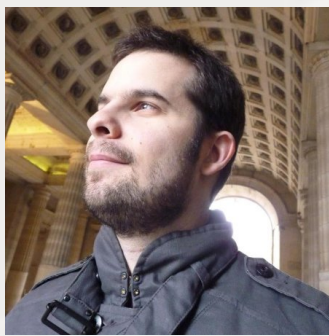
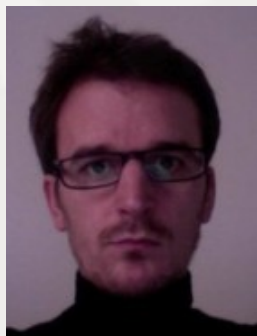


CSS3

Cascading Style Sheets

Auteurs



Erwan Fournel

Gabriel Block

L'ÉCOLE DE FORMATION PROFESSIONNELLE CRÉÉE PAR DES INFORMATICIENS.

Créée en 1994, l'école de la filière numérique est actuellement implantée sur Nantes, Angers et Rennes. IMIE vous propose d'obtenir des diplômes reconnus, et de maîtriser les technologies les plus récentes.

Feuille de style

■ Qu'est-ce?

- Ensemble de mises en forme génériques (position des éléments, aspect des textes à afficher, ...) associés à des balises.
- Norme actuelle : CSS (Cascading Style Sheets)
- Peut être enregistrée dans un fichier séparé de la page Web, et ainsi être utilisée par plusieurs pages.
- Déclaration de la feuille de style ou de son utilisation dans l'entête (`< head > ... </ head >`)

Feuille de style

■ Syntaxe :

■ Balise {

propriété: valeur;

[propriété: valeur; ...]*

}

```
selector{  
  property: value;  
}
```

■ Exemple :

- Le texte contenu dans toutes les balises <p> aura une taille de 10px et une couleur bleue.

```
p{  
  font-size: 10px;  
  color: blue;  
}
```

Feuille de style

■ Définition locale (déconseillée) :

```
<p style="margin-left: 10px; margin-right: 5px; color: #0000FF;">
```

■ Définition globale (dans l'entête <head></head>) :

```
<style type="text/css">  
p {  
    color: red;  
}  
</style>
```

■ Définition externe :

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

- l'URL référence un fichier qui contient le CSS

Définitions de styles : polices

■ font-size : taille de la police.

- une taille relative : xx-small, x-small, small, medium, large, x-large, xx-large, larger, smaller ou en cadratin (em)
- une taille absolue en pixels (px)

■ font-weight : graisse

- normal, bold, bolder, lighter
- ou une valeur numérique comprise entre 100 et 900

Définitions de styles : polices

■ font-style : style d'écriture

- normal, italic, oblique

■ font-family : police

- serif, sans-serif, cursive, fantasy, monospace, ...
- ou police précise

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

- Si la première police n'est pas présente, prend la suivante.

Définitions de styles : polices

■ line-height: définit la hauteur de ligne

■ en valeur fixe, en valeur relative, en pourcentage

Définitions de styles : Webfonts

- @font-face existe depuis CSS 2.0. Mais seul Internet Explorer l'implémentait depuis sa version 4.
- Les différents formats de la police de caractères doivent être placés sur le serveur.
- La police sera chargée au moment de l'appel de la feuille.

```
@font-face {  
    font-family: myFirstFont;  
    src: url(fonts/LinLibertine_R.ttf);  
    font-weight: normal;  
}  
  
div {  
    font-family: myFirstFont;  
}
```

Définitions de styles : Webfonts

■ Il existe différents formats de police :

- .ttf: TrueType Font. IE9 et autres navigateurs.
- .eot: Embedded OpenType. IE uniquement (toutes versions)
- .otf: OpenType Fond. Tous les navigateurs sauf IE
- .svg: SVG Fonts. Safari (ios)
- .woff: Web Open Font Format. IE9 et autres navigateurs.

Définitions de styles : Webfonts

■ Charger plusieurs formats (compatibilité) :

```
@font-face {  
  font-family: 'myFirstFont';  
  src: url('fonts/LinLibertine_R.eot') format('eot'),  
       url('fonts/LinLibertine_R.woff') format('woff'),  
       url('fonts/LinLibertine_R.ttf') format('truetype'),  
       url('fonts/LinLibertine_R.svg') format('svg');  
}  
  
div {  
  font-family: myFirstFont;  
}
```

Définitions de styles : fond

■ background-color : couleur du fond

- #rrvvbb (codage classique)
- valeur symbolique
- rgb(0,0,0) / rgba(0,0,0,0.5)

■ background-image : image de fond

```
body{background-image : url('img/mon_image.png')};
```

■ background-repeat : répétition de l'image de fond

- no-repeat | repeat-x | repeat-y

Définitions de styles : fond

■ background-position : position de l'image de fond

■ top | bottom | left | right | center

■ background-attachment : réaction de l'image au scroll

■ fixed | scroll

■ background : combinaison des propriétés

```
body{
  background: url("image1.png") fixed no-repeat top left;
}
```

■ plusieurs images (1ère au dessus):

```
background: url("image1.png") fixed no-repeat top left, url("image2.png") fixed;
```

Définitions de styles : transparence

■ opacity : transparence de l'élément

- valeur entre 0 (totalement transparent) et 1 (opaque)
- utilisable à partir de IE9 (comme rgba)

Définitions de styles : bordures

■ **border-width:** largeur d'une bordure

■ en pixels (px)

■ **border-color:** couleur d'une bordure

■ valeur symbolique, valeur hexadécimale ou RGB

■ **border-style:** type de bordure

■ none | solid | dotted | dashed | double | groove | ridge | inset | outset

■ **border :** combinaison des 3 propriétés

Définitions de styles : bordures

■ Exemple :

```
table{
  border: 2px solid black;
}
```

■ Il est également possible de ne styler qu'un bord :

- border-top | border-left | border-bottom | border-right
- Ces propriétés sont des combinaisons des propriétés border-top-color, border-top-width, ...

Définitions de styles : bordures

■ border-radius : arrondi des coins (IE9)

- 1 valeur : tous les coins sont arrondis de la même façon
- 4 valeurs : décrit dans l'ordre les coins haut, droite, bas et gauche

```
div{  
    border-radius: 5px;  
}
```

```
div{  
    border-radius: 5px 10px 5px 10px;  
}
```

Définitions de styles : ombres

■ box-shadow : ombre d'une box (6 valeurs possibles)

- la position horizontale (peut être négative)
- la position verticale (peut être négative)
- le flou (optionnel)
- la taille (optionnel)
- la couleur (optionnel sauf Safari)

```
div{  
    box-shadow: 5px 4px 5px 0px rgba(135,35,125, 0.2);  
}
```

- l'ajout de la valeur 'inset' à la fin place l'ombre à l'intérieur

Définitions de styles : ombres

■ text-shadow: ombre du texte (4 valeurs)

- la position horizontale (peut être négative)
- la position verticale (peut être négative)
- le flou (optionnel)
- la couleur

```
h1 {  
  text-shadow: 2px 2px 8px #FF0000;  
}
```

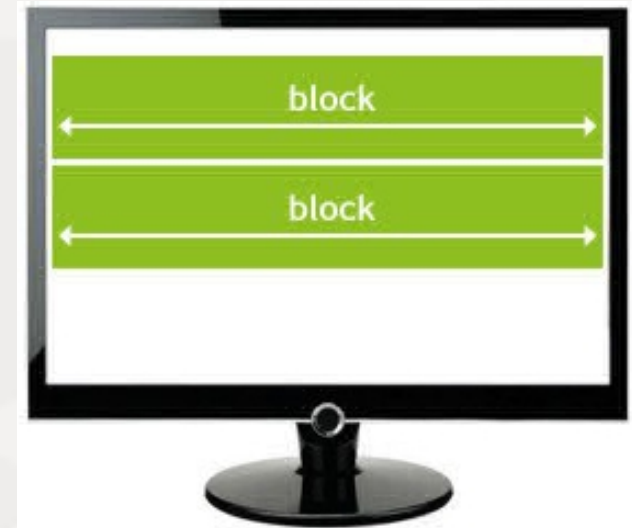
Display

- La propriété display détermine le comportement de l'élément sur la page web
 - inline | block | inline-block | flex | table-cell | none | ...

Display

■ display: block;

- Hauteur par défaut en fonction du contenu
- Largeur par défaut maximale (largeur de l'élément parent)
- Largeur (width), hauteur (height), marges modifiables
- Elements disposés verticalement



Display

■ display: inline;

- Hauteur en fonction du contenu
- Largeur en fonction du contenu
- width, height non modifiables
- Elements disposés côte à côte ("en ligne")

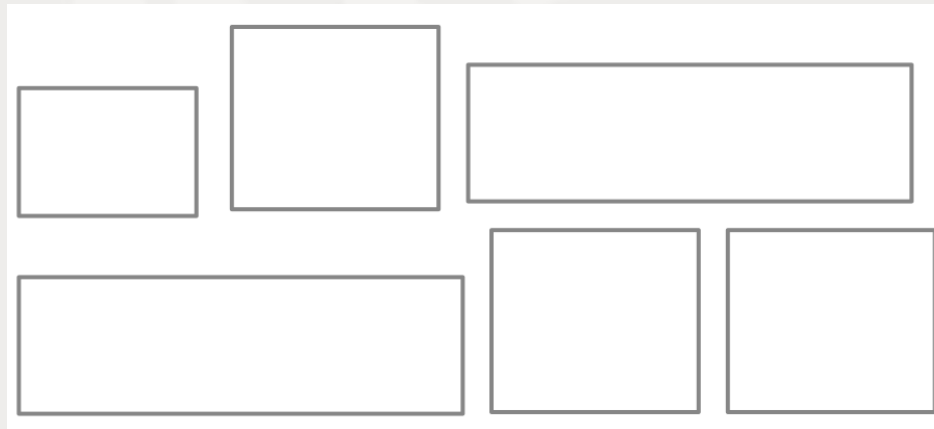
■ (Souvent des morceaux de paragraphe)



Display

■ display: inline-block;

- Hauteur en fonction du contenu
- Largeur en fonction du contenu
- width, height modifiables
- Elements disposés côte à côte ("en ligne")



Display

■ Les balises appartiennent à une des trois catégories :

Balises block	Balises inline	Balises inline-block
<p>		<input>
<h1>		<textarea>
<article>	<a>	<canvas>
<header>		...
		
<div>	...	
...		

Dimensions

- Pour un élément de type block ou inline-block
- width : largeur d'un élément
 - Taille fixe en px
 - Taille en % par rapport à l'élément parent
- min-width: largeur minimale d'un élément
- max-width: largeur maximale d'un élément
- Le comportement est identique pour la propriété height

Dimensions

- Lorsque les dimensions d'un bloc sont définies, celui-ci ne s'adapte plus forcément au texte qu'il contient.
- `overflow`: comportement en cas de texte trop long
 - `hidden` | `overlay` | `scroll` | `visible`
- `text-overflow`:
 - `ellipsis` | `clip`
- `word-wrap`:
 - `break-word` | `normal`

Marges

- Il est possible de définir des marges externes (margin) et internes (padding) pour les éléments



Marges

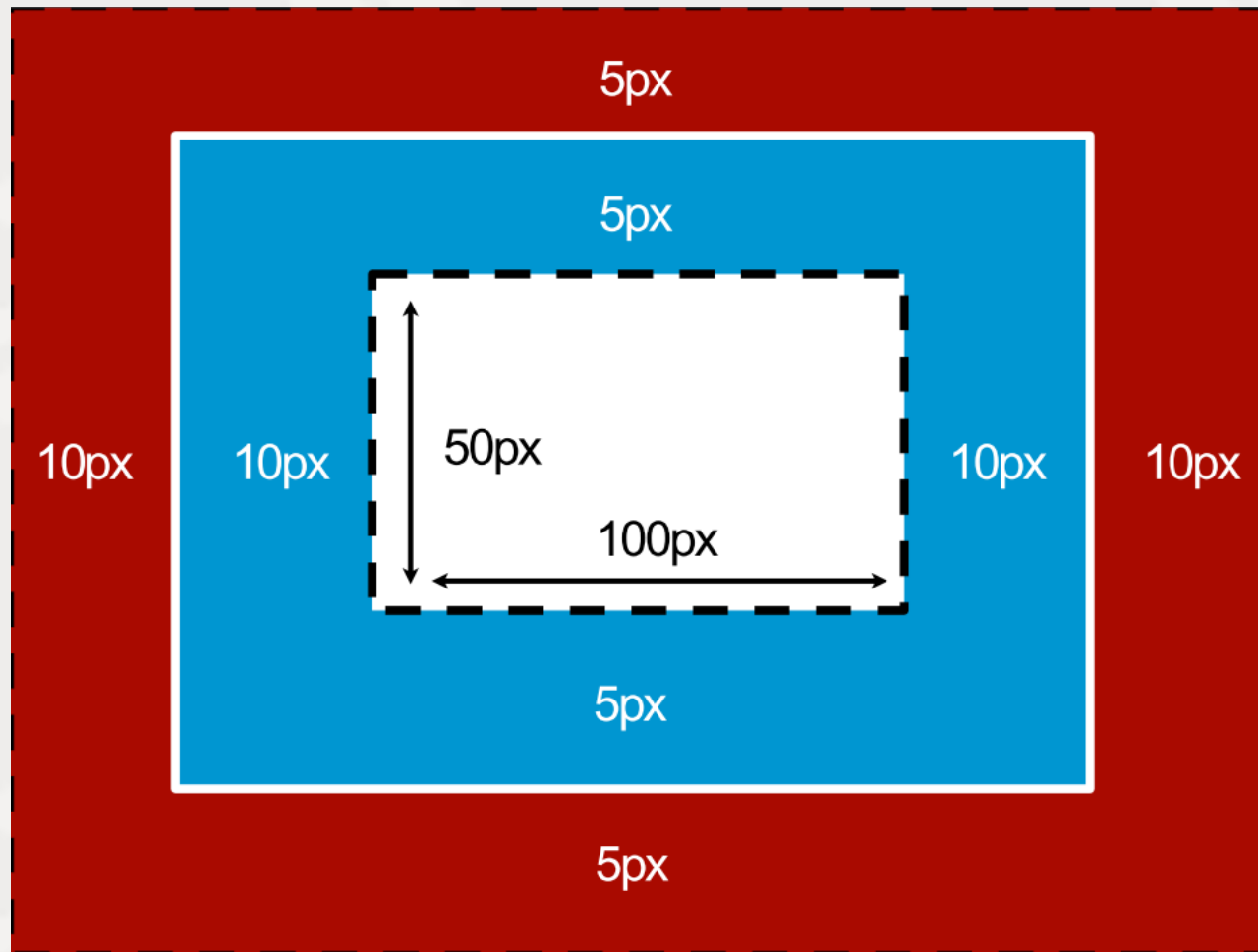
■ Exemple :

```
div {  
  height: 50px;  
  width: 100px;  
  padding: 5px 10px 5px 10px;  
  margin: 5px 10px 5px 10px;  
}
```

- margin est une combinaison des propriétés margin-top, margin-right, margin-bottom, margin-left
- de même pour padding
- Les 4 valeurs sont: haut, droite, bas, gauche

Marges

■ Exemple :



Marges

■ Taille totale :

- $\text{Height} = \text{height} + \text{padding-top} + \text{padding-bottom} + \text{margin-top} + \text{margin-bottom}$
- $\text{Height} = 50\text{px} + 5\text{px} + 5\text{px} + 5\text{px} + 5\text{px} = 70\text{ px}$
- $\text{Width} = \text{width} + \text{padding-left} + \text{padding-right} + \text{margin-left} + \text{margin-right}$
- $\text{Width} = 100\text{px} + 10\text{px} + 10\text{px} + 10\text{px} + 10\text{px} = 140\text{px}$

Marges

- Il est possible de définir les marges à droite et à gauche en automatique afin de centrer un élément block.

```
div{  
  height: 50px;  
  width: 50px;  
  margin: 0px auto;  
}
```

- Il existe d'autres techniques pour centrer un élément verticalement.

Positionnement

■ `position: static;`

- Par défaut. Les propriétés `left`, `top`, `right` et `bottom` ne s'appliquent pas

■ `position: relative;`

- Le positionnement de l'élément se fera par rapport à l'endroit où il a été déclaré. Il sert de bloc de référence pour le positionnement de blocs enfants.

■ `position: fixed;`

- L'élément reste fixe, il ne bouge pas lorsque l'écran défile.

Positionnement

■ position: absolute;

- Le positionnement de l'élément se fait par rapport au coin haut gauche de la page ou par rapport à l'élément parent s'il est relatif.

■ Pour les positions fixed, relative, absolute, il est possible de positionner l'élément grâce aux propriétés top, right, bottom et left.

Positionnement

■ Exemple :

```
p {  
    position: absolute;  
    left: 40px;  
    top: 75px;  
}
```

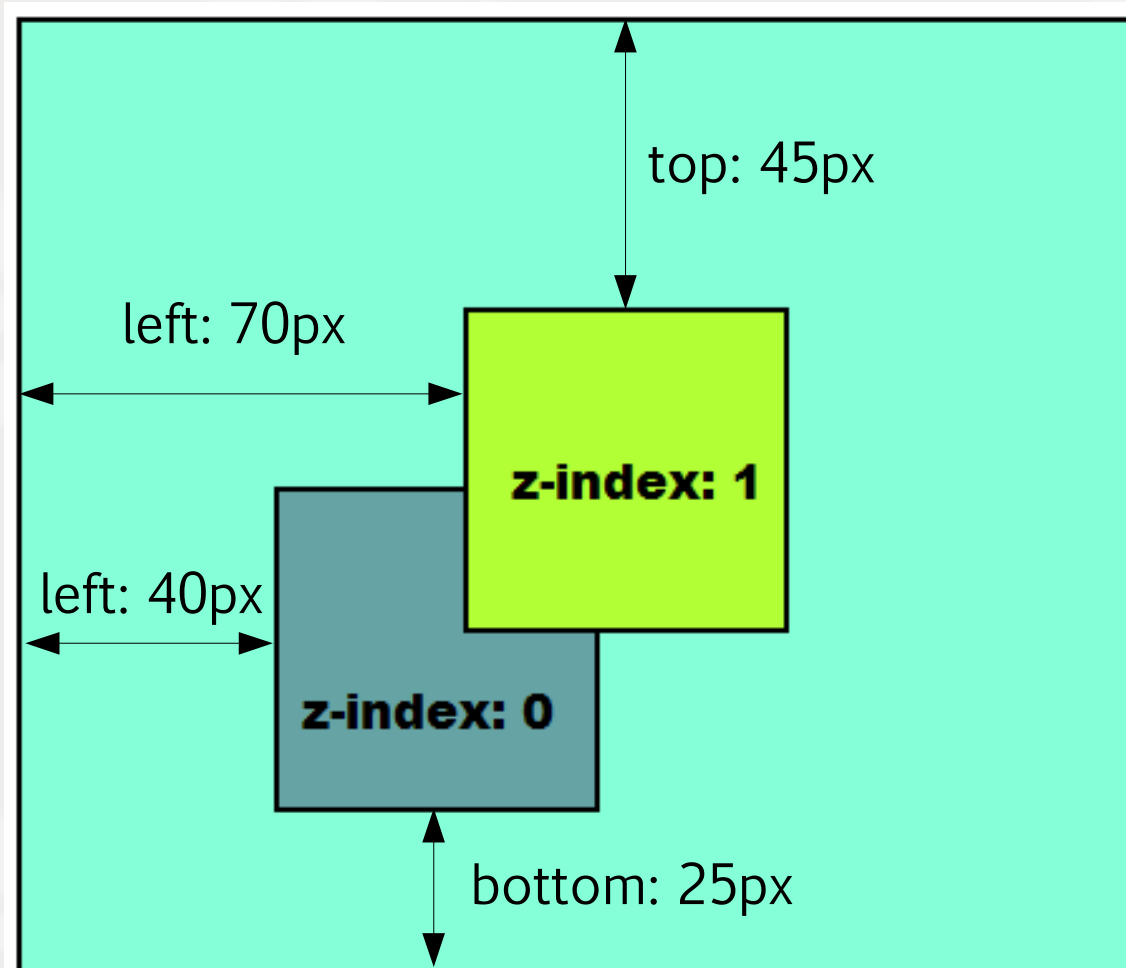
- L'élément est positionné à 40px du bord gauche de l'élément qui lui sert de référence (relatif) et à 75px du bord haut de l'élément qui lui sert de référence.

■ Ce type d'instruction permet de superposer des éléments

- La propriété z-index qui prend pour valeur un nombre, permet de gérer quel élément doit apparaître au dessus des autres.

Positionnement

■ Exemple :



Flottant

- Les éléments flottants sortent du flux normal d'une page html. Les éléments suivants viendront s'adapter au positionnement de l'élément flottant.
- Peut être intéressant notamment pour faire une lettrine.
- float : permet de faire flotter un élément

- left | right

Lorem ipsum dolor sit amet, consectetur
adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud

Les sélecteurs

■ Attribut id et class

- les balises HTML possèdent toutes les attributs id et class.
- un id permet d'identifier un élément HTML de façon unique.
- une classe permet d'identifier des éléments ayant une apparence ou un comportement identique.

Les sélecteurs

■ Sélecteurs de base:

- `p` : Sélectionne l'ensemble des balises `p`
- `#mon_id` : Sélectionne l'élément unique portant l'identifiant *mon_id*
- `.ma_classe` : Sélectionne les éléments portant la classe *ma_classe*
- `p.ma_classe` : Sélectionne les balises `p` ayant la classe *ma_classe*

Les sélecteurs

■ Parmi les autres sélecteurs basiques on retrouve:

- * : Sélecteur universel. On peut l'utiliser pour mettre les marges du documents à zéro
- li, a : Permet d'appliquer un style à deux balises en même temps
- li a : Applique le principe du sélecteur descendant afin de spécifier la sélection aux balises de lien de la liste et pas à tous les liens du document
- #mon_id div li : Sélectionne les éléments listes contenant dans la div contenue dans l'élément portant l'id *mon_id*

Les sélecteurs

■ Parmi les sélecteurs les moins courant:

- `ul + p` : Permet d'appliquer un style au paragraphe directement consécutif à une liste non ordonnée
- `#contenu > ul` : Permet de cibler uniquement les descendants liste non ordonnée de rang 1 par rapport à la balise d'identifiant contenu

Les sélecteurs

- `ul ~ p` : Permet de cibler tous les paragraphes qui suivent le `ul`
- `element[attribut(=value *optional)]` : Permet de cibler n'importe quel élément possédant l'attribut défini et éventuellement possédant telle ou telle valeur.
- `a[href*="domain"]` : Permet de cibler les éléments de type lien qui contiennent le terme *domain*
- `a[href^="domain"]` : Permet de cibler tous les éléments de type lien qui commencent par *domain*

Les sélecteurs

■ `a[href$=".jpg"]` : Permet de cibler tous les éléments de type lien qui se terminent par .jpg

■ Quelles sont les solutions pour sélectionner tous les liens affichant une image de type .jpg, ou .jpeg ou .png ou .gif et leur appliquer un style commun?

Les sélecteurs

■ Solution :

```
a[href$=".jpg"],a[href$=".jpeg"],a[href$=".png"],a[href$=".gif"] {  
    color: red;  
}
```

Les sélecteurs

■ Sélecteurs dynamiques:

- a:link : Permet de définir un style pour tous les liens non visités
- a:visited : Pour les liens déjà visités
- a:active : Lorsque l'on clique sur un lien
- a:focus : Lorsque le lien est sélectionné
- #monElement:hover : Lorsque l'on passe la souris au dessus de l'élément ayant l'identifiant *monElement*

Les pseudo classes

■ IE9 et autres navigateurs:

- `#element::after` : Ajoute du contenu qui peut être stylé après *element*
- `#element::before` : Ajoute du contenu qui peut être stylé avant *element*
- `#element::first-letter` : Ajoute du style à la première lettre du texte *d'element*
- `#element::first-line` : Ajoute du style à la première ligne
- `#element:firstchild` : Ajoute du style au premier enfant *d'element*

Transitions

- Il est possible d'effectuer un effet de transition sur un élément lorsqu'il passe d'un état à un autre (sur un :hover par exemple)
- transition-property:
 - nom des propriétés CSS qui participent à la transition
- transition-duration: durée de la transition
 - valeur en secondes (ex: 3s)

Transitions

- transition-timing-function: courbe de vitesse de l'effet
 - ease | linear | ease-in | ease-out | ease-in-out | cubic-bezier()
- transition-delay: délais avant début de transition
 - valeur en secondes (ex: 3s)

Transitions

■ transition: combine les propriétés précédentes

■ transition : property duration timing-function delay;

```
div {  
  width: 100px;  
  height: 100px;  
  background: red;  
  transition: width 2s;  
  -webkit-transition: width 2s ease-in, background 1s 0.5s;  
}  
div:hover {  
  width: 300px;  
  background: blue;  
}
```


RWD

■ Responsive Web Design

- Solution simple et flexible
- Une seule arborescence DOM
- Un contenu unique

CSS3 Media Queries

■ Possibilité d'adapter l'interface au périphérique utilisé :

- Résolution,
- Profondeur d'écran (nombre de couleurs),
- Type d'écran : TV, encre électronique, tactile, etc
- Braille, synthèse vocale, impression
- Etc...

■ Il existait déjà un attribut « media » dans CSS2

CSS3 Media Queries

■ Depuis CSS2 :

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

- En fonction de la valeur de l'attribut media, on chargeait une feuille de style adaptée
- Valeurs possibles pour media en CSS2 : screen, handheld, print, aural, braille, embossed (imprimante braille), projection (slides), tty (terminal ascii), tv, all

CSS3 Media Queries

■ Autre syntaxe dans le fichier CSS :

```
@media print {  
  body { font-size: 1.5em; color:black; }  
  ...  
}
```

- C'est le navigateur qui interprète cet attribut

CSS3 Media Queries

- Les Media Queries de CSS permettent de spécifier une taille d'affichage :

```
<link rel="stylesheet" media="screen and (max-device-width: 640px)"  
href="smallscreen.css" type="text/css" />
```

- Ou dans le fichier CSS

```
@media (min-width: 480px) {  
  aside{  
    margin-bottom : 5px;  
  }  
  aside div{  
    display : block;  
    margin-right : 200px;  
  }  
}
```

CSS3 Media Queries : Syntaxe

- La Media Queries teste ses arguments et le style ne s'applique que si le résultat de la requête est vrai.

- **and** : Permet de lier deux requêtes

```
@media ( min-width : 600px) and (max-width: 800px) {  
  html {  
    background: red;  
  }  
}
```

- **or** : Le ou s'effectue grâce à une virgule

```
@media ( max-width : 600px) , ( min-width : 800px) {  
  html {  
    background: red;  
  }  
}
```

CSS3 Media Queries : Syntaxe

■ Les règles suivantes sont disponibles :

Règle	Définition	Préfixe min/max
color	Gestion de la couleur (bits)	oui
color-index	Nombre de couleurs (ex: 256)	oui
height	Hauteur de la fenêtre	oui
width	Largeur de la fenêtre	oui
device-height	Hauteur du périphérique	oui
device-width	Largeur du périphérique	oui
orientation	Orientation du périphérique (portrait ou landscape)	non
aspect-ratio	Ratio d'affichage (ex: 16/9)	non
resolution	Densité de pixels (en dpi)	oui

CSS3 Media Queries : Exemple

■ Exemples :

```
@media screen and (device-aspect-ratio: 16/9) { ... }  
@media (orientation:portrait) { ... }  
@media (color) { /* Ecran couleur */ }  
@media (min-color-index: 256) {  
    /* écran support au moins 256  
    couleurs */ }  
@media (monochrome) {  
    /* L'écran est monochrome */ }  
@media(min-monochrome: 2) {  
    /* L'écran a au moins 4 niveaux de  
    gris */ }  
@media (tv, screen and (device-aspect-ratio: 16/9), screen and (device-  
aspect-ratio: 16/10) {  
    /* Ecran ou TV à écran large */ }
```

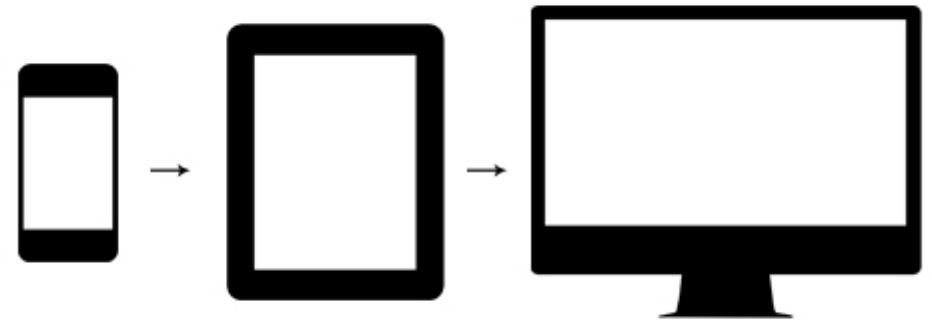
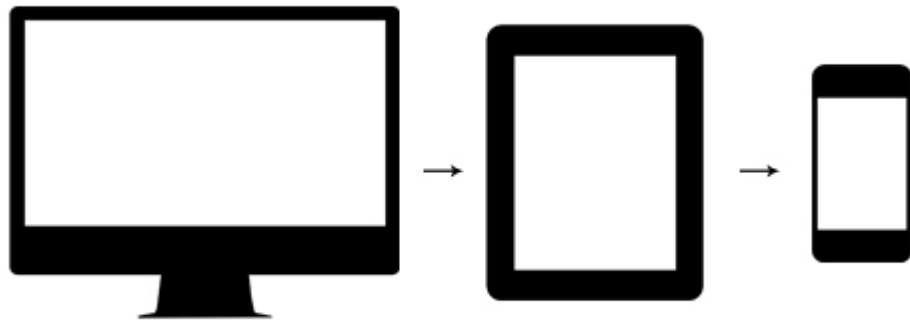

Mobile First



Dégradation

vs

Amélioration



Mobile First

■ Pourquoi commencer par le mobile?

- Priorisation des besoins et de l'activité
- Si le site est performant sur mobile, il le sera sur desktop
- Plus facile d'ajouter des éléments que d'en retirer
- Elements complexes à réduire en format mobile
- Applications plus robustes
- Bientôt plus de visiteurs mobiles que bureau. 7 milliards d'abonnements mobiles dans le monde, 2.3 milliards de smartphones.

Mobile First

■ Exemple :

```
selector { property: value; } /* common properties (desktop / mobile)

@media (min-width: 480px) { /* smartphones & small tablets */
  selector { property: value; }
}

@media (min-width: 768px) { /* wider tablets & notebooks */
  selector { property: value; }
}

@media (min-width: 1024px) { /* laptops */
  selector { property: value; }
}

@media (min-width: 1920px) { /* huge screens */
  selector { property: value; }
}
```

Viewport

- Les mobiles sont beaucoup plus petits que nos écrans d'ordinateurs. Pourtant ils possèdent parfois des résolutions identiques voire supérieures.
- Pour contourner le problème, les navigateurs mobiles ne se basent pas sur la résolution mais sur leur viewport.

Navigateur	Largeur du viewport
Opera Mobile	850px
iPhone Safari	980px
Android Chrome	980px
Windows Phone IE	1024px

Viewport

■ Le viewport peut être redéfini dans l'attribut content d'une balise <meta>. On peut également modifier le comportement par défaut des appareils mobiles.

■ width : permet de redéfinir la taille du viewport

```
<meta name="viewport" content="width=device-width">
```

■ initial-scale : niveau de zoom initial

```
<meta name="viewport" content="initial-scale=1.0">
```

■ user-scalable : possibilité de zoomer

```
<meta name="viewport" content="user-scalable=true">
```

Viewport

- minimum-scale : niveau de zoom minimal

```
<meta name="viewport" content="minimum-scale=0.5">
```

- maximum-scale : niveau de zoom maximal

```
<meta name="viewport" content="maximum-scale=1">
```

- Pour un site responsive, on paramètre en général le viewport de la façon suivante :

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Viewport

■ Valider votre code (HTML5)

- <http://validator.w3.org>

■ Vérifier la compatibilité des fonctionnalités

- <http://caniuse.com>