d'un document d'un seul coup.
- Syntaxe : *



```
*{  
    color : #88BB11;  
}
```

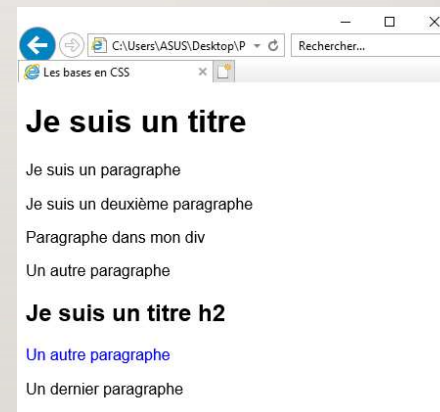
```
<body>  
  <h1>Je suis un titre</h1>  
  <p>Je suis un paragraphe</p>  
  <p>Je suis un deuxième paragraphe</p>  
  <p>Je suis un troisième paragraphe</p>  
</body>
```

Le sélecteur d'attribut

- Ce sélecteur permet de cibler des éléments **en fonction de la valeur d'un de leurs attributs**.
- **Syntaxe** : `[attr]` `[attr=valeur]` `[attr*=valeur]`
- **Exemples** :
 - `a[title]`
 - permet de cibler tous les liens `<a>` qui **possèdent** un attribut `title` (quelle que soit sa valeur).
 - `a[title="Cliquer ici"]`
 - idem mais l'attribut **doit en plus avoir** exactement pour valeur « Cliquez ici ».
 - `a[title*="ici"]`
 - idem, l'attribut doit cette fois **contenir dans sa valeur** le mot « ici » (peu importe sa position).

Le sélecteur de voisin direct

- Le combineur '+' permet de sélectionner les éléments qui **suivent immédiatement** un élément donné.
- Syntaxe : **A + B**
- Exemple : **div + p** permettra de cibler n'importe quel élément **<p>** qui suit immédiatement un élément **<div>**.



```
<body>
  <h1>Je suis un titre</h1>
  <p>Je suis un paragraphe</p>
  <p>Je suis un deuxième paragraphe</p>
  <div>
    <p>Paragraphe dans mon div</p>
    <p>Un autre paragraphe</p>
    <h2>Je suis un titre h2</h2>
  </div>
  <p>Un autre paragraphe</p>
  <p>Un dernier paragraphe</p>
</body>
```

```
body{
  font-family: 'Open Sans', sans-serif;
}

div + p{
  color : blue;
}
```

Le sélecteur de voisins

- Le combinateur '~' permet de sélectionner les éléments qui suivent un élément et qui ont le même parent.
- Syntaxe : **A ~ B**
- Exemple : **h1 ~ p** permettra de cibler les éléments **<p>** qui suivent (immédiatement ou non) un élément **<h1>** et qui ont le même élément parent.



```
<body>
  <h1>Titre h1</h1>
  <p>L'élément em est un élément de type inline</p>
  <div>
    <p>Un paragraphe</p>
  </div>
  <div>
    <h1>Un autre titre h1</h1>
    <p>L'élément p est un élément de type block</p>
    <p>Un <strong> autre</strong> paragraphe</p>
  </div>
</body>
```

```
body{
  font-family: 'Open Sans', sans-serif;
}

h1~p{
  color : blue;
}
```


Le sélecteur d'éléments fils

- Le combineur '>' permet de sélectionner les éléments qui sont des **filis directs** d'un élément donné.
- Syntaxe : **A > B**
- Exemple : **ul > li** permettra de cibler tous les éléments **** qui sont **directement** situés **sous** un élément ****.



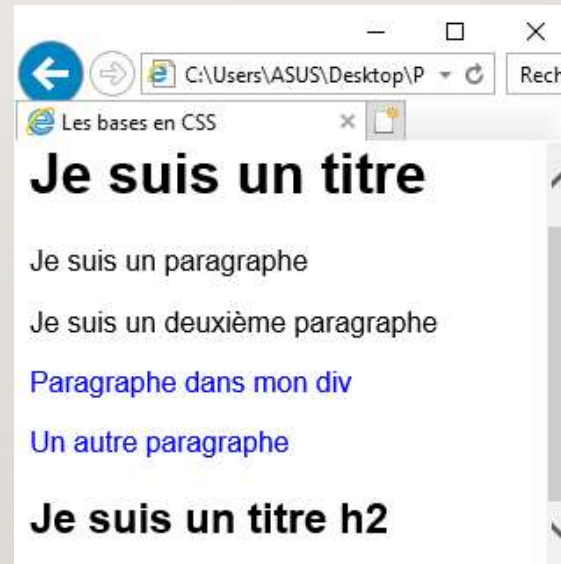
```
<body>
  <h1>Je suis un titre</h1>
  <h2>Un premier titre h2</h2>
  <p>Je suis un paragraphe</p>
  <p>Je suis un deuxième paragraphe</p>
  <div>
    <p>Un paragraphe de plus</p>
    <h2>Titre h2</h2>
  </div>
  <div>
    <p>Paragraphe dans mon div</p>
    <p>Un autre paragraphe</p>
    <h2>Je suis un titre h2</h2>
  </div>
  <p>Un autre paragraphe</p>
  <p>Un dernier paragraphe</p>
</body>
```

```
body{
  font-family: 'Open Sans', sans-serif;
}

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

121 Le sélecteur d'éléments descendants

- Le combineur (espace) permet de sélectionner les éléments qui sont **des descendants** (pas nécessairement des fils directs) d'un élément donné
- **Syntaxe : A B**
- **Exemple : `div p`** permettra de cibler n'importe quel élément `<p>` situé à l'intérieur d'un élément `<div>`.



```
<body>
  <h1>Je suis un titre</h1>
  <p>Je suis un paragraphe</p>
  <p>Je suis un deuxième paragraphe</p>
  <div>
    <p>Paragraphe dans mon div</p>
    <p>Un autre paragraphe</p>
    <h2>Je suis un titre h2</h2>
  </div>
</body>
```

```
body{
  font-family: 'Open Sans', sans-serif;
}

div p{
  color : blue;
}
```

Le pseudo-classes

- Les pseudo-classes permettent de changer le style d'un élément HTML selon son état, d'une façon quasi-dynamique
- Par exemple, pouvoir afficher un paragraphe en gras lorsque l'utilisateur passe la souris dessus ou changer la couleur d'un lien une fois que celui-ci a été cliqué
 - **:link** : état d'un lien non visité
 - **:visited** : état d'un lien visité
 - **:hover** : quand la souris est sur l'élément
 - **:focus** : état d'un élément sélectionné
 - **:active** : état d'un élément au moment de click

A utiliser dans cet ordre !

Le pseudo-éléments

- Les pseudo-éléments permettent de modifier l'apparence d'une partie d'un ou de plusieurs éléments HTML ou encore ajouter du contenu au début ou à la fin d'un élément HTML
 - **::first-letter** : **premier caractère** d'un élément (lettrine)
 - **::first-line** : **première ligne** d'un élément
 - **::selection** : **une partie du document** qui a été sélectionnée par l'utilisateur (via la souris ou un autre dispositif de pointage).
 - **::before** : **insérer** un contenu **avant** l'élément
 - **::after** : **insérer** un contenu **après** l'élément

124

CSS : PROPRIÉTÉS DE MISE EN FORME

Unités de mesure

- On distingue des unités de mesure fixes (absolues) ou relatives.
- **Unités de mesure absolues :**
 - pt : point typographique (1/72 de pouce)
 - pc : pica typographique (12 points)
 - in : pouce (2.54cm)
 - mm : millimètre
 - cm : centimètre
- **Unités de mesure relatives :**
 - em : relative à la taille de police par défaut de l'élément parent.
 - % : relative à la dimension de l'élément parent
 - px : pixel, relative à la densité de pixel du mode de sortie

Mise en forme du texte

- **font-family** : Police ou famille de polices (serif, sans-serif, cursive, fantasy, monospace).
 - Le navigateur essaiera d'abord d'utiliser la 1^{ère} police. S'il ne l'a pas, il essaiera la deuxième, etc.
 - En général, il est bien d'indiquer un choix de trois ou quatre polices (+ Serif ou sans-serif), afin de s'assurer qu'au moins l'une d'entre elles aura été trouvée sur le navigateur du visiteur.

Texte en Arial

Texte en Arial Black

Texte en Comic Sans MS

Texte en Courier New

Texte en Georgia

Texte en Impact

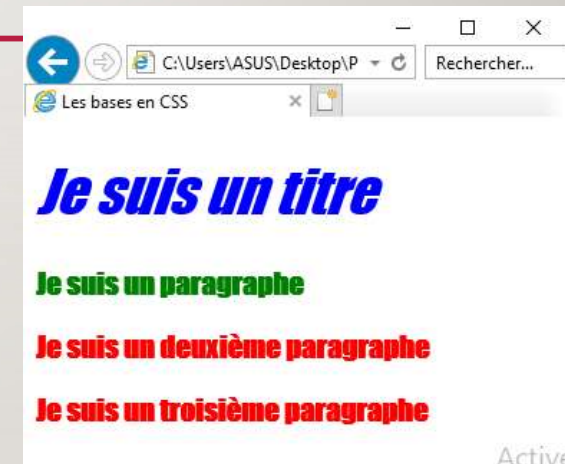
Texte en Times New Roman

Texte en Trebuchet MS

Texte en Verdana

Mise en forme du texte

- **font-size** : Taille de la police
 - valeur numérique (% , **em**, **px**, **cm**, **mm**...)
 - valeur relative (**xx-small**, **x-small**, **small**, **medium**, **large**, **x-large**, **xx-large**)
- **font-style** : Style de la police
 - **italic**, **oblique**, **normal** (par défaut).
- **font-variant** : Casse des caractères
 - **normal**
 - **small-caps** (petites majuscules).
- **font-weight** : Graisse des caractères
 - **normal**
 - **bold**



```
body{
  font-family: Impact, "Arial Black", Arial, Verdana, sans-serif;
}

h1{
  font-size: 40px;
  font-style: italic;
  font-weight: normal;
}

p{
  font-size: 20px;
  font-style: normal;
  font-weight: bold;
}
```

Mise en forme du texte

- **text-decoration** : effet de texte
 - **none** (normal, valeur par défaut)
 - **underline**. (souligné)
 - **overline** (surligné)
 - **blink** (clignotant)
 - **line-through** (barré).
- **text-transform** : effet de case
 - **none** (par défaut)
 - **capitalize** (1ère lettre Maj)
 - **uppercase** (Maj)
 - **lowercase** (Min)
- **text-align** : alignment
 - **left**, **right**, **center**, **justify**
- **text-indent** : retrait (% ,em,px,cm,mm...)
- **line-height** : interligne (% ,em, px,cm,mm...)
- **word-spacing** : espacement des mots
 - **normal**
 - valeur (% ,em,px,cm,mm...)
- **letter-spacing** : espacement des caractères
 - **normal**
 - valeur (% ,em,px,cm,mm...)

Mise en forme du texte

- **vertical-align** : alignement vertical
 - valeur (%;em;px...)
 - **baseline** (par défaut)
 - **sub**
 - **super**
 - **top**
 - **text-top**
 - **middle**
 - **bottom**
 - **text-bottom**
- **color** : couleur de police
 - **color:#990066** (hexadecimal)
 - **color:red** (mot-clés)
 - **color:RGB**(Rouge, Vert, Bleu)
 - Les valeurs de Rouge, vert et Bleu peuvent varier de 0 à 255
- **opacity** : niveau d'opacité (ou transparence)
 - valeur entre 0 (texte transparent) et 1 (texte opaque)
 - Elle peut être définie également dans la propriété **color** en utilisant la valeur : **RGBa**
 - **color:RGBa(128,0,128,0.6)**

Mise en forme des listes

- **list-style-type** : modifie la marque des items de liste :
 - `none` (pas de marque)
 - `disc, circle, square,` ● ○ ■
 - `lower-roman, upper-roman, lower-alpha, upper-alpha` (1. i I a A)
- **list-style-position** : position de la marque (intérieure ou extérieure)
 - `outside`
 - `inside`
- **list-style-image** : utiliser une image comme marque des items
 - `url('image.gif')`

Mise en forme des tables

- **border-collapse** : fusion des bordures
 - `separate` (séparées)
 - `collapse` (fusionnées)
- **border-spacing** : espacement entre les cellules
 - valeur (%;em;px...)
 - 2px (horizontal)
 - 2px 3px (h / v)
- **empty-cells** : les bordures des cellules vides.
 - `show`
 - `hide`
- **table-layout** : largeur de la table
 - `auto` (100%, par défaut)
 - `fixed` (largeur 1ère ligne)
- **caption-side** : Position du titre
 - `top` (en haut)
 - `bottom` (en bas)
- **width** : largeur
 - `auto`
 - valeur (%;em; px;cm;mm...)
- **vertical-align** : Alignement vertical
 - `top`, `bottom`, `middle`, `baseline`

I32

CSS : MODÈLE DES BOÎTES

3^{ÈME} ANNÉE LICENCE EN INFORMATIQUE

FSM 2020-2021

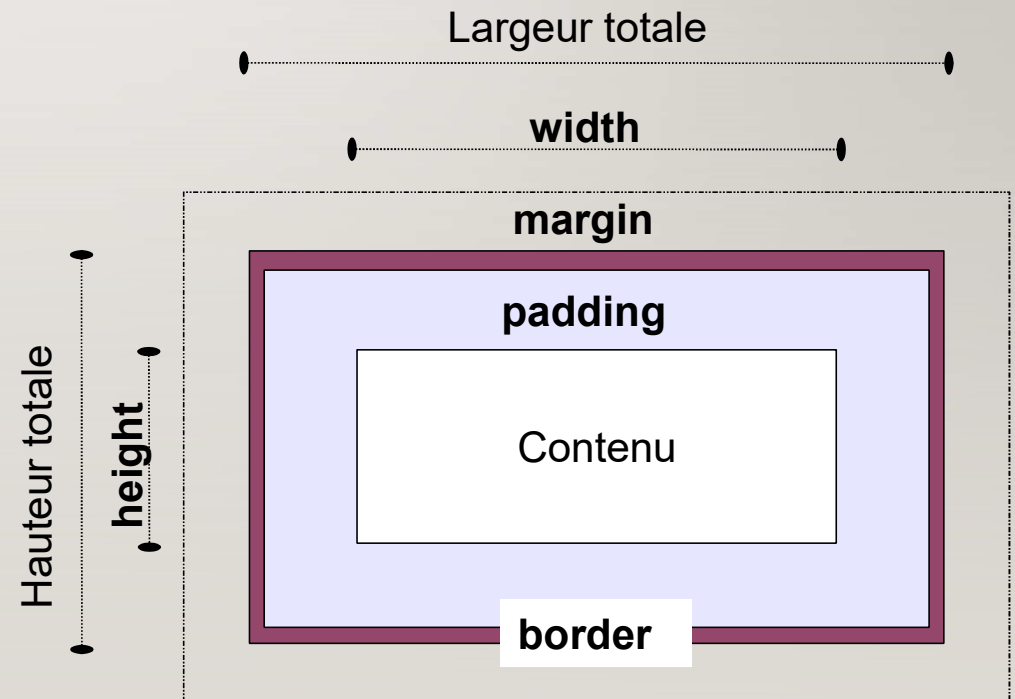


Le modèle des boîtes

- Le modèle des boîtes est un concept essentiel pour ensuite mettre en page et positionner les différents éléments du page web.
- Le modèle des boîtes repose sur le concept que « Tout élément HTML peut être considéré comme une boîte rectangulaire de type **block** ou **inline** ».
- Les boîtes de type **block** (sous forme bloc), qui s'affichent les unes en dessous des autres.
 - Ex : div, h1, p, ul, ol, dl, table, ...
- Les boîtes de type **inline** (en-ligne), qui s'affichent les unes à côté des autres.
 - Ex : span, a, strong, em, img, q, cite, acronym, abbr...

Le modèle des boîtes

- Autour de la boîte contenant l'élément en soit, on peut dessiner d'autres boîtes qui représentent les propriétés permettant d'indiquer la longueur, la largeur, la marge extérieure, le padding (marge intérieure) et les bordures d'un élément forment le modèle des boîtes.
- Largeur totale : $\text{width} + \text{padding} + \text{border}$
- Hauteur totale : $\text{height} + \text{padding} + \text{border}$

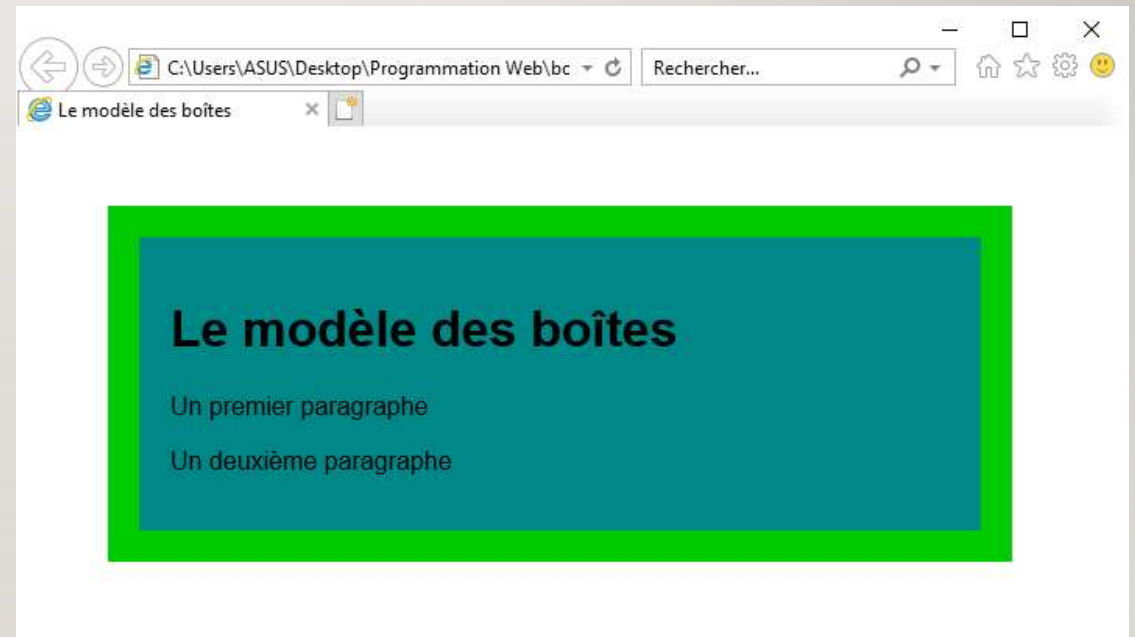


Le modèle des boîtes : Exemple

```
<body>
  <div class="div1">
    <h1>Le modèle des boîtes</h1>
    <p class="p1">Un premier paragraphe</p>
    <p class="p2">Un deuxième paragraphe</p>
  </div>
</body>
```

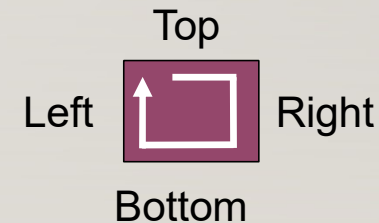
```
body{
  font-family: 'Open Sans', sans-serif;
}

.div1{
  background-color: #088;
  width: 500px;
  padding: 20px;
  border: 20px solid #0C0;
  margin: 50px;
}
```



Le modèle des boîtes

- Pour une boîte de type **bloc**, toutes les propriétés de dimensions et de marges sont stylables.
- Pour une boîte de type **inline**, les propriétés **width** et **height** ne sont pas stylables, la propriété **margin** n'aura d'effet que sur les marges gauche et droite.
- Chaque dimension (hauteur, longueur, marges) peut être traitée de manière indépendante grâce aux mots clés : **top**, **right**, **bottom**, **left**:
 - `border-right: 1px` (bordure droite de 1 pixel)



137

CSS : PROPRIÉTÉS DU MODÈLES DES BOÎTES

Largeur et hauteur

- **width** : largeur du contenu
 - auto
 - %,em, px,cm,mm...
- **min (max)-width** : largeur minimale/ maximale du contenu
 - min-width: 5em (%,em, px,cm,mm...)
- **height** : hauteur du contenu
 - auto
 - %,em, px,cm,mm...
- **min(max)-height** : hauteur minimale/ maximale du contenu
 - min-height: 5em (%,em, px,cm,mm...)

Les marges

- **margin** : marge extérieure
 - auto
 - %,em, px,cm,mm...
- **margin-val** : marge extérieure spécifique
 - val peut être : top right bottom left
 - margin-top: 2px (marge haute 2 pixels)
- **padding** : marge intérieure
 - auto
 - %,em, px,cm,mm...
- **padding-val** : marge intérieure spécifique
 - val peut être : top right bottom left
 - padding-top: 2px (marge haute 2 pixels)

Les bordures

- **border** : bordure
 - La propriété **border** peut être associée aux mots clés **top**, **right**, **bottom**, **left**, **style**, **color** pour construire des propriétés composées.
 - **style** peut prendre les valeurs : **none**, **dotted**, **dashed**, **solid**, **double**, **groove**, **ridge**, **inset**, **outset**
 - **width** peut utiliser toutes les unités (% ,em, px,cm,mm...)
- **Exemples :**
 - border-top-style:double
 - border-width:1px
 - border-bottom-width:3px
 - border-left-color:#000000
 - border-color:#000000

Les bordures

- **border-radius** : créer des bordures arrondis
 - **X** (axe horizontal) / **Y** (axe vertical)
 - Ces deux valeurs vont pouvoir être des valeurs de type longueur (en **px** par exemple) ou des pourcentages (%).
- **Exemples :**
 - border-radius : 100px / 20px;
 - border-radius: 50px;/*Identique à border-radius: 50px/50px*/
 - border-radius: 25%;/*Identique à border-radius: 25%/25%*/

- **background** : nature du fond d'un élément.
 - **background-color** : une couleur (rgb, nom, hexadecimal)
 - background-color:#990066
 - **background-image** : une image (url)
 - background-image:url(image.gif)
 - **background-repeat** : répéter l'image
 - **no-repeat** : pas de répétition
 - **repeat** (-x | -y): répétition (verticale | horizontale)
- **background-attachment** : détermine comment l'image est traitée.
 - **scroll** : l'image défile avec le contenu
 - **fixed** : l'image reste figée
- **background-position** : position de l'image par rapport au coin supérieur gauche
 - **background-position : horizontal vertical**
 - background-position: 30px 50px;
 - **top, bottom, left, center, right**
 - Il est possible de combiner ces mots.
background-position: top right;

Remarques

- Les boîtes de type **bloc** accèdent à toutes les propriétés du modèle de boîte.
- Les boîtes de type **en-ligne** n'accèdent pas aux propriétés : **width, height**
 - **Note importante** : IE/win traite différemment ces propriétés qui vont être appliquées sous certaines conditions.
 - **Il est donc très important de ne pas utiliser ces propriétés sur une boîte en ligne, si son statut n'a pas été modifié par la propriété display.**

144

CSS : PROPRIÉTÉS DE POSITIONNEMENT



Propriété position

- **position** : détermine comment une boîte doit être positionné.
 - **static** : par défaut
 - **absolute**: position absolue
 - **relative** : position relative
 - **fixed**: position fixe (sauf IE5 & 6)
 - Une boîte positionnée (absolute, relative, fixed) utilise les propriétés **top**, **right**, **bottom**, **left**
 - La référence est le coin supérieur gauche qui est positionné par rapport à son plus proche parent positionné. Si aucun parent n'est lui-même positionné, c'est la boîte body qui est utilisée.

Propriétés z-index

- **z-index** : positionne des boîtes de types block positionnées (absolute, relative ou fixed) sur un axe de profondeur, représenté par un empilement (une épaisseur)
 - La propriété **z-index** permet de choisir quel élément doit apparaître au-dessus de quel ordre en donnant un index sous forme de nombre (0..n).
 - Lorsque deux éléments se chevauchent, celui possédant la plus grande valeur pour son z-index apparaîtra au-dessus de l'autre.

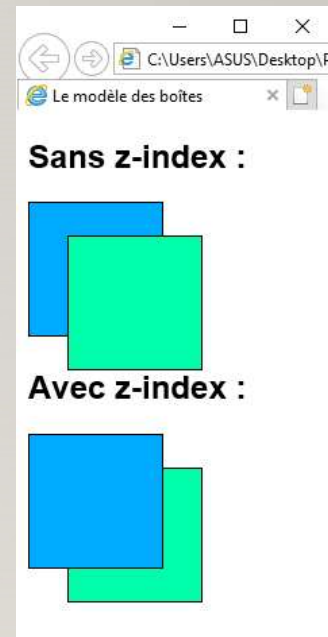
```
<body>
  <h2>Sans z-index :</h2>
  <div class="carre bleu relatif"></div>
  <div class="carre relatif vert decale"></div>

  <h2>Avec z-index :</h2>
  <div class="carre bleu relatif z100"></div>
  <div class="carre relatif vert decale"></div>
</body>
```

```
p{
  background-color: #0CC;
}
div{
  width: 100px;
  height: 100px;
  display: inline-block;
  border: 1px solid black;
  box-sizing: border-box;
}
.bleu{background-color: #0AF;}
.vert{background-color: #0FA;}

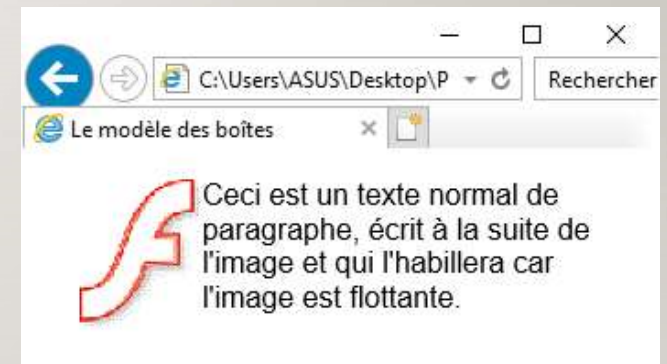
.relatif{position: relative}
.decale{
  right: 75px;
  top: 25px;
}

.z100{
  z-index: 100;
}
```



Propriété float

- **float** : faire flotter une boîte à droite ou à gauche de telle sorte que les contenus adjacents s'écoulent le long de celle-ci sur toute sa hauteur.
 - **left** : flotter à gauche
 - **right** : flotter à droite
- On doit placer l'élément flottant en premier dans le code HTML. Si on place l'image après le paragraphe, l'effet ne fonctionnera pas.



```
<body>
  <p>
    
    Ceci est un texte normal de paragraphe, écrit à la suite
    de l'image et qui l'habillera car l'image est flottante.
  </p>
</body>
```

```
.imageflottante
{
  float: left;
}
```

Propriété clear

- **clear** : interdit à une boîte d'afficher un contenu à droite ou à gauche d'elle-même.
 - **left** : pas de boîte à gauche
 - **right** : pas de boîte à droite
 - **both** : pas de boîte à droite ou à gauche

149

CSS : RESPONSIVE DESIGN

3^{ÈME} ANNÉE LICENCE EN INFORMATIQUE

FSM 2020-2021



Introduction

- Lorsqu'on parle de « responsive design », on fait référence à l'idée selon laquelle un site web devrait s'afficher aussi bien sur un écran de PC que sur un écran de smartphone ou sur n'importe quel type d'appareil.
- Pour cela, on utilise les **Media Queries**
 - Permettent d'appliquer certaines règles CSS de manière conditionnelle. Par exemple, on va pouvoir définir une largeur pour un élément pour certaines tailles d'écrans et une largeur différente pour le même élément pour d'autres tailles d'écran.



Media Types

- Dans la règle **@media**, on peut placer deux types de conditions ou de contraintes :
 - une condition sur le type du media utilisé pour afficher la page
 - une condition sur les caractéristiques du media.
- Types de médias :
 - **all** : valeur par défaut. Les règles s'appliquent à tous les appareils
 - **Screen** : Les règles s'appliquent qu'aux appareils dotés d'un écran
 - **Printer** : Les règles s'appliquent que pour les imprimantes

Media Queries

- On peut inverser la valeur logique d'un test avec le mot clef **not**. En plaçant le mot clef not avant le type de media, nos règles s'appliqueront à tous les appareils media sauf celui spécifié
- On peut créer autant de conditions sur des caractéristiques différentes que l'on souhaite. Il faut entourer chaque condition sur une caractéristique media avec un couple de parenthèses et séparer deux conditions avec les mots clefs and (**et**) ou or (**ou**).
- En pratique, nous utiliserons généralement des conditions de taille pour distinguer entre différents appareils et utiliserons pour cela les propriétés **width**, **min-width** et **max-width**.
- Sachez toutefois qu'on peut également conditionner l'application de nos styles CSS à la hauteur d'un media, sa résolution, son processus de scan, à la présence d'un appareil de pointage parmi les mécanismes de saisie et sa précision ou encore à la capacité de l'appareil à survoler les éléments.

Media Queries : Exemples

- `@media all and (max-width: 1024px) {}`
 - cette règle va cibler tous les appareils dont la largeur ne dépasse pas 1024px.
- `@media screen and (min-width: 780px) and (max-width: 979px) {}`
 - cette règle va cibler les appareils media dotés d'un écran dont la largeur est comprise entre 780px et 979px.
- `@media not screen {}`
 - cette règle va cibler les appareils media sauf ceux qui sont dotés d'un écran
- `@media print and (min-resolution: 200dpi) {}`
 - cette règle va cibler les imprimantes dont la résolution est au minimum 200dpi

Media Types, Media Queries

- Plus élégant, dans des fichiers séparés avec déclaration dans l'entête HTML

```
<link rel="stylesheet" href="style.css" media="screen and (max-width:1024px)">
```