

# **Programmation avancée pour le Web**

## **Les feuilles de styles CSS3 (Cascading Style Sheets)**

**Réaliser par :** Mr: Chouha Adel

**E-mail :** adch05@yahoo.fr

## **Introduction**

- Les feuilles de styles permettent de décrire la présentation des éléments HTML, de façon séparée du contenu et de sa structure, décrits par le HTML.
- Cette séparation permet facilement de modifier le style d'un site en éditant un seul fichier au lieu de parcourir les pages HTML et changer le style de chacune des balises.

## Cascade de styles

Il existe quatre niveaux de définition des styles appelé cascade :

1. Le style par défaut défini par chaque navigateur pour toutes les balises HTML
2. Le style défini dans une feuille de style séparée et inclut dans une page HTML grâce à la balise `<link rel='stylesheet' href='URL d un fichier de styles .css' />`
3. Le style peut être défini au niveau d'une page du site en introduisant la balise `<style> </style>` au niveau de la balise `<head>`. ( **n'est pas supportés**)
4. L'attribut global **style** permet de définir pour toutes balise HTML un style. . ( **n'est pas supportés**)

3

## Cascade de styles

- Ces quatre niveaux de définitions de style définissent ce qui nommé la cascade CSS.
- Un style défini au niveau feuille séparée remplace un style défini au niveau par défaut ;
- un style défini au niveau de la balise « style » remplace le style défini au niveau feuille séparée,
- et un style défini au niveau attribut remplace le style défini aux autres niveaux.

4

# Cascade de styles

**Imbrication des définitions** : la plus locale l'emporte.

- Ex. Définition de style dans un fichier externe vs. Définition dans un élément → la définition locale de l'élément prévaut

**Imbrication des éléments** :

le style d'un élément s'applique aux éléments imbriquées.

- **Ex.** `<p>...<em>...</em>...</p>` Si un style est appliqué à `<p>`, il s'applique à `<em>`.

5

# Cascade de styles

**Unités de taille:**

- ***Unités absolues***

- **px** : pixel
- **pt** : point

- ***Unités relatives***

- **em** : hauteur de ligne
- **%** : proportion par rapport à la taille de la fenêtre

6

# Écriture d'une feuille de style

**Feuille de style** : ensemble de règles

- Un sélecteur
  - Des propriétés
  - Une valeur pour chaque propriété
- **La feuille de style css** est constituée d'un ensemble de règles **CSS**. Chaque règle CSS est constituée de deux partie : un sélecteur et des déclarations (les déclarations sont entre deux accolades).
  - Enfin chaque déclaration est constituée d'une propriété CSS et d'une valeur séparée par deux point ' : ' et terminée par un point-virgule ' ; '.

**Exemple**

```
h1 {  
  color: yellow;  
  font-weight: bold;  
}
```

7

## Sélecteurs

- **Sélecteur universel \*** : tous les éléments HTML
  - Ex: \* { color: blue; }
- **Style d'une catégorie d'éléments**
  - Ex: h1 { color: #FF00FF; }
- **Style de plusieurs catégories d'éléments**
  - Ex: h1, h2, h3, p { color: green; }
- **Style pour éléments imbriqués**
  - Ex: p em { color: red; } /\* Italiques dans un paragraphe \*/

8

# Sélecteurs

Cas particuliers des attributs class et id :

- **E.c** tout élément **E** appartenant à la **classe c**
  - *ex* : div.exercice, \*.solution,
- **E#ident** tout élément **E** dont l'**id** vaut **ident**
  - *ex* : img#joconde, \*#joconde, #unique

9

# Classe de style

- **Sélecteur défini par l'utilisateur:**

qui permet de dissocier un style d'éléments particuliers

**Ex:**

- .style\_rouge { color: red; }
  - .style\_bleu { color: blue; }
  - p.parag\_vert { color: green; }
- **Sélecteur associé à un identifiant particulier**
    - **Ex:** #titre { text-align:center; color: navy; }

10

# Classe de style

- **Utilisation dans une page Web**

**Ex:**

- `<h1 class="style_rouge">Titre rouge</h1>`
- `<p class="style_rouge">Texte rouge</p>`
- `<p class="style_bleu">Texte bleu</p>`
- `<p class="parag_vert">Texte vert</p>`

- **Application à un ensemble d'éléments : `<div> </div>`**

**Ex:**

- `<div class="style_bleu">`
- `<h1>...</h1>`
- `<p>...</p>`
- `<p>...</p>`
- `</div>`

11

# Classe de style

- **À une partie d'un paragraphe : `<span> </span>`**

**Ex:**

```
<p class="style_rouge">Ceci est rouge sauf <span class="style_bleu">ce qui est
bleu</span>
</p>
```

- **À un élément identifié :**

**Ex:**

```
<div id="titre"> ... </div>
<p id="titre"> ... </p>
```

12

# Combinaison de sélecteurs

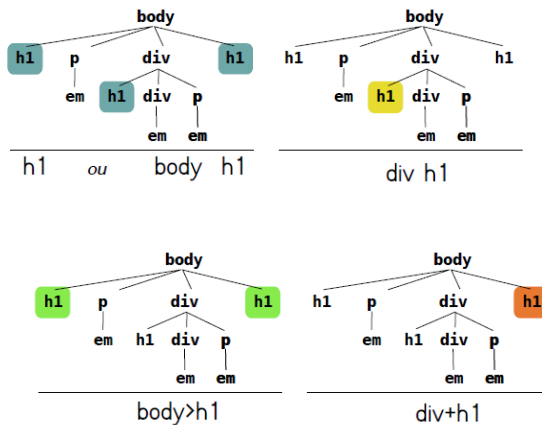
S'appuie sur la structure **arborescente** du document

Si **Sel1** et **Sel2** sont des sélecteurs :

- **Sel1 Sel2** tout élément sélectionné par **Sel2 descendant** d'un élément sélectionné par **Sel1**
- **Sel1 > Sel2** tout élément sélectionné par **Sel2** qui est **fils** d'un élément sélectionné par **Sel1**
- **Sel1 + Sel2** tout élément sélectionné par **Sel2** qui **suit immédiatement** un élément sélectionné par **Sel1** (premier **frère** suivant)
- **Sel1 ~ Sel2** tout élément sélectionné par **Sel2** qui **suit** un élément sélectionné par **Sel1** (un **frère** suivant)

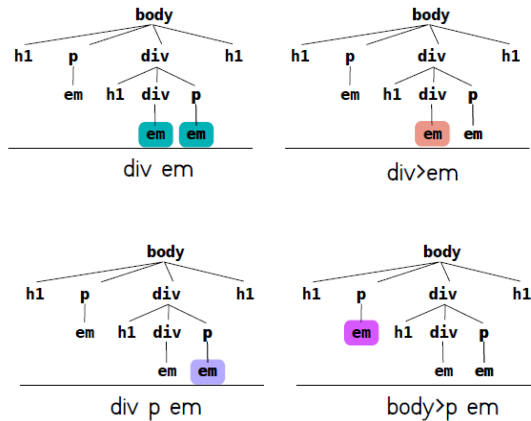
13

## Exemple



14

# Exemple



15

# Sélecteurs et Attributs

- **A[attribut] : une balise qui possède un attribut**

```
a[title]
{
}
```

Sélectionne tous les liens <a> qui possèdent un attribut title.

- **A[attribut="Valeur"] : une balise, un attribut et une valeur exacte**

```
a[title="Cliquez ici"]
{
}
```

Idem, mais l'attribut doit en plus avoir exactement pour valeur « Cliquez ici ».

- **A[attribut\*="Valeur"] : une balise, un attribut et une valeur**

```
a[title*="ici"]
{
}
```

Idem, l'attribut doit cette fois contenir dans sa valeur le mot « ici » (peu importe sa position).

16



## Sélecteurs et Attributs

- **E[att^="prefixe"]** tout élément **E** dont la valeur de l'attribut **att** commence exactement par *prefixe*

*ex* : a[href^="http://fil.univ-lille1.fr"]

- **E[att\$="suffixe"]** tout élément **E** dont la valeur de l'attribut **att** se termine exactement par *suffixe*

*ex* : img[src\$=".png"], a[href\$=".pdf"]

- **E[att^="prefixe"]** tout élément **E** dont la valeur de l'attribut **att** commence exactement par *prefixe*

*ex* : a[href^="http://fil.univ-lille1.fr"]

17

## sélecteurs de pseudo-classes et pseudo-éléments

- **E:pseudoC** tout élément **E** appartenant à la **pseudo-classe** *pseudoC*

*ex* : a:visited

- **E:pseudoE** tout **pseudo-élément** *pseudoE* de l'élément **E**

*ex* : h1:first-letter, p[lang=fr]:first-line

18

## Pseudo-classes

- **:empty** un élément **E** sans descendant (y compris nœud texte)
- **:first-child** un élément qui est premier fils d'un autre élément  
*ex* : `div.exercice:first-child`
- **:last-child** élément dernier fils d'un autre élément
- **:hover** est « sous » le pointeur de la souris
- **:visited** (<a> uniquement) lien déjà visité
- **:link** (<a> uniquement) lien non encore visité'

19

## Pseudo- éléments

- **:first-line** la première ligne « formatée » d'un élément
- **:first-letter** la première lettre « formatée » d'un élément
- **:before** insertion de contenu avant l'élément
- **:after** insertion de contenu après l'élément

20

## Conflit entre règles

### Calcul de priorité

- On compte pour chaque sélecteur :
  - a. nombre de sélecteurs d'id (= nombre de #)
  - b. nombre de classes, pseudo-classes ou d'attributs
  - c. nombre d'éléments ou de pseudo-éléments
- Le sélecteur reçoit la priorité **a b c**.
- Le sélecteur avec la plus grande priorité est choisi.
- En cas d'égalité, la dernière déclaration est choisi.

21

## Conflit entre règles

sélecteur	a	b	c
* {...}	0	0	0
h1 {...}	0	0	1
div.reponse {...}	0	1	1
#joconde {...}	1	0	0
div a {...}	0	0	2
div a:visited {...}	0	1	2
p span.fichier {...}	0	1	2
p a[href\$=".pdf"] {...}	0	1	2
p.enonce a[href\$=".pdf"] {...}	0	2	2
ol.exercice li.question {...}	0	2	2
div#diaporama img.gauche {...}	1	1	2
article p::first-letter {...}	0	0	3
article#special p::first-letter {...}	1	0	3

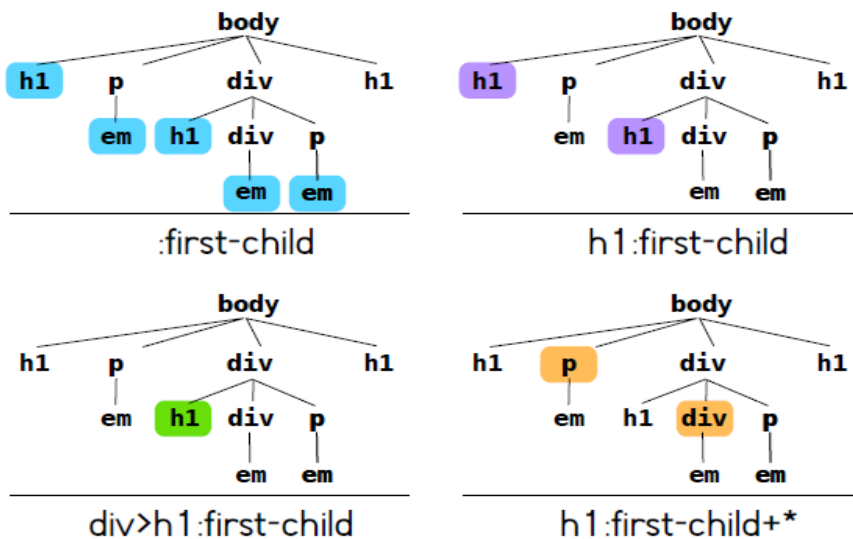
22

# Héritage

- lorsque pour un élément aucune règle ne définit de valeur pour une propriété, c'est la valeur de cette propriété pour son parent qui s'applique
- toutes les propriétés ne s'héritent pas  
*ex* : margin, padding, etc

23

## Exemple



24

# Propriétés de couleur

## Spécification de couleurs:

- **Nom prédéfini** : aqua, black, blue, fuchsia, green, gray, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow
- **Code RGB hexadécimal**
  - Ex: Jaune : #FFFF00
- **Spécification rgba(rouge,vert,bleu,transparence)** avec :
  - rouge, vert et bleu : nombres entre 0 et 255
  - transparence : nombre entre 0 et 1

- **color** `color: black;`  
`color: #000000;`

25

# Arrière-plan

- **background-color** `background-color: white;`  
`background-color: #FFFFFF;`
- **background-image** `background-image: url("URL");`
- **background-repeat** `background-repeat: no-repeat;`
- **background-position** `background-position: center;`

26

## Propriétés de typographie

- **font-family**  
font-family: times;  
font-family: arial, verdana;
- **font-size**  
font-size: 1.5em;
- **font-style**  
font-style: italic;
- **font-weight**  
font-weight: bold;  
font-weight: bolder;
- **text-decoration**  
text-decoration: underline;  
text-decoration: line-through;

27

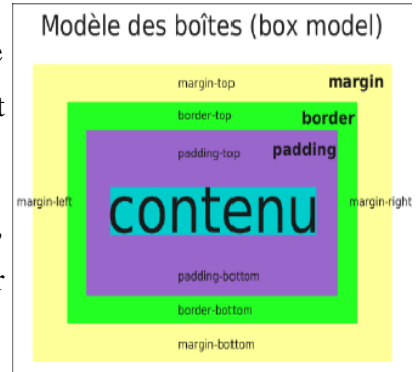
## Propriétés de texte

- **text-align**  
text-align: left;  
text-align: right;  
text-align: center;  
text-align: justify;
- **text-indent**  
text-indent: 10pt;  
text-indent: 15px;  
text-indent: 5%;

28

## Le modèle en boîte

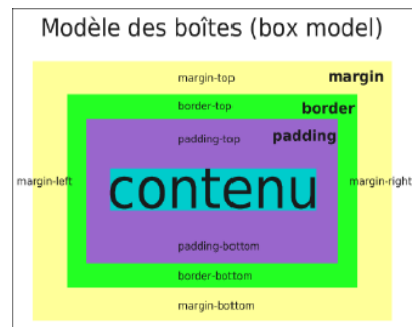
- les blocs suivent ce qu'on appelle le modèle en boîte.
- ils possèdent une **bordure**, un espacement intérieur entre leur contenu et la **bordure** (**padding**) et un espacement extérieur entre la **bordure** et les autres blocs (**margin**).
- Il existe une marge supérieure (**top**), droite (**right**), inférieure (**bottom**) et gauche (**left**). Et de même pour l'espacement intérieur (padding).



29

## Le modèle en boîte

- largeur : width
- hauteur : height
- marge intérieure : padding
- marge extérieure : margin
- bordure : border: type size color;
  - type : solid, double, inset, outset
  - size : épaisseur de la bordure
  - color : n'importe quelle couleur
- arrondis : border-radius : taille;
- padding, border et margin peuvent être différents pour chaque côté (-left, -right, -top et -bottom)
- width et height donnent uniquement la taille du contenu.



30

## Le modèle en boîte

- Tous ces paramètres du bloc peuvent être spécifié comme ceci :

```
#unbloc {  
  margin : 12px 3px 6px 9px;  
  padding : 12px 3px 6px 9px;  
}
```
- Ils peuvent aussi être spécifiés séparément comme ceci :

```
margin-top : 12px ;  
margin-right: 3px ;  
margin-bottom:6px ;  
margin-left : 3px ;  
padding-top : 12px ;  
padding-right: 3px ;  
padding-bottom:6px ;  
padding-left : 3px ;
```

31

## Propriétés d'hyperlien

- **Hyperliens**
  - Ex: a { color: blue; }
- **Hyperliens visités**
  - Ex: a:visited { color: purple; }
- **Hyperliens survolés (avec le pointeur de souris)**
  - Ex: a:hover { color: white; background-color: blue; }
- **Hyperliens activés**
  - Ex: a:active { color: purple; }

32



## Propriétés de liste

### Listes non-ordonnées

- list-style                   list-style: disc;  
                                  list-style: circle;  
                                  list-style: square;
- list-style-image           list-style-image: url("URL");

### Listes ordonnées

- list-style                   list-style: decimal;  
                                  list-style: upper-roman;  
                                  list-style: lower-alpha;

33

## Propriétés de tableau

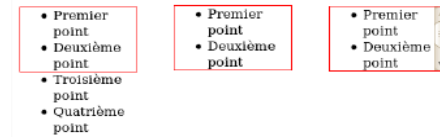
- **Élément table :**           border, width,  
                                  border-collapse: collapse...
- **Élément caption :**       caption-side: top;  
                                  caption-side: bottom;
- **Élément tr :**               height
- **Éléments td et th :**       border, width...  
                                  vertical-align: top;  
                                  vertical-align: middle;  
                                  vertical-align: bottom;

34

# Dépassement et type de balises

- **overflow** : comportement en cas de dépassement

- **visible** : ce qui dépasse est affiché
- **hidden** : caché
- **auto** : barre de défilement (scroll)



- **display** : modification du comportement d'affichage

- **block** : s'affiche comme un élément de type block
- **inline** : s'affiche comme un élément de type inline
- **inline-block** : taille, marge mais pas de retour à la ligne
- **none** : supprime l'élément

35

# Positionnement flottant

- **HTML** :

```
<div class="menu_jaune">  
  <p>Menu</p>  
</div>  
<div>  
<!-- Contenu de la page -->  
</div>
```

Exemple 1 : menu flottant à droite



- **CSS:**

```
.menu_jaune { float: right;  
  background-color: yellow;  
  width: 100px; }
```

36

# Positionnement flottant

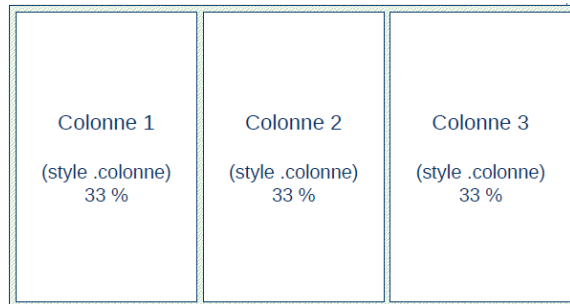
## Exemple 2 : pagination en trois colonnes

### HTML

```
<div class="page"> <!-- Conteneur -->
<div class="colonne"> </div>
<div class="colonne"> </div>
<div class="colonne"> </div>
</div>
```

### CSS

```
.page { float: left; width: 100%; }
.colonne { float: left; width: 33%; }
```



37

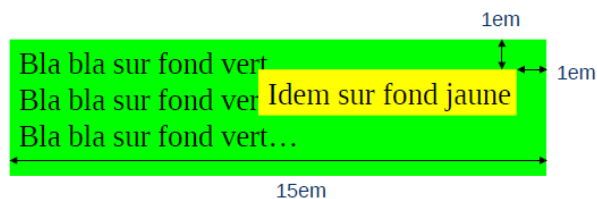
# Positionnement absolu

### HTML:

```
<div class="boite_verte">
<p>Bla bla sur fond vert...</p>
<div class="boite_jaune">
<p>Idem sur fond jaune</p>
</div>
</div>
```

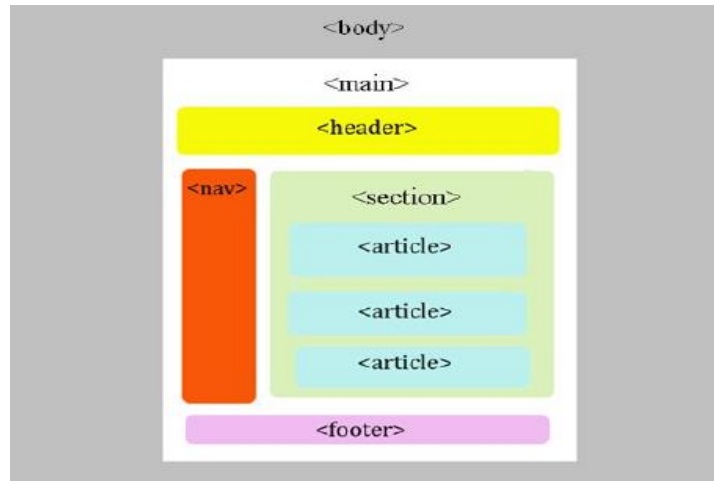
### CSS:

```
.boite_verte {
position: relative;
background-color: #00FF00;
width: 15em;
}.
boite_jaune {
position: absolute;
top: 1em;
right: 1em;
background-color: #FFFF00;
}
```



38

# Exemple de mise en page



39

# Exemple de mise en page

## HTML:

```
<main>
  <header>
    <h1>Exemple de mise en page</h1>
  </header>
  <div id="conteneur">
    <nav>
      <a href="http://www.univ-Batna2.dz">Université Batna2</a>
    </nav>
    <section>
      <article>Article 1</article>
      <article>Article 2</article>
      <article>Article 3</article>
    </section>
  </div>
  <footer>
</footer>
</main>
```

40

## Exemple de mise en page

### CSS:

```
* {
  font-family: arial;
}
body {
  background-color: silver;
}
main {
  width: 1000px;
  margin-left: auto;
  margin-right: auto;
  background-color: white;
  padding: 1em;
}
header {
  background-color: yellow;
  text-align: center;
  padding: 1em;
  border-radius: 15px;
}
#conteneur {
  width: 100%;
  margin-top: 1em;
}
nav {
  float: left;
  width: 200px;
  background-color: red;
  padding: 1em;
  border-radius: 15px;
}
section {
  float: left;
  width: 750px;
  background-color: lime;
  margin-left: 1em;
  border-radius: 15px;
}
article {
  background-color: aqua;
  margin: 1em;
}
footer {
  background-color: fuchsia;
  text-align: center;
  padding: 1em;
  border-radius: 15px;
}
```

41

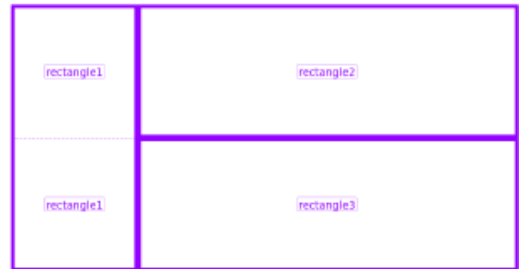
## Grilles

- Mise en page à l'aide de grilles (en CSS)
- Le conteneur se comporte comme une grille/tableau
- **Ses enfants** sont rangés dans les cases
- Pas de modification du code HTML
- Indiquer que c'est une grille avec **display:grid**
- Indiquer le nombre de lignes (et les tailles) avec **grid-template-rows**
- Indiquer le nombre de colonnes (et les tailles) avec **grid-template-columns**
- Former des zones (rectangulaires) nommées avec **grid-template-areas**

42

# Grilles

```
.conteneur{  
  display: grid; /* C'est une grille */  
  /* Avec deux lignes de 100 pixels */  
  grid-template-rows: 100px 100px;  
  /* Avec deux colonnes (200 et 50 pixels) */  
  grid-template-columns: 200px 50px;  
  grid-template-areas:  
  "rectangle1 rectangle2"  
  "rectangle1 rectangle3";  
}
```



43

# Grilles

## Remarques:

- Il existe pour les tailles de colonnes et lignes l'unité fr (pour free space)  
**grid-template-columns:** 1fr 3fr 200px;
  - une colonne
  - une colonne 3 fois plus large
  - une colonne de 200px⇒ Les 4 premières colonnes prennent toute la largeur disponible.
- On peut utiliser **repeat(n,taille)** pour donner les tailles
- On peut ajouter un espace entre les lignes/colonnes avec **row-gap** et **column-gap** (anciennement **grid-row-gap** et **column-row-gap**)
- On n'est pas obligé de préciser **grid-template-rows** (les lignes prennent alors la place minimale nécessaire)

44

# Grilles

## Enfants sur la grille:

- Pour chaque enfant, on précise la zone dans laquelle il doit s'afficher avec **grid-area: nom\_de\_la\_zone**
  - Chaque enfant doit être affecté à une zone
  - Toutes les zones doivent être utilisées

```
<section class="conteneur">
  <p>Paragraphe 1</p>
  <p>Paragraphe 2</p>
  <p>Paragraphe 3</p>
</section>
```

```
section > p:nth-child(1){
  grid-area: rectangle1;
}
section > p:nth-child(2){
  grid-area: rectangle2;
}
section > p:nth-child(3){
  grid-area: rectangle3;
}
```

45

# Responsive design

## Principe:

- Affichage "confortable" du site quel que soit le support (taille)
- Pas de modification du fichier HTML sauf ajout dans le head :

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```
- Du code CSS s'applique pour certaines tailles d'écran seulement
- Utilisation des *media queries*
  - **@media** (condition){  
/\* Règles CSS qui sont prises en compte si la condition est vérifiée \*/  
}
  - condition est définie par min-width:taille et/ou max-width:taille

46