



Université Abdelmalek Essaâdi
Ecole Nationale des Sciences Appliquées
Al Hoceima



Chapitre 3 : Le style des pages Web (CSS & Bootstrap)

Plan

- I. Introduction
- II. Sélecteurs
- III. Propriétés CSS
- IV. Les grilles en CSS
- V. Le responsive Web design
- VI. Bootstrap

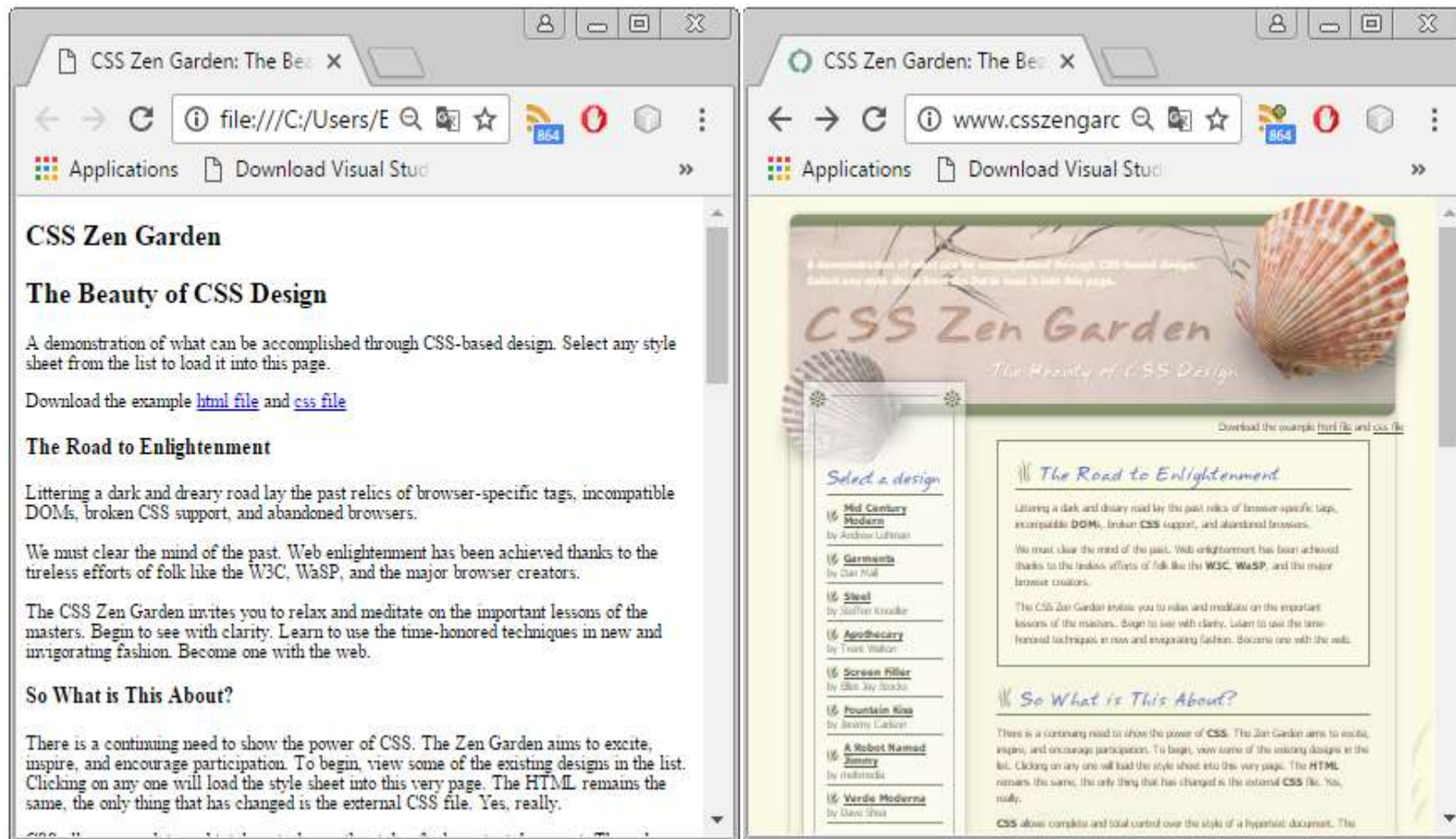
I. Introduction

1. CSS

- ▶ **Cascading Style Sheets (CSS)** est un langage de feuille de style utilisé pour définir l'apparence visuelle et la mise en page des pages web.
- ▶ HTML définit uniquement la structure du contenu, c'est le langage CSS, qui **détermine l'apparence de ce contenu**.
- ▶ CSS **décrit la façon dont les éléments doivent être affichés à l'écran**. C'est grâce à ce langage qu'on peut :
 - ▶ choisir la couleur et la taille de texte (text)
 - ▶ sélectionner la police de caractère (font)
 - ▶ définir les bordures (border), le fond
 - ▶ l'emplacement des éléments du document.
 - ▶ etc.
- ▶ Ce langage spécifié (ou défini) par le W3C **utilise des sélecteurs pour appliquer des règles de styles aux éléments HTML**.
- ▶ CSS3 est une évolution de CSS, avec des fonctionnalités avancées ajoutées à la spécification de base de CSS pour permettre aux développeurs de créer des designs plus créatifs et sophistiqués pour les pages web.

.... la suite

► Exemple (source : <http://www.csszengarden.com/>)




HTML sans CSS

HTML + CSS

2. Principe de fonctionnement

- ▶ Pour personnaliser l'apparence d'une page web, CSS utilise **des sélecteurs** pour **identifier les éléments HTML** auxquels des règles de style doivent être appliquées.
- ▶ Chaque **règle de style** est composée d'une **liste de propriétés**, qui forment une déclaration CSS. Ces déclarations sont ensuite regroupées dans des blocs de déclarations.
- ▶ La syntaxe générale pour définir une règle CSS est la suivante :

```
le sélecteur {  
    propriété1: valeurX;  
    propriété2: valeurY  
}
```

- ▶ *Propriété* : permettant de définir une fonctionnalité donnée, comme la couleur, style d'écriture, la taille, etc ...
- ▶ *Valeur* : décrit comment la fonctionnalité doit être utilisée par le navigateur
- ▶ Exemple :


Sélecteur propriété valeur

3. Notion de cascade

- ▶ Quand plusieurs règles sont mises en œuvre, **celle qui est spécifique a la priorité**. D'où, ce langage est nommé langage de feuille de style en cascade.
- ▶ La notion de « cascade » fait référence **aux règles de priorité** qui existent **entre les différents sélecteurs**.
- ▶ Un ordre de cascade : une même règle appliquée aux éléments enfants annule celle de l'élément père.

```
<body><-- élément grand-père  
  <p> </p>  
  <ul><-- élément père  
    <li> </li>  
    <li> </li>  
  </ul>  
</body>
```

Diagram illustrating the cascade of styles:

- <-- élément grand-père** (body)
- <-- éléments voisins** (p, ul)
- <-- deux éléments enfants** (li, li)

```
ul{  
  color: blue;  
}  
li{  
  color: red;  
}
```

Dans ce cas c'est la couleur rouge qui sera appliquée.

4. Liaison HTML-CSS

- ▶ Il est possible d'écrire le code CSS directement dans les balises `<style>`, mais il est plus adapté de créer des fichiers `.css` séparés de façon à réutiliser les informations de mise en forme sur d'autres pages.
- ▶ Il existe trois différents endroits où il est possible de mettre du code CSS:

- ▶ **Méthode 1:** directement dans l'en-tête `<head>` du fichier HTML. Cela consiste à insérer le code CSS directement dans une balise `<style>` à l'intérieur de l'en-tête `<head>`. Un exemple :

```
<head>  
  <style> p { color: red; } </style>  
</head>
```

- ▶ **Méthode 2:** directement dans les balises du fichier HTML via un attribut `style` (méthode la moins recommandée). Un exemple :

```
<body>  
  <p style="color: blue;">Bonjour et bienvenue sur ce site !</p>  
</body>
```

.... la suite

- ▶ **Méthode 3:** dans **un fichier séparé portant l'extension .css**. *Il s'agit de la méthode la plus recommandée.* Les avantages de cette méthode sont :
 - ▶ *Eviter la redondance* : le style défini dans un fichier de façon séparé peut être appliqué sur plusieurs document html, par contre, pour les deux méthodes précédentes le style concerne que le document sur lequel le style est appliqué.
 - ▶ *Eviter de tout mélanger dans un même fichier.*
- ▶ La liaison d'un document html avec un fichier .css se fait grâce à l'attribut href de l'élément `<link>`. Exemple : application de la couleur rouge sur les paragraphes d'un document(nommé test.html). A noter que le fichier du style (nommé style.css) et le document se trouvent dans le même dossier.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<link rel="stylesheet" type="text/css" href="style.css">
<title>Style CSS </title>
</head>
<body>
  <h1>Pour tester CSS</h1>
  <p>Bonjour et bienvenue sur ce site !</p>
</body>
</html>
```

test.html

```
p{
  color: red;
}
```

style.css



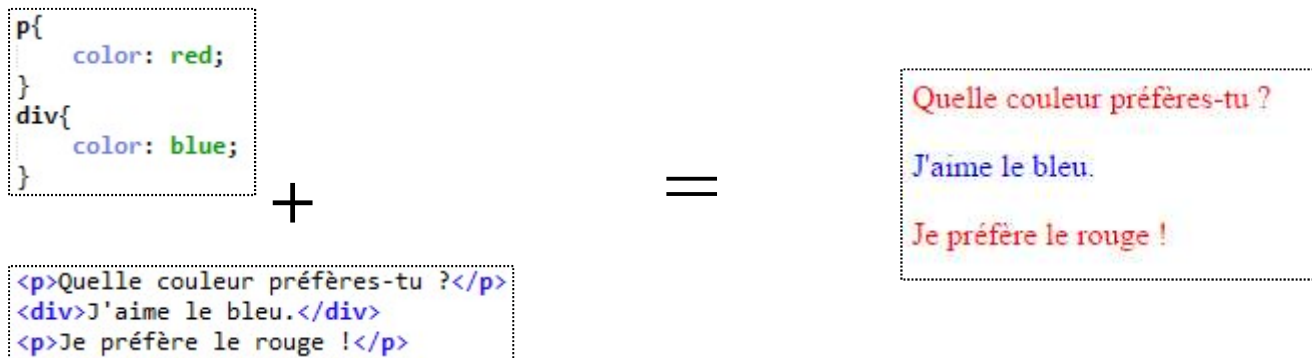
II. Les sélecteurs

1. Présentation

- ▶ Le rôle d'un sélecteur en CSS est de **localiser un ou plusieurs éléments HTML** pour leur appliquer des règles de style. Il existe une variété de méthodes pour cibler et localiser les éléments HTML sur une page web, et on peut les regrouper en deux types :
 - ▶ Les sélecteurs simples :
 - ▶ en fonction du nom d'un élément,
 - ▶ en fonction de la valeur d'un attribut d'un élément,
 - ▶ en fonction d'une *pseudo-classe*,
 - ▶ de façon globale.
 - ▶ Les sélecteurs combinés :
 - ▶ en fonction de la hiérarchie d'un élément,
 - ▶ de la position d'un élément.

2. Les sélecteurs de type (*type selectors*)

- ▶ Aussi appelés sélecteurs d'élément, ces sélecteurs correspondant aux éléments HTML (exemple `<p>`).
- ▶ C'est la méthode la plus simple pour cibler tous les éléments d'un type donné. Prenons un exemple :



- ▶ Son inconvénient est que ‘par exemple’ tous les paragraphes possèdent la même présentation ici, ils seront donc tous écrits en rouge.
- ▶ La question est comment faire pour que certains paragraphes seulement soient écrits d'une manière différente.

3. Les sélecteurs de classe et d'identifiant

- Pour résoudre le problème liée à l'utilisation de nom de l'élément html, on peut utiliser deux attributs spéciaux *qui fonctionnent sur toutes les balises* :
 - L'attribut « class » : ceci permet de définir un sélecteur de classe. Ce dernier est composé d'un point (.), suivi d'un nom de classe.
 - L'attribut « id » : ceci permet de définir un sélecteur d'identifiant. Ce dernier commence par un dièse (#), suivi par le nom de l'identifiant attribué à un élément.

► Exemple :

```
<p class="pragraphe1">je suis le premier paragraphe</p>  
<p class="pragraphe2">je suis le deuxieme paragraphe</p>
```



```
.pragraphe1{  
    color: red;  
}  
.pragraphe2{  
    color: blue;  
}
```

```
<p id="pragraphe1">je suis le premier paragraphe</p>  
<p id="pragraphe2">je suis le deuxieme paragraphe</p>
```



```
#pragraphe1{  
    color: red;  
}  
#pragraphe2{  
    color: blue;  
}
```

4. Les sélecteurs d'attribut et leurs valeurs

- ▶ Les sélecteurs d'attribut permettent de sélectionner les éléments HTML en fonction de noms de leurs attributs et des valeurs de ceux-ci.
- ▶ Pour utiliser ces sélecteurs, on écrira des crochets "[" dans lesquels on place le nom de l'attribut et éventuellement une condition sur la valeur de l'attribut. Les sélecteurs d'attributs peuvent être classés en deux catégories :
 - ▶ Les sélecteurs d'attribut avec ou sans valeur :
 - ▶ **[attr]**: sélectionne tous les éléments avec l'attribut attr, quelque soit sa valeur.
 - ▶ **[attr=val]**: sélectionne tous les éléments avec l'attribut attr, mais seulement si la valeur est égale à val.
 - ▶ **[attr~=val]**: sélectionne tous les éléments dont la valeur de l'attribut attr est une liste de mots séparés par des espaces, dont l'un est exactement "val".
 - ▶ Les sélecteurs d'attribut utilisant un filtre sur les fragments de chaînes:
 - ▶ **[attr^=val]**: sélectionne tous les éléments dont la valeur de l'attribut attr commence par val.
 - ▶ **[attr\$=val]**: sélectionne tous les éléments dont la valeur de l'attribut attr finit avec val.
 - ▶ **[attr*=val]**: sélectionne tous les éléments dont la valeur de l'attribut attr contient la chaîne val.
 - ▶ **[attr|=val]**: sélectionne tous les éléments dont l'attribut attr vaut val ou commence par val suivi de -. Exemple :

.... la suite

► Exemples :

```
a[title]
{
  /* Sélectionne tous les liens <a> qui possèdent un attribut title.*/
}

a[title="Cliquez ici"]
{
  /* dans ce cas l'attribut doit en plus avoir exactement pour valeur « Cliquez ici »./
}
```

Les modules de deuxième semestre :

```
<ul>
<li module-obligatoire="Développement">Programmation Orientée Objet en C++ </li>
<li module-obligatoire="Mathématiques et Informatique">Algorithmique avancée et complexité </li>
<li module-obligatoire>Modélisation avec UML </li>
<li module-obligatoire="Mathématiques et Informatique">Recherche Opérationnelle</li>
<li module-obligatoire="Développement">Développement d'applications Web</li>
<li module-optionnel>Techniques et économie de l'entreprise </li>
</ul>
```

+

```
[module-obligatoire] {
  color: green
}
[module-obligatoire=Développement] {
  color: goldenrod;
}
[module-obligatoire~Informatique] {
  color: red;
}
```

=

Les modules de deuxième semestre :

- Programmation Orientée Objet en C++
- Algorithmique avancée et complexité
- Modélisation avec UML
- Recherche Opérationnelle
- Développement d'applications Web
- Techniques et économie de l'entreprise

5. Sélecteur pseudo-classes

- ▶ Une pseudo-classe est utilisée pour définir un état spécial d'un élément. **Le style s'applique en fonction de l'état de l'élément.**
- ▶ Par exemple, il peut être utilisé pour **appliquer de style quand un utilisateur passe la souris sur l'élément.**
- ▶ Ce sont **des mots-clés précédés par deux points (:)** et qui sont ajoutés aux sélecteurs. **La syntaxe est** `:selector:pseudo-class {
property:value;
}`
- ▶ Voici une liste non-exhaustive :
 - **:*hover*** : cibler un élément lorsqu'il est survolé par la souris.
 - **:*active*** : sélectionne l'élément actif
 - **:*visited*** : sélectionne tous les liens visités
 - **:*first-child*** : sélectionne l'élément qui est le premier enfant de son parent. Il existe aussi, **:*last-child***, **:*only-child***, **:*nth-child(n)*** . Ce dernier peut être utilisé de plusieurs façon en changeant *a* et *b* dans **:*nth-child(an+b)***.
 - **:*not(selector)*** : Sélectionne tous les éléments sauf celui entre ().

.... la suite

► Exemples :

- Exemple 1 : le code suivant permet de changer la couleur d'un élément `<div>` lorsqu'on passe la souris dessus.

```
div {  
  background-color: green;  
  color: white;  
  padding: 25px;  
  text-align: center;  
}  
div:hover {  
  background-color: blue;  
}
```



Couleur initiale



Souris dessus

- Exemple 2 :

```
p:first-child {  
  color: blue;  
}
```

+ `<p>paragraphe 1 </p>` = paragraphe 1
`<p>paragraphe 2 </p>` = paragraphe 2

- Exemple 3 :

```
p:nth-child(2n+1) {  
  color: blue;  
}
```

+ `<p>paragraphe 1 </p>` = paragraphe 1
`<p>paragraphe 2 </p>` = paragraphe 2
`<p>paragraphe 3 </p>` = paragraphe 3
`<p>paragraphe 4 </p>` = paragraphe 4

6. Les pseudo-éléments

- ▶ Les pseudo-éléments ressemblent beaucoup aux pseudo-classes : ce sont des mots-clés précédés par deux deux-points (:) et qui sont ajoutés aux sélecteurs. Ils permettent de cibler des parties de l'élément concerné.
- ▶ On peut voir cela comme un « emplacement » par rapport à l'élément sélectionné : *::after*, *::before*, *::first-letter*, *::first-line*, *::selection*, ...

```
p.intro::first-letter {  
  color: red;  
  font-size: 200%;  
}
```

+ `<p class="intro"> Cette introduction ...</p>`

= Cette introduction ...

```
a {  
  color: red;  
  font-weight: bold;  
  text-decoration: none;  
}  
[href*=ensah]::after {  
  content: '-->';  
}
```

+ `cliquer `

= cliquer -->

```
p::selection {  
  color: red;  
  background: yellow;  
}
```

Le style s'applique sur le texte qu'on est en train de le sélectionner.

7. Les combinateurs

- ▶ CSS donne la possibilité de **combiner les sélecteurs pour obtenir un résultat précis**. Selon les relations entre les éléments, CSS permet de combiner les sélections. Ces relations sont exprimées sous la forme « combinateurs ».
- ▶ **Exemple** : `h3+p{ }`, le style concerne la première balise `<p>` située après un titre `<h3>`.
- ▶ Dans le tableau qui suit, A et B représentent n'importe quel sélecteur :

	Combinateur	Élément(s) sélectionné(s)
	A, B	Tout élément correspondant à A et à B
B inclus dans A	A B	Tout élément correspondant à B et qui est un descendant d'un élément correspondant à A (c'est-à-dire que l'élément correspondant à B sera un fils (voire un fils d'un fils, voire un fils d'un fils d'un fils...) d'un élément correspondant à A.
	A > B	Tout élément correspondant à B et qui est un fils direct d'un élément correspondant à A
B voisin de A	A + B	Tout élément correspondant à B et qui est le prochain voisin d'un élément correspondant à A (c'est-à-dire le prochain fils du même parent)
	A ~ B	Tout élément correspondant à B et qui est un voisin d'un élément correspondant à A (c'est-à-dire un des fils du même parent)

.... la suite

► Exemples :

<pre> Liste item1 Liste item2 Liste item2-1 Liste item2-2 Liste item3 </pre>	<pre>ul li { color: red; }</pre> <p>Toutes les items listes auront une couleur rouge y compris les li de (ol)</p>
	<pre>ul > li { color: red; }</pre> <p>Seuls les listes Item 1, 2 et 3 auront une couleur rouge, car ils sont les enfants direct de ul, alors que les items 2-1 et 2-2 sont ses petits-enfants.</p>
<pre><h1>Titre 1 </h1> <p>Paragraphe 1</p> <p>Paragraphe 2</p> <p>Paragraphe 3</p></pre>	<pre>h1+p { font-size: 1.5em; font-family: arial; } h1~p { font-size: 1.5em; font-family: arial; }</pre> <p>Seul le paragraphe 1 aura le style spécifié.</p> <p>Tous les paragraphes auront le même style spécifié.</p>
<pre>ul,h1{ font-weight: bold; }</pre>	<p>Tous les éléments de la liste et tous les titres h1 auront le style spécifié</p>

8. Le sélecteur universel

- ▶ Le sélecteur universel, représenté par *, est le plus large. Il permet de sélectionner tous les éléments d'une page. Il est rarement utile d'appliquer une même mise en forme sur toute une page.
- ▶ Exemple :

```
<h1>Pour tester CSS</h1>  
<p class="paragraphe1"> Je suis le premier paragraphe</p>  
<p class="paragraphe2"> Je suis le deuxieme paragraphe</p>  
<p class="paragraphe3"> Je suis le troisième paragraphe</p>
```

+

```
*{  
    color:red;  
}  
.paragraphe1{  
    color:blue;  
}  
.paragraphe2{  
    font-weight: bold  
}  
p{  
    text-decoration: underline;  
}
```

=

Pour tester CSS

Je suis le premier paragraphe

Je suis le deuxieme paragraphe

Je suis le deuxieme paragraphe