


R1.02 - WEB - CHAP 4 - AFFICHAGE ET POSITIONNEMENT

Enseignants (S1) : Amira Ben Hadid, Nicolas Buyle-Bodin, Stéphane Leroux

Références :

- Le cours s'appuie sur le site de Pierre-Antoine Champin :
<http://champin.net/enseignement/intro-web>
- Ce travail est sous licence Creative Commons Attribution-ShareAlike 3.0

 copyright

CHANGER LE MODE D'AFFICHAGE (1/2)

Les deux valeurs principales de la propriété `display` sont :

- `inline` : l'élément apparaît dans la continuité de la ligne :
 - pas de retour à la ligne avant ou après.
 - taille non modifiable : il prend la largeur et la hauteur minimale nécessaire.
- `block` : l'élément apparaît comme une *boîte* :
 - retour à la ligne avant et après
 - prend toute la largeur disponible et la hauteur nécessaire
 - taille modifiable par les propriétés `width` et `height`

Classez ces balises de références selon leur `display` par défaut :

`<a>`, `<p>`, ``, ``, ``, ``, ``, `<section>`, `<article>`

CHANGER LE MODE D'AFFICHAGE (2/2)

Deux valeurs utiles :

- `inline-block` : mélange des deux précédents :
 - pas de retour à la ligne avant et après
 - taille modifiable par les propriétés `width` et `height`
- `none`: n'est pas affiché

Nous verrons plus tard d'autres affichages importants : flex et grid.

Quelques exemples pour comprendre.

(clic droit pour ouvrir dans une nouvelle fenêtre)

BALISES NON SÉMANTIQUES

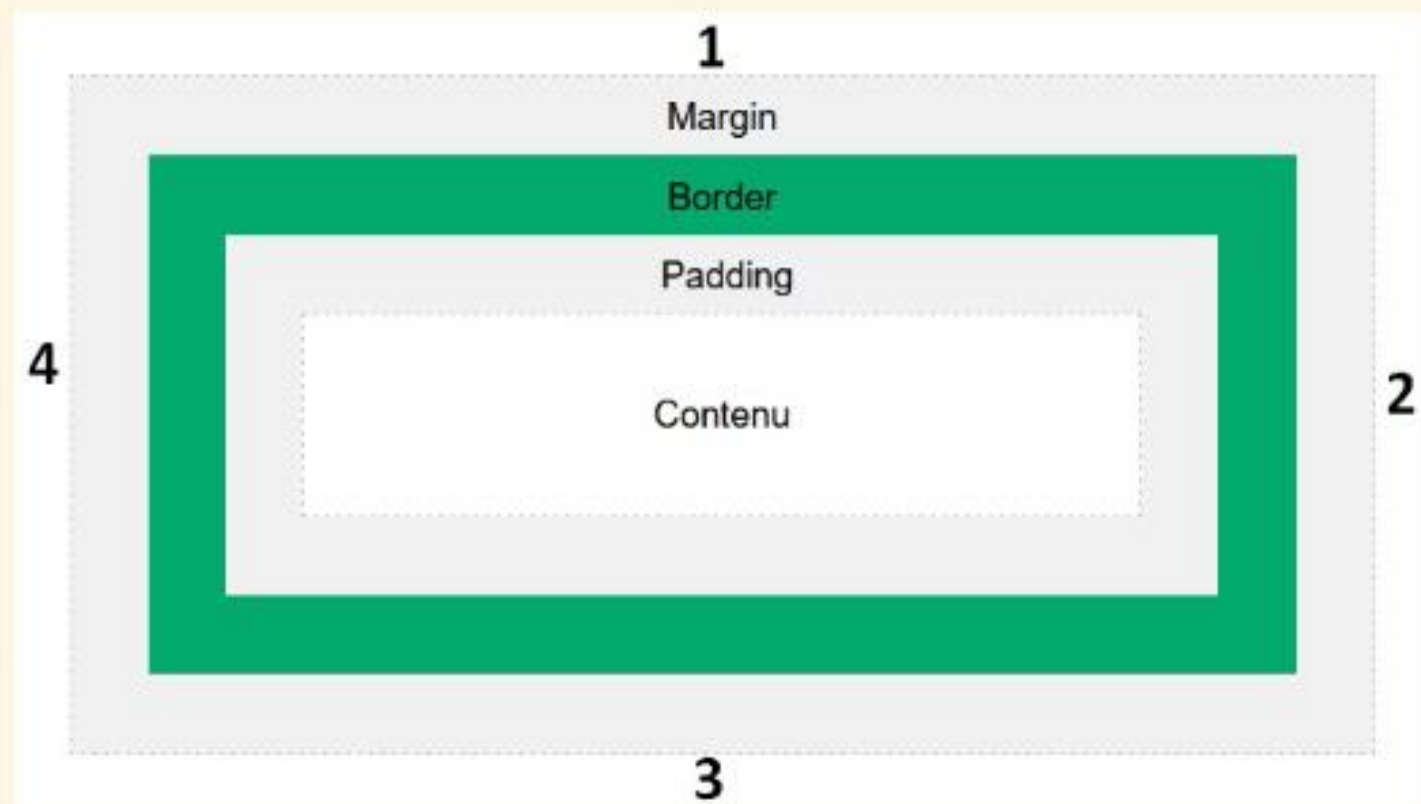
Deux balises non sémantiques peuvent être utilisées :

- `<div>` de type block
- `` de type inline.

Elles doivent toujours contenir une classe ou un identifiant pour garder une bonne qualité de code.

LES BOITES CSS

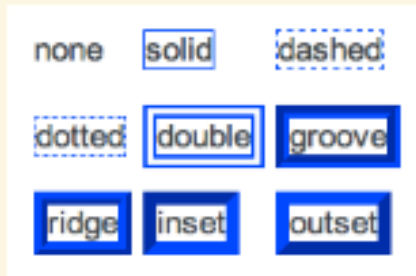
Toutes les balises peuvent être vues comme des boîtes qui entourent le contenu à afficher.



(Source de l'image : w3schools)

BORDURES (1/2)

On peut tracer un cadre autour de certains éléments avec la propriété `border` :
`border: (largeur) (style) (couleur)`



Différents styles :

On peut séparer les 3 propriétés :

- `border-width`: largeur
- `border-style`: style
- `border-color`: couleur

BORDURES (2/2)

Il est aussi possible de donner des propriétés différentes aux quatre côtés du cadre :

- `border-top`: (largeur) (style) (couleur)
- `border-right`: (largeur) (style) (couleur)
- `border-bottom`: (largeur) (style) (couleur)
- `border-left`: (largeur) (style) (couleur)

De la même manière, chacune de ces propriétés peut se décliner en sous-propriétés, par exemple :

- `border-top-width`: (largeur)
- `border-top-style`: (style)
- `border-top-color`: (couleur)

PADDING

Le padding d'un élément est l'espace entre le contenu et le bord extérieur de l'élément.

En général, on utilise un padding > 0 uniquement lorsque le bord de l'élément est visible (border, background-color).

On peut spécifier le padding par quatre propriétés (une par côté) :

- `padding-top`: (longueur)
- `padding-right`: (longueur)
- `padding-bottom`: (longueur)
- `padding-left`: (longueur)

ou par une propriété synthétique acceptant 1 à 4 valeurs :

- `padding`: (top) (right) (bottom) (left)
- si left est omis, il hérite de right
- si bottom est omis, il hérite de top
- si right est omis, il hérite de top

MARGE

La marge d'un élément est l'espace libre minimum qui doit être laissé autour d'un élément. L'espace effectif entre deux éléments voisins est déterminé en prenant la plus grande des deux marges.

Propriété margin :

- margin-top : (longueur)
- margin-right : (longueur)
- margin-bottom : (longueur)
- margin-left : (longueur)

On peut aussi utiliser la version synthétique, qui accepte 1 à 4 valeurs :

margin : (top) (right) (bottom) (left)

- si left est omis, il hérite de right
- si bottom est omis, il hérite de top
- si right est omis, il hérite de top

Quelques exemples pour comprendre.

(clic droit pour ouvrir dans un nouvel onglet)

EXERCICE

Tester cet exemple pour [centrer horizontalement](#)
(clic droit pour ouvrir dans une nouvelle fenêtre)

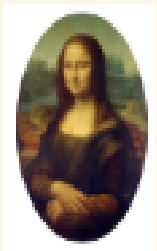
COINS ARRONDIS

Il est possible d'arrondir les coins d'un élément grâce à la propriété `border-radius`:

- avec une seule valeur : tous les coins sont concernés
- avec 4 valeurs dans cet ordre : (top-left) (top-right) (bottom-right) (bottom-left)

Il est possible de préciser pour chaque coin séparément `border-top-left-radius`: (rayon) ...

On peut également exprimer le rayon en pourcentage de la taille de l'élément. Avec une valeur de 50%, on obtient une ellipse :



(source : Wikipedia)

IMAGES FLOTTANTES ET ALIGNEMENT VERTICAL

Par défaut, les images appartiennent au flux du texte.

```
<p>La Joconde   
est probablement le tableau le plus connu au monde.</p>
```



Il est possible de préciser dans les règles CSS de `img` :

- la position relative de l'image avec `float` : left, right, none...
- la position verticale du texte avec `vertical-align` : baseline, top, middle, text-top...

Vous pouvez tester sur [un autre exemple](#)

EXERCICE

Tester cet exemple pour [centrer verticalement](#)
(clic droit pour ouvrir dans une nouvelle fenêtre)

IMAGE OU IMAGE DE FOND ?

Il convient de distinguer :

- les images qui font partie intégrante du document (comme les figures) doivent être insérées dans le document par une balise ``,
- les images liées à la présentation doivent être incluses dans la feuille de style via la propriété `background`.

On peut appliquer un fond à n'importe quel élément.

IMAGES DE FOND (1/4)

Avant de présenter les propriétés en détail, essayons de comprendre ce code.

A quoi va ressembler l'arrière plan du site avec ce code CSS ?

```
body {  
  background-image: url("sun_reduit.webp");  
  background-attachment: fixed;  
  background-position: top right;  
  background-repeat: no-repeat;  
  background-color: blue;  
  background-image: url("moon.png");  
  background-attachment: scroll;  
  background-repeat: repeat-y;  
  background-position: top left;  
}
```

Analyser la réponse

IMAGES DE FOND (2/4)

Présentation des propriétés :

- background-image: (url)
- background-position: (position gauche) (position haut)
 - positions en pixels
 - positions en anglais (ex: top left)
- background-attachment:
 - fixed si l'image de fond reste fixe
 - scroll si l'image défile (par défaut)
- background-repeat:
 - no-repeat si l'image est unique
 - repeat-x si l'image est répétée sur la première ligne
 - repeat-y si l'image est répétée sur la première colonne
 - repeat si l'image est répétée en mosaïque (par défaut)

IMAGES DE FOND (3/4)

Les propriétés précédentes peuvent se combiner dans une seule propriété avec plusieurs valeurs :

`background: (v1) (v2) (v3) (v4) (v5) ;`

- `v1` : `background-color`
- `v2` : `background-image`
- `v3` : `background-repeat`
- `v4` : `background-attachment`
- `v5` : `background-position`

Une des valeurs peut manquer mais l'ordre doit être respecté.

IMAGES DE FOND (4/4)

On peut également combiner plusieurs images de fond dans une seule propriété `background` en séparant ces images et leurs options respectives par une virgule.

Que se passe-t-il pour cette mise en page ?

```
body {  
    background: url("sun.png") no-repeat fixed top right,  
               url("lune.png") repeat-y top left,  
               blue;  
}  
h1 {  
    text-align: center;  
}
```

Voir le [résultat](#) ou sur le [site du cours](#)

COMPLEMENTS

Pour aller plus loin

Les diapositives suivantes présentent des compléments au cours.

Les contenus ne sont pas exigibles en examen

Elles peuvent servir sur des exercices plus avancés comme le TP de synthèse.

COMPLÉMENT : LE POSITIONNEMENT FIN

Les display Grid et flexbox remplacent progressivement les anciennes méthodes, mais il peut être utile de connaître la propriété `position`.

La propriété `position` admet les valeurs :

- `static` : elle suit le flux normal du texte. C'est la valeur par défaut.
- `relative` : elle se positionne relativement à l'endroit où elle devrait être selon le flux normal. Il faut préciser sa position avec d'autres propriétés : `top`, `left`
- `absolute` : L'élément est retiré du flux normal. Sa position est liée à son parent.
- `sticky` : elle est static jusqu'à atteindre une position où elle devient fixe. Bien utile pour maintenir élément visible après un scroll. Uniquement besoin de `top`.
- `fixed` : L'élément est fixe par rapport à l'écran de l'utilisateur.

Exemples : <https://devdocs.io/css/position>