

Technologies du Web

CSS, les feuilles de style

Notion 2

Florent Carlier, Valérie Renault

prenom.nom@univ-lemans.fr
Université du Maine

- Pourquoi dit-on qu'une balise est "deprecated" ou "obsolètes" ?
- Qu'est-ce qu'une méta-données ?
- A quoi sert le validateur W3C pour HTML ?
- Quels sont les balises permettant d'insérer une image ou un lien ?
- Quelle est la différence entre la balise `<head>` et la balise `<title>` ?
- Quelle est la différence entre la balise `` et la balise `<p>` ?
- Qu'est ce qu'un moteur de rendu ?

Technologies du Web

CSS, les feuilles de style

Notion 2

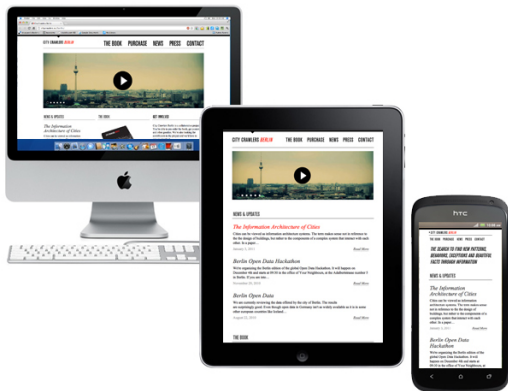
Florent Carlier, Valérie Renault

prenom.nom@univ-lemans.fr
Le Mans Université

CSS : Pourquoi faire ?



CSS : Pourquoi faire ?



Définition : Responsive Web Design (RWD)

- Contenus Web auto-adaptables en fonction des interfaces de consultations de l'utilisateur ;
- Minimum de redimensionnement, de recadrage et de défilement selon les écrans ;
- Réorganisation /suppression du contenu de la page ;
- Pas de duplication des pages en fonction du support cible => un référencement optimal ;
- Une seule adresse Web (URL) => pas de sous-domaine "mobile" ;

Définition : Design "Static"

- Dimension figée (exple 960px) quelque soit la taille de l'écran ;
- Version du site web dédiée pour le mobile ;

Définition : Design "Fluide" (Liquide)

- Dimensions exprimées en unités variables (pourcentage) ;
- Dimensions s'adaptent à la taille de le fenêtre ;

Définition : Design "Adaptatif"

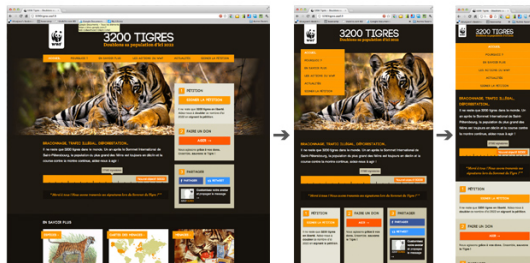
- Unités de largeur sont fixes mais différentes selon la taille des écrans ;
- Taille détectée via CSS3 Media Queries ;
- Autant de gabarits que de points de ruptures identifiés (320px, 480px, 1024px, etc.)

Définition : Design "Responsive"

- Méthodes CSS3 Media Queries modifiant les styles (réorganisation de la page) quel que soit le point de rupture ;
- Le HTML ne change pas ;
- Le contenu est conservé ;

Démonstration :
<http://www.liquidapsive.com/>

CSS : Pourquoi faire ?



Intérêt des CSS

- gestion séparée du contenu et de la présentation ;
- meilleure portabilité du contenu ;
- portabilité du style : harmonisation de la charte graphique ;
- gestion du code simplifiée avec plus de lisibilité en cas de modification et de mise à jour.

Exemple : 1 HTML / 2 CSS

Bonjour

Ceci est un exemple pour les CSS

- menu 1
- menu 2
- menu 3

Titre	Année	Nom Réalisateur	Prénom	Année naissance
Alien	1979	Scott	Ridley	1943
Vertigo	1958	Hitchcock	Alfred	1899

une cellule grise une cellule bleue

Bonjour

Ceci est un exemple pour les CSS

- menu 1
- menu 2
- menu 3

Titre	Année	Nom Réalisateur	Prénom	Année naissance
Alien	1979	Scott	Ridley	1943
Vertigo	1958	Hitchcock	Alfred	1899

une cellule grise une cellule bleue

Solution 1 : Déclaration d'une CSS par fichier externe

Dans la balise <head> du fichier .html

```
<link rel="stylesheet" type="text/css" href="style1.css" media="screen" />

<link rel="stylesheet" type="text/css" href="style1_print.css" media="print" />

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

Le fichier style1.css

```
h1{
  font-family:Arial, Helvetica, sans-serif ; /* style de la police */
  font-size:14 px ;                          /* taille de la police */
  color:#00ff00 ;                             /* couleur de la police */
  background-color:#000000 ;                  /* couleur du fond */
}
```

Solution 2 : Déclaration d'une feuille de style interne

Dans la balise <head> du fichier .html

```
<head>
  <style type="text/css">
    <!--
      h1{
        font-family:Arial, Helvetica, sans-serif ; /* style de la police */
        font-size:14 px ;                          /* taille de la police */
        color:#00ff00 ;                             /* couleur de la police */
        background-color:#000000 ;                  /* couleur du fond */
      }
    -->
  </style>
</head>
```

Moins grande portabilité

Solution 3 : Déclaration d'un style en ligne

POSSIBLE MAIS MAUVAISE PRATIQUE

```
<h2 style="text-align:center; color=red;">  
le titre de la page  
</h2>
```

Syntaxe des CSS

- Sélecteur : la balise HTML dont on modifie les propriétés ;
- Bloc de déclarations : ensemble des modifications ;
- Déclarations : propriétés + valeurs ;

Règle de syntaxe

```
sélecteur {  
    propriété :      valeur;  
    propriété :      valeur;  
}
```

Exemple d'une règle CSS

```
h1 {  
    font-size :      16pt;  
    font-family :    arial;  
}
```

Règle de syntaxe

```
body { /* l'ensemble du document aura ces propriétés */  
    color: #43aA6a;  
    background-color: white;  
    background-image:url(..fleur1.jpg);  
}  
h1 {  
    color: #e8eaff;  
    background-color: #babcec;  
}
```

Lien URL

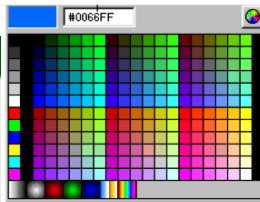
- '.' Répertoire local
- '..' Répertoire précédent
- 'http ://...' Lien externe

Codage de la couleur

- Une composante couleur (Rouge, Vert, Bleu)
- Une couleur est représentée sur 8 bits soit 256 valeurs possibles (valeur entre 0 et 255)
 - 0 absence de couleur (Noire)
 - 255 valeur maximale
- 1 couleur est le mélange de Rouge/Vert/Bleu
- [0 .. 255] -> [00 .. FF] : code en hexadécimal en html

Notation des couleurs

#FF0000	Rouge
#00FF00	Vert
#0000FF	Bleu
#ED00FF	Turquoise
#000000	Noir
#FFFFFF	Blanc



Propriétés associées aux polices

- **font-family** : la police de caractères ; valeur : une police , par exemple *Verdana, Arial, Helvetica, Geneva, ...*
- **font-size** : taille de la police ; valeur : une taille , par exemple *12px*
- **font-style** : style de la police ; valeur *normal, italic*
- **font-weight** : épaisseur de la police ; valeur : *normal, bold, bolder, lighter*

Valeurs possibles

- px : pixel
- % : pourcentage relatif à la page
- in, inch ; cm, centimètre...

Propriétés associées aux textes

- `text-decoration` : soulignement du texte ; valeur : `none`, `underline`, `overline`, `linethrough`
- `text-align` : alignement du texte ; valeur : `left`, `right`, `center`, `justify`
- `letter-spacing` : distance entre les lettres ; valeur : une distance, par exemple `1px`

Propriétés associées aux listes et aux tableaux

- Listes

- `list-style-type` : type de puces à appliquer dans la liste ; valeur : `disc`, `circle`, `square`, `decimal`, `none` ;
- `list-style-image` : spécification d'une image à utiliser comme une puce à la place des puces habituelles utilisées dans des listes ; valeur : `url(image.gif)` ;

- Tableaux

- `border-color` : couleur de la bordure ; valeur : une couleur ;
- `border-style` : forme de la bordure ; valeur : `none`, `hidden`, `dotted`, `dashed`, `solid`, `double groove`, `ridge`, `inset`, `outset`
- `border-width` : épaisseur de la bordure ; valeur : épaisseur ou `thin`, `medium`, `thick` ;

Exercice 1

- Ecrivez le code HTML et CSS pour obtenir le tableau ci-dessous :

Titre	Année	Nom	Réalisateur	Prénom	Année naissance
Alien	1979	Scott	Ridley		1943
Vertigo	1958	Hitchcock	Alfred		1899

Correction Exercice 1 (1/2)

Titre	Année	Nom Réalisateur	Prénom	Année naissance
Alien	1979	Scott	Ridley	1943
Vertigo	1958	Hitchcock	Alfred	1899

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css"
    href="td2-corrige.css" media="screen" />
  <meta charset="UTF-8">
  <title>Exercices CSS</title>
</head>
<body>
  <div><h1>Exercice 1</h1>
    <table >
      <tr><th>Titre</th><th>Année</th><th>Nom Réalisateur</th>
        <th>Prénom</th><th>Année naissance</th></tr>
      <tr><td>Alien</td><td>1979</td>
        <td>Scott</td><td>Ridley</td><td>1943</td></tr>
      <tr><td>Vertigo</td><td>1958</td>
        <td>Hitchcock</td><td>Alfred</td><td>1899</td></tr>
    </table>
  </div>
</body></html>
```

Correction Exercice 1 (2)

Titre	Année	Nom Réalisateur	Prénom	Année naissance
Alien	1979	Scott	Ridley	1943
Vertigo	1958	Hitchcock	Alfred	1899

```
table, td {
  border-color : #6495ed;
  border-top-style : solid ;
  border-bottom-style : solid ;
  border-width : thin;
  text-align : center;
}
td {
  border-left-style : solid ;
  border-right-style : solid ;
}
```

Propriétés propres aux tables

Titre	Année	Nom Réalisateur	Prénom	Année naissance
Alien	1979	Scott	Ridley	1943
Vertigo	1958	Hitchcock	Alfred	1899

- **padding** : espace entre la bordure et le contenu des cellules ; valeurs : la largeur ;
- **border-collapse** : fusionner la bordure du tour du tableau et la bordure des cases ; valeur : **collapse** pour fusionner, **separate** pour séparer les bordures.,

Pour l'exercice précédent : avec "border-collapse : collapse"

Titre	Année	Nom Réalisateur	Prénom	Année naissance
Alien	1979	Scott	Ridley	1943
Vertigo	1958	Hitchcock	Alfred	1899

- Permet d'appliquer un style précis pour une balise particulière.
- Syntaxe pour déclarer une classe dans le CSS :

```
balise.NomClasse
{
    propriété1 : valeur1;
    propriété2 : valeur2;
    ...
    propriétéN : valeurN ;
}
```

- Appel dans le HTML :

```
<balise class="NomClasse">...</balise>
```

un titre

une cellule grise	une cellule bleue	une cellule grise
-------------------	-------------------	-------------------

- HTML :

```
<h1> un titre </h1>
<table>
  <tr>
    <td class="grise"> une cellule grise </td>
    <td class="bleue"> une cellule bleue </td>
    <td class="grise"> une cellule grise </td>
  </tr>
</table>
</div>
```

- CSS :

```
td.grise {background-color : grey ;}
td.bleue {background-color : blue ;}
```

Exercice 2

	• Tennis
	• Math
	• Info
•	Mardi
	• Biologie
	• Tennis
	• Anglais
•	Mercredi
	• Tennis
	• Histoire
	• Théâtre

Créez la liste imbriquée suivante en HTML :

- Lundi : tennis, math, info ;
- Mardi : biologie, tennis, anglais ;
- Mercredi : tennis, histoire, théâtre.

En utilisant les classes, écrivez le code CSS nécessaire pour que les jours de la semaine apparaissent en bleu clair, en police arial, gras, sur un fond bleu foncé, les matières apparaissent en noir sur fond bleu clair, en police verdana, avec des images comme des puces.

Exercice 2 corrigé (1/3)

```
<ul class="jour">
  <li>Lundi
    <ol class="matiere1">
      <li>Tennis</li><li>Math</li>
      <li>Info</li> </ol> </li>
    <li>Mardi
      <ol>
        <li class="matiere2">Biologie</li>
        <li class="matiere2">Tennis</li>
        <li class="matiere2">Anglais</li>
      </ol> </li>
    <li>Mercredi
      <ol class="matiere1">
        <li>Tennis</li><li>Histoire</li><li>Théâtre</li>
      </ol>
    </li></ul>
```

- **Lundi**
 - ✪ Tennis
 - ✪ Math
 - ✪ Info
- **Mardi**
 - ✪ Biologie
 - ✪ Tennis
 - ✪ Anglais
- **Mercredi**
 - ✪ Tennis
 - ✪ Histoire
 - ✪ Théâtre

Exercice 2 corrigé (2/3)

```
ul.jour {  
  color : lightblue; /*#add8e6*/  
  background-color: darkblue; /*#00008b*/  
  font-family : "Arial";  
  font-weight : bold;  
}  
  
ol.matiere1 {  
  color : black ;  
  background-color: lightblue;  
  font-family : "Verdana";  
  font-weight : bold;  
  list-style-image : url(train.gif);  
}
```

- **Lundi**

- ✪ **Tennis**
- ✪ **Math**
- ✪ **Info**

- **Mardi**

- ✪ **Biologie**
- ✪ **Tennis**
- ✪ **Anglais**

- **Mercredi**

- ✪ **Tennis**
- ✪ **Histoire**
- ✪ **Théâtre**

- **Lundi**

- ✪ **Tennis**
- ✪ **Math**
- ✪ **Info**

- **Mardi**

- ✪ **Biologie**
- ✪ **Tennis**
- ✪ **Anglais**

- **Mercredi**

- ✪ **Tennis**
- ✪ **Histoire**
- ✪ **Théâtre**

Exercice 2 corrigé (3/3)




- **Lundi**

-  **Tennis**
-  **Math**
-  **Info**

- **Mardi**

-  **Biologie**
-  **Tennis**
-  **Anglais**

- **Mercredi**

-  **Tennis**
-  **Histoire**
-  **Théâtre**

```
li.matiere2 {  
  color : black ;  
  background-color: lightblue;  
  font-family : "Verdana";  
  font-weight : bold;  
  list-style-image : url(train.gif);  
}
```

Les identifiants

- Permet de déclarer un élément unique dans une page.
- Syntaxe pour déclarer un ID :

```
#identifiant {propriétéN : valeurN ;}
```

ou

```
balise#identifiant {propriétéN : valeurN ;}
```

- Appel dans le HTML :

```
<balise id="identifiant">...</balise>
```

Syntaxe à préférer

Les identifiants : exemple

un paragraphe normal

Texte sur fond rose, entouré d'une bordure verte.

La suite du texte.

- CSS :

```
#conteneur {  
  border : 1px solid green;  
  background-color: pink;  
}
```

- HTML :

```
<p> un paragraphe normal </p>  
<p id="conteneur"> Texte sur fond rose, entouré d'une  
bordure verte. </p>  
<p> La suite du texte. </p>
```


Exercice 3

Titre	Année	Réalisateur
La grande vadrouille	1966	Gérard Oury
Volver	2005	Pedro Almodovar
Les Infiltrés (The Departed)	2006	Martin Scorsese

Exercice 3 corrigé (1/2)

Titre	Année	Réalisateur
La grande vadrouille	1966	Gérard Oury
Volver	2005	Pedro Almodovar
Les Infiltrés The Departed	2006	Martin Scorsese

```
<table class="exercice3">
<tr>
  <th>Titre</th><th>Année</th>
  <th>Réalisateur</th>
</tr><tr>
  <td>La grande vadrouille </td>
  <td>1966</td><td>Gérard Oury</td>
</tr><tr>
  <td>Volver</td><td>2005</td>
  <td>Pedro Almodovar</td>
</tr><tr>
  <td>Les Infiltrés <span class="violet">
    (The Departed)</span>
  </td><td>2006</td><td>Martin Scorsese</td>
</tr>
</table>
```

Exercise 3 correction (2/2)

Titre	Année	Réalisateur
La grande vadrouille	1966	Gérard Oury
Volver	2005	Pedro Almodovar
Les Infiltrés The Departed	2006	Martin Scorsese

```
table.exercice3 {  
  font-family : Cambria;  
  border: medium dashed grey;  
  border-collapse : collapse;  
}  
th,td {  
  border: thin solid grey;  
  text-align: justify;  
  padding: 10px;  
}  
.violet{  
background-color : purple;  
}
```

Les marges des blocs

- Pour les balises <div>, <p>, <h1>, ...
- Exemples de marges :

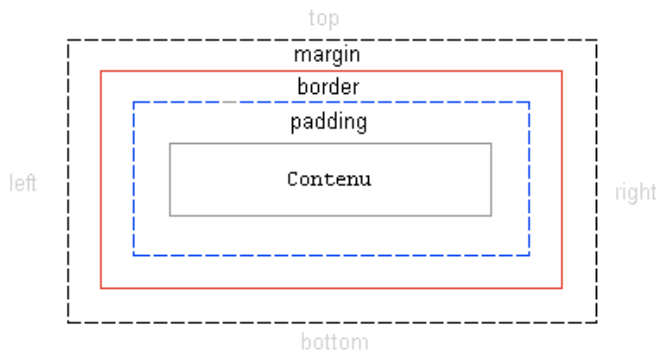
```
p {margin:0;} /*toutes les marges sont à 0 */
```

```
/* les marges de gauche et droite sont de 4 pixel */  
p {margin-left : 4px; margin-right : 4px; }
```

```
/* les marges sont calculées automatiquement et de  
   taille égale */  
p {margin-left : auto; margin-right : auto;}
```

```
/* marges haut, droite, bas puis gauche, en  
% de la largeur du bloc */  
p {  
    margin: 5% 10% 5% 10%;  
}
```

Les marges des blocs



- bord de la marge
- bord de la bordure
- - - bord de l'espacement
- bord du contenu

Positionnement des blocs : STATIC

```
.static {  
  position: static;  
}
```

CSS

```
<div class="static">
```

`static` est la valeur par défaut de tous les éléments. Un élément avec `position: static;` n'est positionné d'aucune manière spéciale. Un élément `static` est dit *non positionné* et un élément avec une propriété `position` ayant une valeur autre que `static` est dit *positionné*.

```
</div>
```

Positionnement des blocs : RELATIVE

```
.relative1 {  
  position: relative;  
}  
.relative2 {  
  position: relative;  
  top: -20px;  
  left: 20px;  
  background-color: white;  
  width: 500px;  
}
```

CSS

<div class="relative1">

relative se comporte de la même façon que static sauf si vous ajoutez quelques propriétés en plus.

</div>

<div class="relative2">

Ajouter les propriétés top, right, bottom et left à un élément positionné en relative va le placer ailleurs que sa position normale. Le reste du contenu ne sera pas ajusté pour prendre la place dans l'espace laissé par l'élément.

</div>

Positionnement des blocs : ABSOLUTE

```
.relative {  
  position: relative;  
  width: 600px;  
  height: 400px;  
}  
.absolute {  
  position: absolute;  
  top: 120px;  
  right: 0;  
  width: 300px;  
  height: 200px;  
}
```

CSS

`<div class="relative">`

Cet élément est positionné en relative. S'il était positionné en `position: static`, son élément enfant positionné en absolute s'échapperait et serait positionné de manière relative au corps du document.

`<div class="absolute">`

Cet élément est positionné en absolute. Sa position est relative à son parent.

`</div>`

`</div>`

Positionnement des blocs : FIXED

- Un élément positionné en fixed est positionné par rapport à la fenêtre du navigateur.
- Toujours à la même place même si la page défile.
- Utiliser les propriétés top, right, bottom et left.

```
.fixed {  
  position: fixed;  
  bottom: 0;  
  right: 0;  
  width: 200px;  
  background-color: white;  
}
```

CSS

```
<div class="fixed">
```

Bonjour ! Ne faites pas
attention à moi pour
l'instant.

```
</div>
```

Positionnement des blocs : exemple

```
.container {  
  position: relative;  
}  
nav {  
  position: absolute;  
  left: 0px;  
  width: 200px;  
}  
section {  
  /* position is static by default */  
  margin-left: 200px;  
}  
footer {  
  position: fixed;  
  bottom: 0;  
  left: 0;  
  height: 70px;  
  background-color: white;  
  width: 100%;  
}  
body {  
  margin-bottom: 120px;  
}
```

<nav> <div container>

- Accueil
- Menu Taco
- Projet de liste
- Horaires
- Directions
- Contact

</nav>

<section>

La propriété `margin-left` pour les `sections` assure qu'il y a de la place pour la `nav`.

</section>

<section>

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisl, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.

</section>

<section>

Remarquez ce qu'il se passe lorsque vous redimensionnez votre navigateur. Ça marche plutôt bien !

</section>

Cet exemple marche parce que l'élément conteneur est plus grand que la `nav`. Si ce n'était pas le cas, la `nav` déborderait de son conteneur. Dans les pages suivantes nous discuterons d'autres techniques de mise en page qui ont différents avantages et inconvénients.

<footer>

Si vous utilisez un header et un footer positionné en `fixed`, faites en sorte qu'il y ait de la place pour eux ! J'ajoute un `margin-bottom` sur le `body`.

</footer>

Florent Carlier, Valérie Renault (prenom.nomTechnologies du Web CSS, les feuilles de sty

42 / 42

HTML 5 : nouvelles balises

- Par exemple, l'en-tête d'une page en XHTML 1.0 s'écrit :

```
<div id="header"> Ceci est mon en-tête </div>
```

- avec le CSS suivante :

```
#header {  
  width:960px;  
  height:100px;  
  background-color:gray;  
}
```

- Le même exemple devient en HTML5 :

```
<header> Ceci est mon en-tête </header>
```

- avec la CSS suivante :

```
header {  
  width:960px;  
  height:100px;  
  background-color:gray;  
}
```

Exercice 4

- Cherchez les erreurs dans le code suivant :

```
h1 {  
  font-family: Verdana, Courier ;  
  size: 14;  
  background-color :#000000  
  font-color : #0f0;  
}
```

Exercice 4 corrigé

- Cherchez les erreurs dans le code suivant :

```
h1 {  
  font-family: Verdana, Courier ;  
  [font-]size: 14[px];  
  background-color :#000000[;]  
  [color] : [#00ff00];  
}
```

Exercice 5

- Cherchez l'erreur :

```
#bordJaune {  
  border: 1px solid yellow;  
}  
#bordRouge {  
  border: 2px thin red;  
}  
<p> Un paragraphe normal </p>  
<p id="bordJaune">Un second paragraphe</p>  
<p id="bordRouge">Un troisième paragraphe</p>  
<p id="bordJaune">Un quatrième paragraphe</p>
```

Exercice 5

- Cherchez l'erreur :

```
#bordJaune {  
  border: 1px solid yellow;  
}  
#bordRouge {  
  border: 2px thin red;  
}  
<p> Un paragraphe normal </p>  
<p id="bordJaune">Un second paragraphe</p>  
<p id="bordRouge">Un troisième paragraphe</p>  
<p id="bordJaune">Un quatrième paragraphe</p>
```

message du validateur : Duplicate ID bordJaune.

Exercice 5 corrigé

- Cherchez l'erreur :

```
.bordJaune {  
    border: 1px solid yellow;  
}  
.bordRouge {  
    border: 2px thin red;  
}  
<p> Un paragraphe normal </p>  
<p class="bordJaune">Un second paragraphe</p>  
<p class="bordRouge">Un troisième paragraphe</p>  
<p class="bordJaune">Un quatrième paragraphe</p>
```