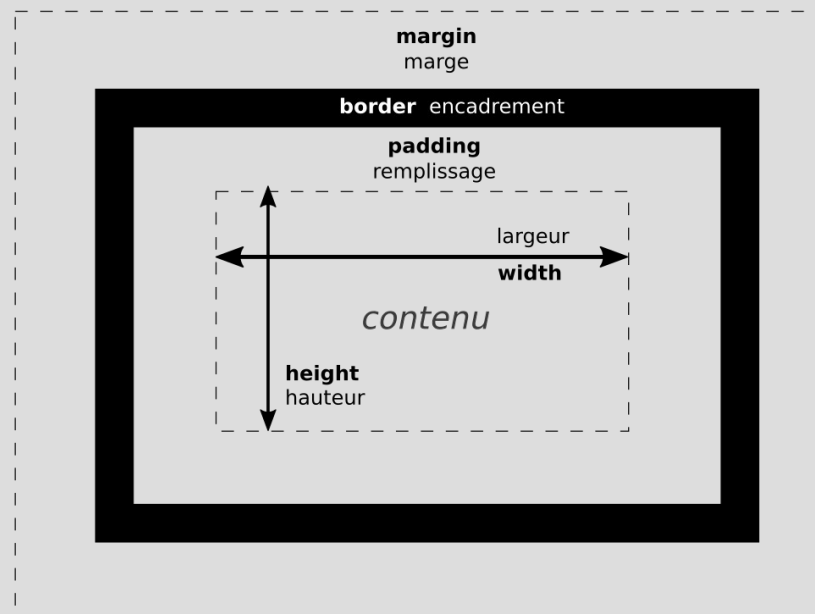


# CSS3 : mise en page

- Modèle des boîtes
- Dimensions
- Remplissage, Marges, Bordures
- Techniques de mise en page
- Positionnement
- Dépassement
- Inline-block et tableaux

# Modèle des boîtes

- Le modèle des boîtes en CSS est le concept de base de la **mise en page sur le Web**
- Chaque élément est représenté comme une **boîte rectangulaire**, avec un contenu, un remplissage, un encadrement et une marge



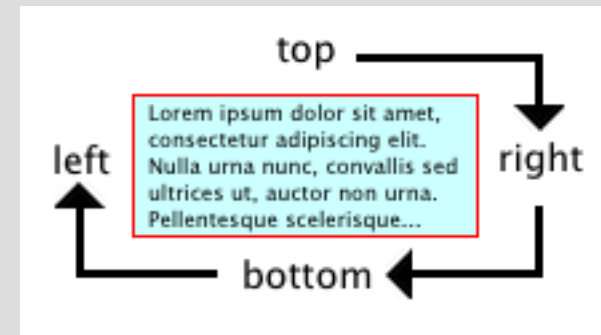
# Dimensions

- Avant de placer les différents éléments dans la mise en page, on doit définir leurs **dimensions**
- Le **contenu** se trouve dans une boîte avec :
  - Une hauteur définie par : **height**
  - Une largeur définie par : **width**
- On peut aussi donner des **contraintes** (valeurs minimum et maximum)
  - **min-width** et **max-width**
  - **min-height** et **max-height**

# Remplissage (padding)

- Dans le modèle des boîtes, on trouve ensuite un **espacement** autour du contenu mais à l'intérieur d'une bordure
- On peut avoir des **valeurs différentes** pour les espacements en haut, en bas, à gauche et à droite avec les propriétés :

**padding-top**  
**padding-right**  
**padding-bottom**  
**padding-left**

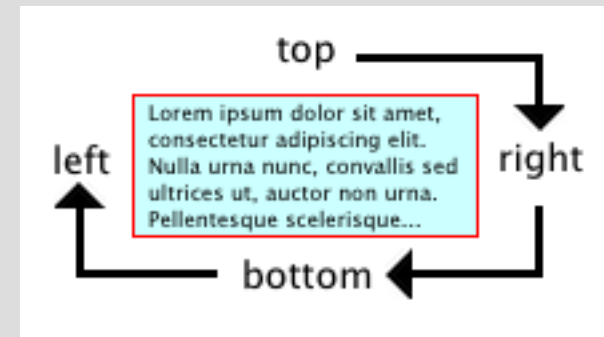


- On peut les définir en une fois avec **padding**

# Bordure (border)

- Dans le modèle des boîtes, on trouve ensuite une **bordure** autour du contenu et de son éventuel espacement intérieur
- On peut avoir des **valeurs différentes** pour les bordures en haut, en bas, à gauche et à droite avec les propriétés :

**border-top**  
**border-right**  
**border-bottom**  
**border-left**



- On peut les définir en une fois avec **border**

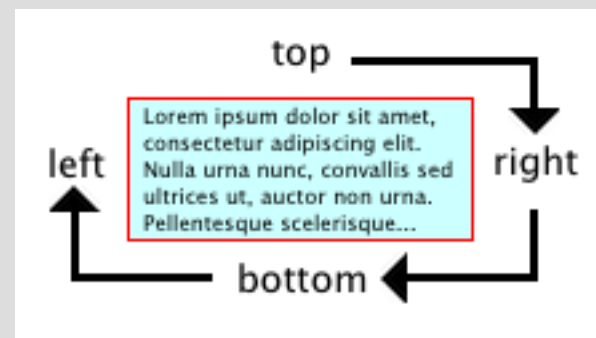
# Propriétés des bordures

- Une bordure peut être définie par son **épaisseur**, son **style** et sa **couleur**.
- Ces propriétés peuvent être aussi regroupées :
  - Par la position : **border-left**, **border-right**, **border-top**, **border-bottom**
  - Par les caractéristiques : **border-width**, **border-style**, **border-color**
- Ou séparées :  
**border-top-width**, **border-top-style**,  
**border-top-color**, **border-right-width**,...

# Marge (margin)

- Dans le modèle des boîtes, on trouve enfin la **marge** autour de la bordure mais à l'extérieur
- On peut avoir des **valeurs différentes** pour les espacements en haut, en bas, à gauche et à droite avec les propriétés :

**margin-top**  
**margin-right**  
**margin-bottom**  
**margin-left**



- On peut les définir en une fois avec **margin**

# Techniques de mise en page

- Dès que tous les éléments ont été définis, il faut les placer comme dans un **puzzle** pour reconstituer l'ensemble de la page
- On va réaliser le **positionnement** des éléments en employant différentes techniques
- Les éléments HTML se comportent principalement selon :
  - Un modèle de boîte de type bloc (**block**)
  - Un modèle de boîte de type en-ligne (**inline**)



# Modèles de boîte

- Chaque élément HTML suit un **modèle de boîte** et de **positionnement** par défaut
  - **Block** pour <p>, <div>, <h1>-<h6>,...
  - **Inline** pour <a>, <span>, <img>, <em>,...
  - **Table** pour <table>, <tr>, <td>,...
- Selon le type de modèle, **certaines propriétés sont acceptées et d'autres non** (width, height, padding, margin, border, vertical-align,...)
- Il y a aussi : display, position, float, clear,...

# Types de positionnement

- Il y a plusieurs types de positionnement :
  - Positionnement dans le **flux**
  - Positionnement **absolu**
  - Positionnement **fixé**
  - Positionnement **relatif**
  - Positionnement **flottant**
- **Par défaut**, il s'agit d'un positionnement statique (position:static)

# Rendu et flux courant

- L'ordre dans lequel apparaissent les balises dans le code est celui dans lequel les boîtes sont affichées dans le document, c'est le **flux courant**
- Chaque élément :
  - Est situé sur le **même plan** que les autres
  - Se place **le plus haut et le plus à gauche** possible au sein de son parent
  - Est **dépendant de l'élément frère précédent** (Les *block* s'empilent, les *inline* se suivent)
  - Peut être **modifié** avec la propriété **display**

# Positionnement absolu

- Un élément est positionné en absolu avec la propriété **position:absolute**;
- L'élément est alors "**sorti du flux**" et se retrouve dans un autre plan (au-dessus du flux, à régler avec **z-index**)
- Les **autres éléments se repositionnent** comme si l'élément positionné n'existait plus dans le flux
- On place l'élément absolu avec les propriétés **top**, **right**, **bottom** et **left** par rapport à son premier ancêtre positionné

# Profondeur : z-index

- Tous les éléments de la page se positionnent dans l'écran **en x (axe horizontal vers la droite)** et **en y (axe vertical vers le bas)**
- L'**axe z** est un axe perpendiculaire sortant de l'écran qui représente l'empilement des éléments (comme les calques dans Photoshop)
- Pour la **couche de base, z-index vaut 0**
- Plus la valeur de **z-index augmente** et plus les couches s'empilent, **la plus grande valeur est donc tout au-dessus**

# Positionnement fixé

- Un élément est positionné en fixé avec la propriété **position:fixed**;
- **Comme en absolu**, l'élément est "sorti du flux", il se retrouve dans un autre plan et les autres éléments se repositionnent entre eux
- L'élément fixé **ne bouge pas** dans l'écran **même si on fait défiler** le contenu avec les scrollbars
- L'élément est fixé avec top, right, bottom et left **par rapport à la fenêtre du navigateur**

# Positionnement relatif

- Un élément est positionné en relatif avec la propriété **position:relative**;
- Les propriétés top, right, bottom et left indiquent un **décalage** et **pas un emplacement**
- L'élément **reste dans le flux**, les autres éléments se repositionnent
- L'élément relatif se place **par rapport à sa position classique** dans le flux, éventuellement décalé, et l'espace libéré est bloqué pour d'autres éléments

# Positionnement flottant

- A l'origine, le **flottement** sert à pousser, à droite ou à gauche, du texte ou une image, et à remplir l'espace libre avec le reste du contenu
- Cet usage a été **détourné pour positionner** des éléments, mais il peut varier selon le navigateur d'où des erreurs d'affichage
- Un élément flottant est sorti du flux et placé à l'extrême gauche (**float:left**) ou droite (**float:right**) de son conteneur tout en restant sur sa hauteur de ligne initiale, le reste s'écoule autour.



# Des blocs côte à côte

- Un élément **flottant** prend par défaut la **largeur de son contenu**. Si le contenu est important, il vaut mieux fixer avec `width` ou `max-width`
- S'il y a un **2ème élément flottant**, il reste sur le même plan et donc se place à ses côtés
- Attention, les flottants sortent du flux, donc le parent sera vide et le contenu va dépasser. Il faut donc **fixer la hauteur** du parent ou placer un **élément non flottant** après

# Annuler le flottement

- La propriété **clear** interdit à un élément de se placer sur la même ligne qu'un bloc flottant et le force à passer en-dessous
- On annule les flottants à gauche (**clear:left**) ou à droite (**clear:right**) ou les deux (**clear:both**)
- **clear** permet, d'empêcher les dépassements de flottants, ou de placer un élément **toujours en bas** du flottant le plus long

# Dépassement

- La propriété **overflow** définit la méthode de dépassement des textes de leur conteneur.
  - **visible** : le contenu qui sort est visible.
  - **hidden** : le contenu est rogné aux limites du conteneur, et pas de barre de défilement
  - **scroll** : le contenu est rogné aux limites du conteneur, et des barres de défilement
  - **auto** : le choix est laissé aux navigateurs.
- CSS3 étend ce mécanisme en proposant les propriétés, **overflow-x** et **overflow-y**

# Inline-block

- Avec la propriété **display:inline-block**; on peut combiner les avantages des 2 types principaux
  - Inline : des blocs peuvent se placer l'un à côté de l'autre et rester verticalement alignés
  - Block : pouvoir être dimensionnés
- Combiné avec la propriété **vertical-align**, on peut régler l'alignement des blocs selon la valeur (baseline, top, bottom, middle,...)

# Rendu de tableau en CSS

- Il est possible de faire une **mise en page** sous forme de **tableau**.
- Pas avec `<table>`, `<tr>`, `<td>` !
- Mais avec des valeurs spéciales pour **display**
  - **table, table-row, table-cell**
  - **table-header-group, table-footer-group**
  - **table-caption,...**