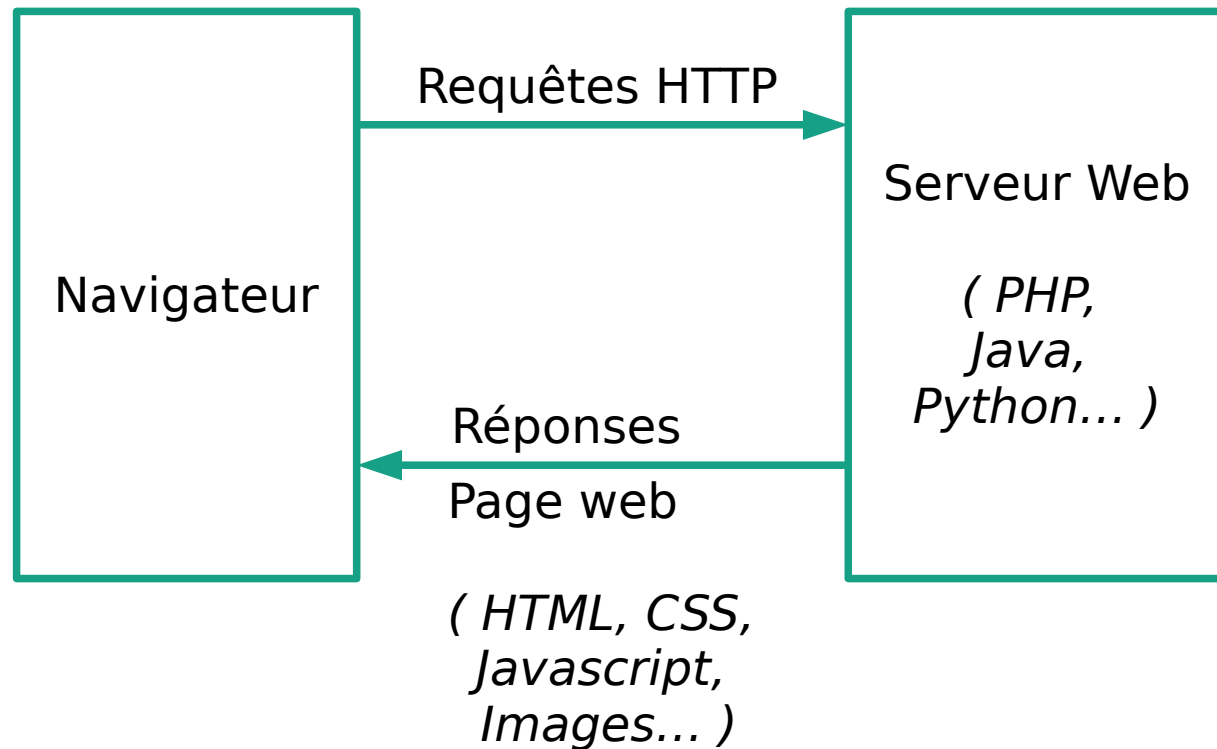


# CSS Avancé

CSS Avancé  
Nathanaël Martel

# Architecture *n-tiers*



# ***HTML / CSS / javascript***

- **Ce sont des langage normé**
- **Ils sont interprété (exécuté) par le navigateur**
- **Les normes évoluent et tous les navigateurs ne respectent pas complètement les normes**
  - Un outil comme [caniuse.com](https://caniuse.com) vous permet de vérifier le support de tel ou tel fonctionnalité

# HTML / CSS

- **Séparation du fond et de la forme**
- **HTML :**
  - Le fond
  - Le sens, et donc l'utilisation de balise qui ont du sens
- **CSS :**
  - La forme
  - L'apparence
  - Le comportement

- **Pourquoi ?**

- Créé et maintenu par des personnes différentes
- La forme dépend du medium :
  - Écran large
  - Mobile
  - Télévision
  - Montre
  - Imprimante
  - Vocalisation
  - Tablette braille
  - Robot

# HTML / DOM

- **HTML**

- Code source qui est envoyé par le serveur
- Clic droit / afficher la source

- **DOM**

- Structure généré par le navigateur en fonction du code source
- Arbre avec des nœuds, des parents, des enfants...
- Le navigateur y attache le CSS
- Clic droit / examiner (inspecter) pour visualiser et debugger
- Peut être modifié en javascript

# ***HTML / Flux***

- **La mise en place des différents éléments de la page se fait par défaut selon le Flux courant.**
- **Les éléments (balises) sont placés les uns après les autres dans le code HTML de la page.**
- **L'ordre dans lequel apparaissent les balises dans le code HTML sera l'ordre dans lequel ils seront affichés et apparaîtront dans le document, c'est le Flux.**

# Lien HTML / CSS

- **Indiquer la feuille de style**
  - `<link href="/assets/css/style.css" type="text/css" />`
  - Il est possible d'en mettre plusieurs
- **Utiliser la balise `<style>`**
  - À éviter un maximum
- **Utiliser l'attribut `style=""`**
  - À éviter un maximum



# Rappel : sélecteurs

- **\*** → all elements
- **div** → all div tags
- **div,p** → all divs and paragraphs
- **div p** → paragraphs inside divs
- **div > p** → all p tags, one level deep in div
- **div + p** → p tags immediately after div
- **div ~ p** → p tags preceded by div
- **.classname** → all elements with class
- **#idname** → element with ID
- **div.classname** → divs with certain classname
- **div#idname** → div with certain ID
- **#idname \*** → all elements inside #idname

# Rappel : sélecteurs, pseudo classes

- **\*** → all elements
- **a:link** → link in normal state
- **a:active** → link in clicked state
- **a:hover** → link with mouse over it
- **a:visited** → visited link
- **p::after{content:"yo";}** → add content before p
- **p::before** → add content after p
- **div:empty** → element with no children
- **p::first-letter** → first letter in p
- **p::first-line** → first line in p
- **p:first-of-type** → first of some type
- **p:last-of-type** → last of some type
- **p:lang(en)** → p with en language attribute
- **:not(span)** → element that's not a span

# Rappel : sélecteurs, pseudo classes

- **p:first-child** → first child of its parent
- **p:last-child** → last child of its parent
- **p:nth-child(2)** → second child of its parent
- **p:nth-child(3n+1)** → nth-child ( $an + b$ ) formula
- **p:nth-last-child(2)** → second child from behind
- **p:nth-of-type(2)** → second p of its parent
- **p:nth-last-of-type(2)** → ...from behind
- **p:only-of-type** → unique of its parent
- **p:only-child** → only child of its parent
- **:root** → documents root element
- **::selection** → portion selected by user
- **:target** → highlight active anchor

# *Rappel : sélecteurs, pseudo classes*

- **input:checked** → checked inputs
- **input:disabled** → disabled inputs
- **input:enabled** → enabled inputs
- **input:focus** → input has focus
- **input:in-range** → value in range
- **input:out-of-range** → input value out of range
- **input:valid** → input with valid value
- **input:invalid** → input with invalid value
- **input:optional** → no required attribute
- **input:required** → input with required attribute
- **input:read-only** → with readonly attribute
- **input:read-write** → no readonly attrib.

# Rappel : sélecteurs, attributs

- `input:checked` → checked inputs
- `a[target]` → links with a target attribute
- `a[target="_blank"]` → links which open in new tab
- `[title~="chair"]` → title element containing a word
- `[class^="chair"]` → class starts with chair
- `[class|="chair"]` → class starts with the chair word
- `[class*="chair"]` → class contains chair
- `[class$="chair"]` → class ends with chair
- `input[type="button"]` → specified input type

# Priorités des sélecteurs

## Quel est la couleur du texte ?

```
<div id="test">  
  <span>Text</span>  
</div>
```

```
div#test span { color: green; }  
div span { color: blue; }  
span { color: red; }
```

# *Priorités des sélecteurs*

- **Balise** → **1**
- **Class** → **1 0**
- **Id** → **1 0 0**
- **Inline** → **1 0 0 0**

**Les pseudo-classes compte comme les classes**

# *Priorités des sélecteurs*

- `#alpha.beta.gamma` → 0 1 2 0
- `ul.maliste li` → 0 0 1 2
- `ul#maliste li` → 0 1 0 2



# *Priorités des sélecteurs*

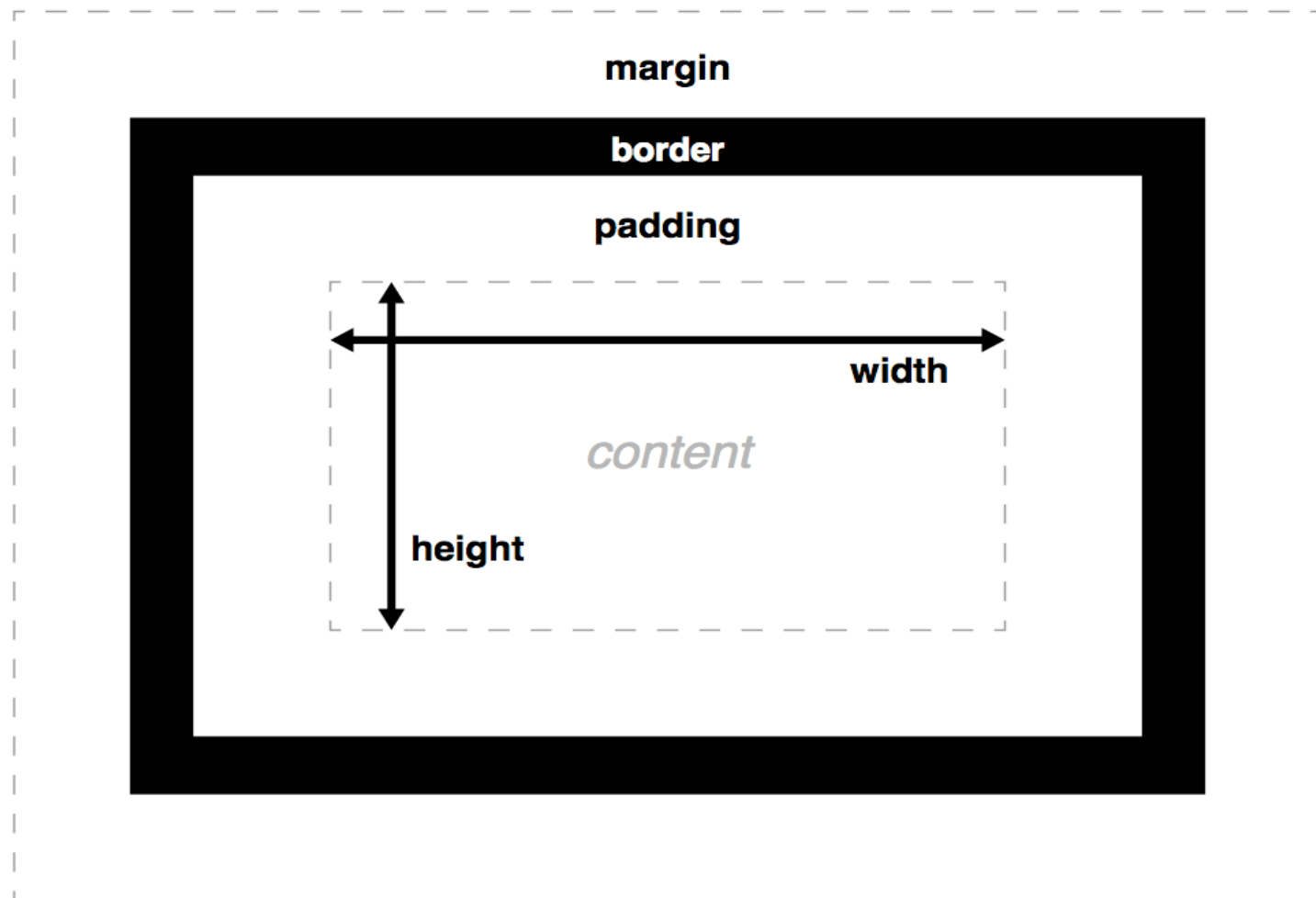
- **Si le poids est égales, la dernière déclaration écrase les précédentes**
- **Le mot clé « !important » sur une propriété ne peut être écrasé**
- **Dans la pratique :**
  - **Utiliser des classes**
  - **Préciser avec des balises**

# *Typographie*

```
font-family: Ubuntu, sans-serif;  
font-size: 1em;  
letter-spacing: 0;  
word-spacing: 0;  
color: #000000;  
font-weight: normal;  
text-decoration: none;  
font-style: normal;  
font-variant: normal;  
text-transform: none;
```

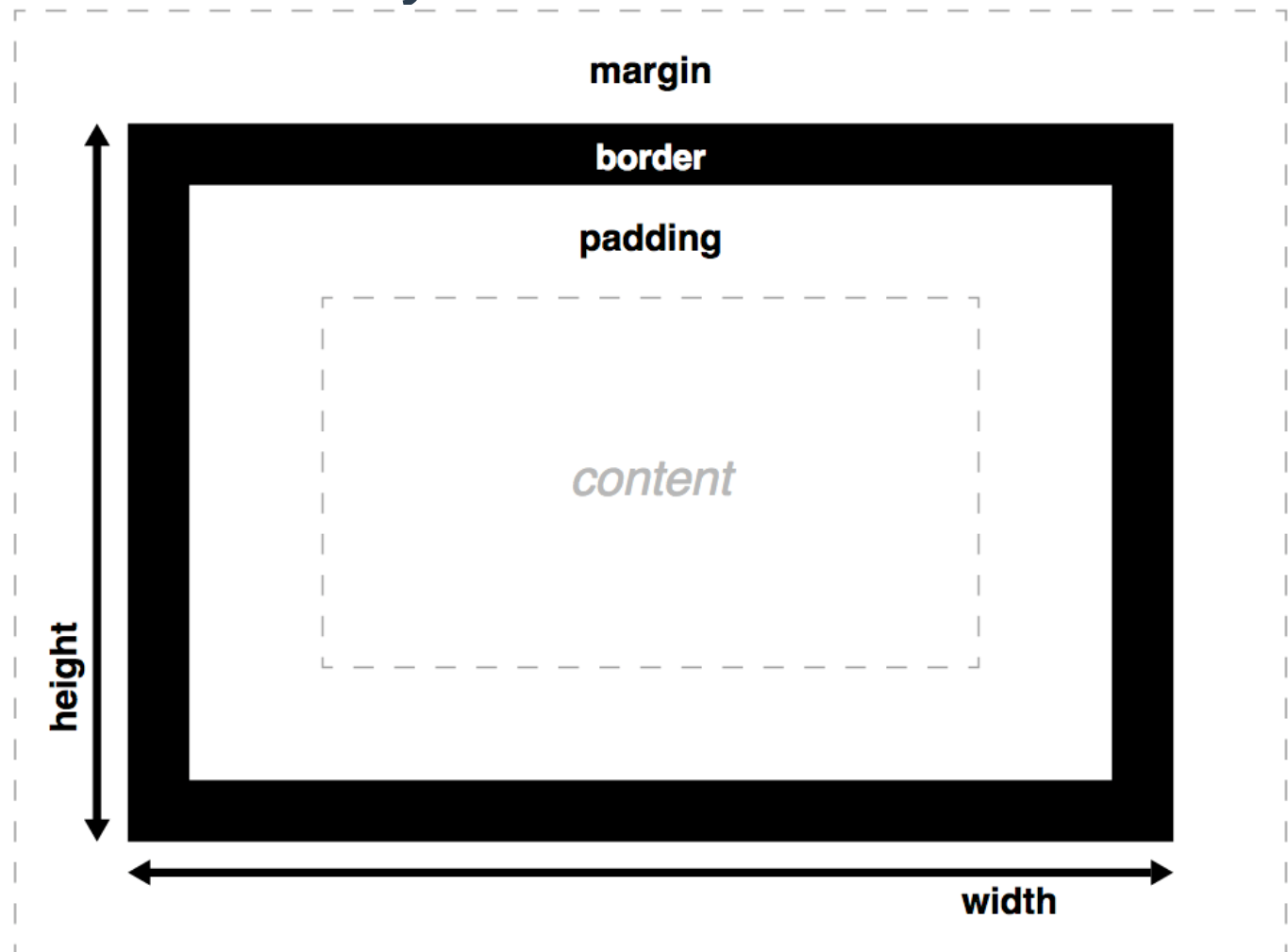
# Modèle de boîte

## Box-sizing: content-box; (default)



# Modèle de boîte

Box-sizing: border-box;



# *Propriété display: block vs inline*

## La balise <div>

- Élément neutre d'un point de vue sémantique
- De type « block » (`display:block`)
- Prend la largeur de l'espace disponible
- S'adapte à la hauteur de son contenu
- Est positionné dans le flux, **en dessous** de l'élément suivant

# *Propriété display: block vs inline*

## Éléments Block

- **body, section, nav, article, ul, ol, p, h1...**
- **Il est facile de les manipuler :**
  - **Padding**
  - **Margin**
  - **Height, max-height, min-height**
  - **Width, max-width, min-width**
  - **border**

# *Propriété display: block vs inline*

## La balise <span>

- Élément neutre d'un point de vue sémantique
- De type « inline » (`display:inline`)
- Prend la largeur du texte qu'il contient, et passe à la ligne pour ne pas dépasser la largeur disponible
- S'adapte à la hauteur de son contenu
- Est positionné dans le flux, **à la suite** de l'élément suivant, la hauteur de ligne s'adaptant à l'élément précédent

# *Propriété display: block vs inline*

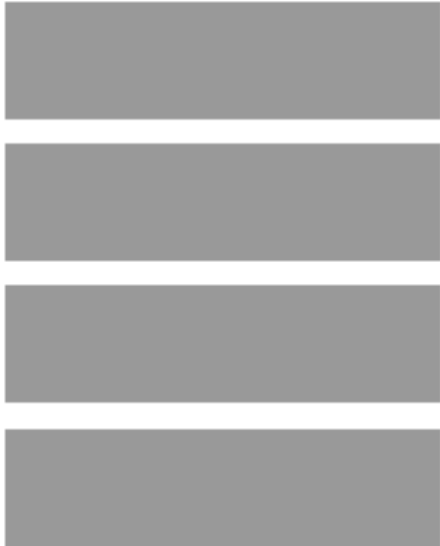
## Éléments span

- Em, strong, i, b, abbr, a, img...



# *Propriété display: block vs inline*

## **BLOCK:**



## **INLINE:**



# Propriété display: block vs inline

## Inline vs Block

### Inline



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.

`<img>`  
`<span>`  
`<strong>`  
`<em>`

### block

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

List of the best pies

- Apple
- Cherry
- Pumpkin

`<p>`  
`<div>`  
`<li>`  
`<h1>`

# Unités

## Les unités absolue :

- **px**
  - pixel
- **mm, cm, in**
  - millimètres, centimètres et pouces (inches).
  - Utile pour l'impression
- **pt, pc :**
  - points (1/72 de pouce) ou picas (12 points.)

## Les unités relatives :

- %
  - Pourcentage relatif à...
- em
  - 1em est identique à la taille de police de l'élément actuel (La largeur du M majuscule)
  - C'est l'unité la plus courante
- rem :
  - Idem que rem, mais relative à la valeur par défaut

## Les unités relatives :

- vw, vh
  - Viewport width, viewport height
  - Un centième de la largeur (hauteur) de la vue

# Unités

- **0 (zéro) n'a pas besoin d'unité**
  - `0 = 0px = 0em...`
- **Line-height:1.5 ;**
  - Si la propriété line-height est donné sans unité, elle correspond à un coefficient multiplicateur de la taille de la police
  - Exemple : `Font-size:16px;line-height:1.5;` → `line-height:24px;`

# Tableau

```
table {  
    display:table ;  
    border-collapse: separate ;  
    // collapse  
    border-spacing: 2px;  
    table-layout: auto;  
    // fixed  
    width:100%;  
}
```

```
tr {  
    display:table-row;  
}
```

```
th, td {  
    display:table-cell;  
    border: 1px solid black;  
    vertical-align : middle ;  
    // bottom, top  
}
```

```
tr:nth-child(2n+1) {  
    background-color: #f2f2f2;  
}
```

# ***Interface fluide***

**La mise en forme s'étire et se déforme avec les dimensions de l'écran**

**Pour y parvenir, nous utilisons ces propriétés :**

- **max-width, max-height**
- **vw, vh, %**



# ***Interface responsive***

**La mise en forme et modifié en fonction des dimensions de l'écran (et des spécificités du media)**

**Pour y parvenir, nous utilisons les media query**

# Interface responsive

## Choix du media

- Dans le CSS :

```
@media print {  
    #menu, #footer, aside {  
        display:none;  
    }  
}
```

- Import

- @import url("fineprint.css") print;

- Dans le HTML :

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

# *Interface responsive*

## Règles disponible :

- width, min-width, max-width
- height, min-height, max-height
- media
  - screen, speech, print, all
- orientation
  - portrait, paysage
- Autres :
  - aspect-ratio, resolution, scan, grid, update, color, color-gamut, color-index, display-mode, inverted-color, pointer, hover...

# *Interface responsive*

## Séparateurs

- **And**
  - `@media (min-width: 30em) and (orientation: landscape) { ... }`
- **or**
- **,**
  - `@media (min-height: 680px), screen and (orientation: portrait) { ... }`
- **not()**

# *Interface responsive*

```
<style>
.test {
    background:red;
    height:80vh;
}
@media (min-width: 30em) {
    .test {
        background:green;
    }
}
</style>
<div class="test"></div>
```

# Variable CSS

- **Déclaration**

- élément {  
    --main-bg-color: brown;  
}

- **Utilisation**

- élément {  
    background-color: var(--main-bg-color);  
}

# Variable CSS

## Exemple :

```
:root {  
  --main-bg-color: brown;  
}  
  
.un {  
  background-color: var(--main-bg-color);  
}  
  
.trois {  
  background-color: var(--main-bg-color);  
}
```

# *Variable CSS*

- **Attention à l'héritage, il faut que la variable soit déclaré dans sélecteur parent pour être disponible**
- **Attention au support internet explorer**
- **Évidement, il est possible de combiner avec du responsive**



# ***Positionnement CSS***

**Voici un aperçu des méthodes disponible**

- **Table**
- **Absolute**
- **Float**
- **Flex**
- **Grid**