BDW - Programmation web - CSS

Fabien Duchateau

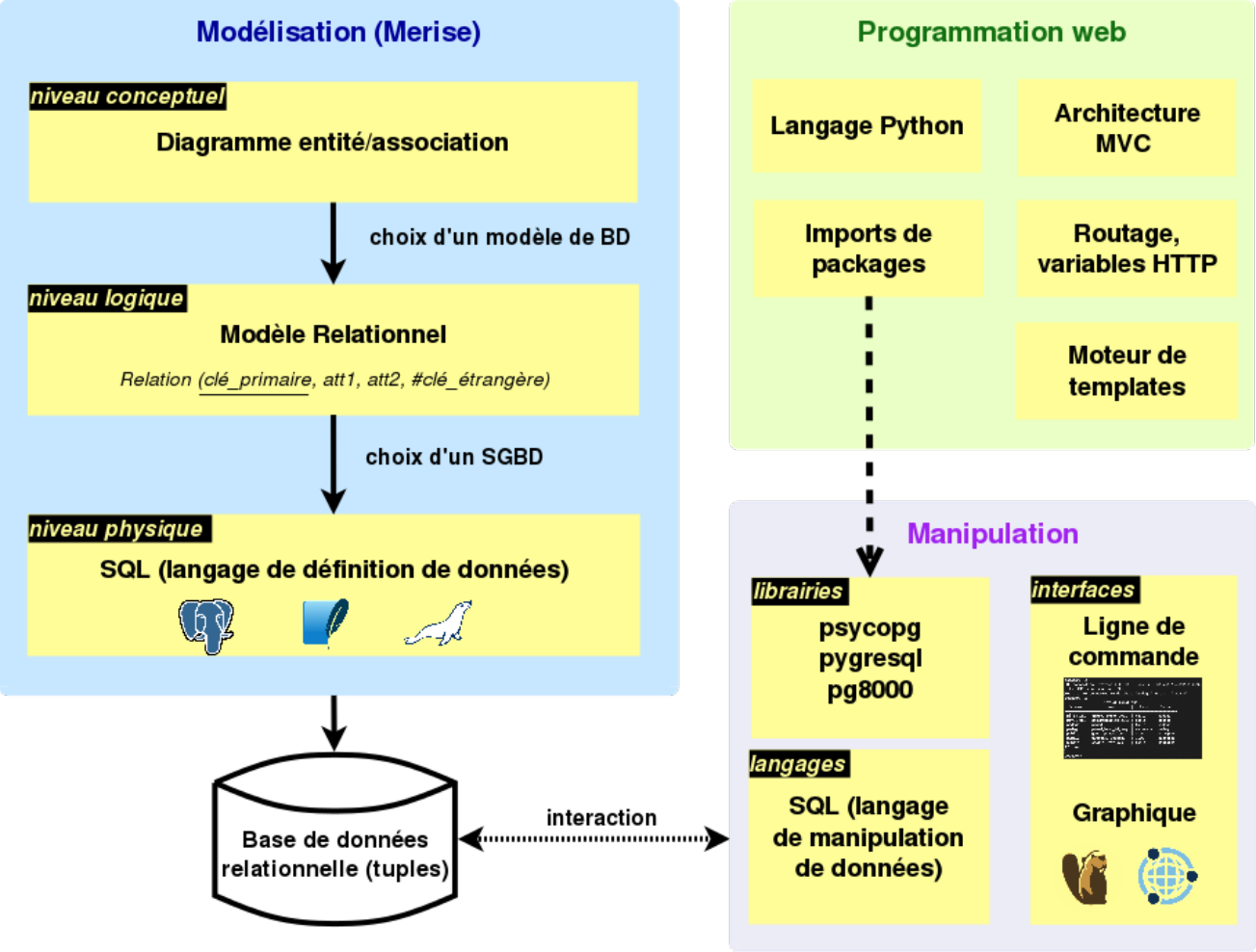
fabien.duchateau [at] univ-lyon1.fr

Université Claude Bernard Lyon 1

2024 - 2025

https://perso.liris.cnrs.fr/fabien.duchateau/BDW/

Positionnement dans BDW



Ces diapositives utilisent le genre féminin (e.g., chercheuse, développeuses) plutôt que l'écriture inclusive (moins accessible, moins concise, et pas totalement inclusive)

Plan

Le CSS

Concepts

Sélecteurs

Généralités

Avant, mise en forme et mise en page mélangées avec la structuration des données (balises obsolètes `<font>`, etc.)

Aujourd'hui, séparation entre structuration des données et mise en forme / mise en page :

- Simplification et lisibilité du code (maintenance)
- Uniformisation du code entre les pages du même site et entre différents médias (e.g., impression, écran)
- Améliorer l'accessibilité
- Facilité de portage pour un autre site

Généralités

Avant, mise en forme et mise en page mélangées avec la structuration des données (balises obsolètes `<font>`, etc.)

Aujourd'hui, séparation entre structuration des données et mise en forme / mise en page :

- Simplification et lisibilité du code (maintenance)
- Uniformisation du code entre les pages du même site et entre différents médias (e.g., impression, écran)
- Améliorer l'accessibilité
- Facilité de portage pour un autre site

Mise en forme/page avec les Cascading Style Sheets

Cascading Style Sheets

CSS pour Cascading Style Sheets :

- Langage de présentation/de rendu des données
- Origine : 1994-1995 (standard W3C en 1996)
- Toujours en développement (CSS3 en 2010, et depuis chaque module évolue indépendamment)
- Extension d'un fichier CSS : `.css`

---

http://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade
https://www.w3.org/TR/css-2021/

Une page web sans CSS, puis avec CSS

En tête

Menu

- Partie 1
- Partie 2
- Partie 3
- Partie 4

Titre de la section

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce nec elit condimentum, ullamcorper purus at, pulvinar ante. Ut pellentesque convallis dui eu suscipit. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Phasellus et venenatis sapien, ac rhoncus felis. Cras posuere sodales aliquet. Curabitur posuere nulla et ante faucibus lobortis. Integer quis rutrum justo. Vestibulum egestas ante vitae purus dapibus, eu dapibus risus suscipit.
Sed vel sodales massa. Proin in gravida leo. Morbi eget laoreet nunc. Etiam auctor feugiat ante ac vulputate. Curabitur id auctor dui. Cras ac bibendum metus, a commodo velit. Proin eleifend facilisis interdum. Aliquam semper suscipit nibh, fringilla molestie leo eleifend id. Quisque ac luctus tortor. Nulla et tincidunt turpis. Phasellus sagittis eros magna, id dictum elit commodo sit amet. Fusce quis eros mi.

Pied de Page - Texte généré par Lipsum

En tête

Menu

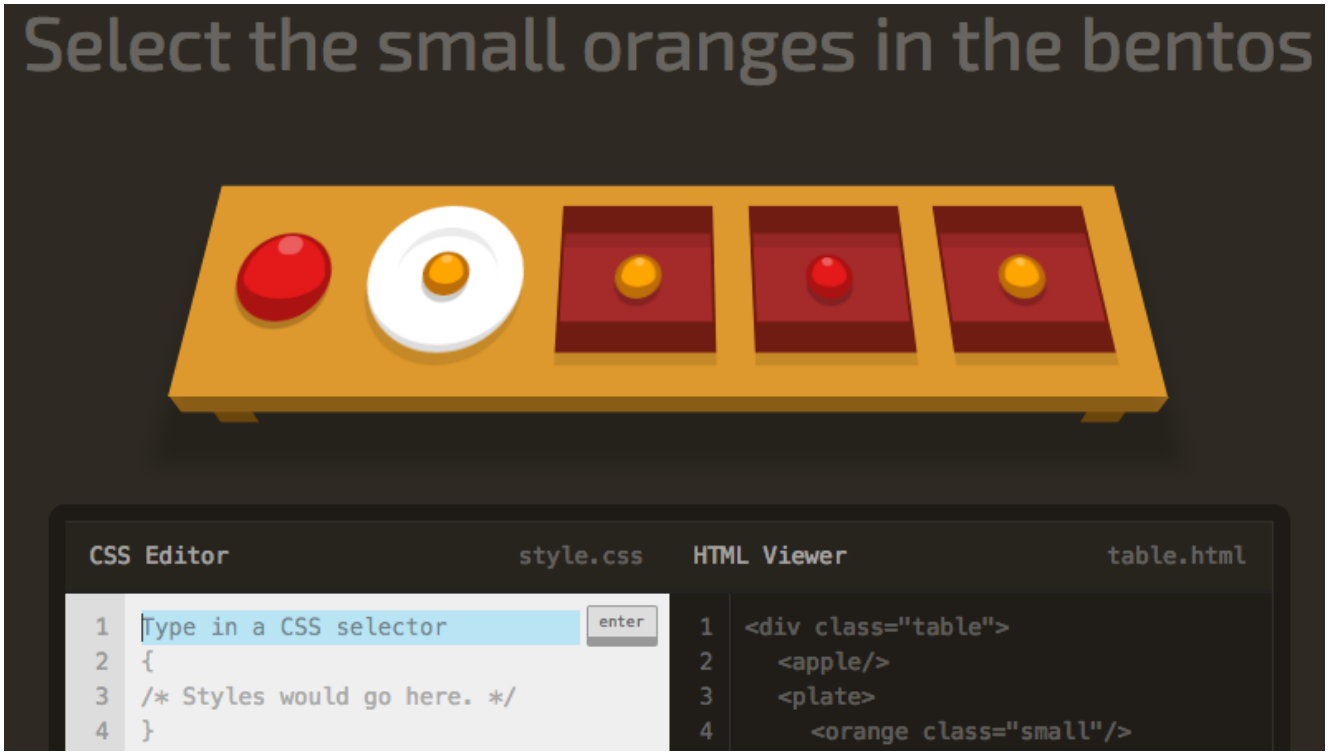
- Partie 1
- Partie 2
- Partie 3
- Partie 4

Titre de la section

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce nec elit condimentum, ullamcorper purus at, pulvinar ante. Ut pellentesque convallis dui eu suscipit. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Phasellus et venenatis sapien, ac rhoncus felis. Cras posuere sodales aliquet. Curabitur posuere nulla et ante faucibus lobortis. Integer quis rutrum justo. Vestibulum egestas ante vitae purus dapibus, eu dapibus risus suscipit.
Sed vel sodales massa. Proin in gravida leo. Morbi eget laoreet nunc. Etiam auctor feugiat ante ac vulputate. Curabitur id auctor dui. Cras ac bibendum metus, a commodo velit. Proin eleifend facilisis interdum. Aliquam semper suscipit nibh, fringilla molestie leo eleifend id. Quisque ac luctus tortor. Nulla et tincidunt turpis. Phasellus sagittis eros magna, id dictum elit commodo sit amet. Fusce quis eros mi.

Pied de Page - Texte généré par Lipsum

Quelques ressources



Des tutoriels "jeux" pour apprendre le CSS, comme Flukeout

---

https://flukeout.github.io/
https://developer.mozilla.org/en-US/docs/Learn/CSS
http://www.w3schools.com/css/
http://css.mammouthland.net/

Plan

Le CSS

Concepts

Sélecteurs

Généralités

Le CSS permet de personnaliser certains éléments d'une page web en leur appliquant un style

- Où insérer du CSS ?
- Comment écrire un style CSS ?
- Quel est l'impact des styles en cascade et de l'héritage ?

Cinq niveaux d'insertion de code CSS

- Dans une balise, via un attribut (**inline**)
- Dans un script intégré dans l'entête de la page web (**interne**)
- Dans une **feuille externe**
- Dans **plusieurs feuilles externes**
- Dans une **feuille utilisatrice** (cas très rare)

Niveau d'insertion inline

CSS inline : déclaration avec l'attribut `style` dans la balise sur laquelle le style s'applique

```
<balise style="...">...</balise>
```

```
<body style="background-color: #CCEEBB;">
```

Niveau d'insertion interne

CSS interne : déclaration dans l'entête `<head>` de la page HTML avec la balise `<style>`

```
<head>
<style>
...
</style>
</head>
```

```
<style type="text/css">
body {
  background-color: #CCEEBB;
}
...
</style>
```

Niveau d'insertion externe (une feuille)

CSS dans une seule feuille externe : la page HTML et sa feuille CSS externe sont liées avec une balise `<link>` (dans l'entête `<head>` de la page HTML)

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

Extrait de demo.css

```
body {
  background-color: #CCEEBB;
}
...
```

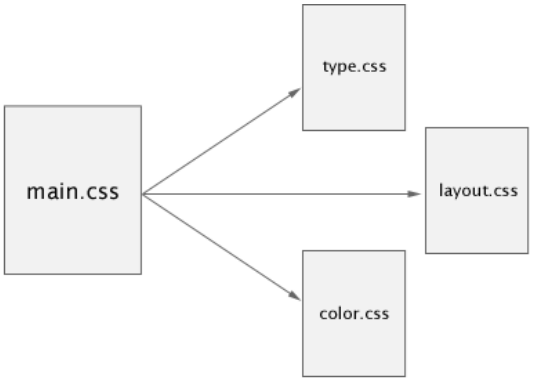
Extrait de index.html

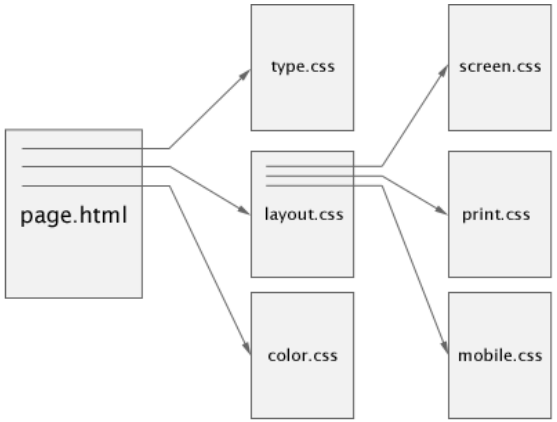
```
<link href="demo.css" rel="stylesheet" media="all"
↪  type="text/css">
```

Niveau d'insertion externe (plusieurs feuilles)

CSS dans plusieurs feuilles externes : la page HTML et ses feuilles CSS externes sont liées avec des balises `<link>` (dans l'entête `<head>` de la page HTML)

```
<link rel="stylesheet" type="text/css" href="types.css">
<link rel="stylesheet" type="text/css" href="layout.css">
<link rel="stylesheet" type="text/css" href="color.css">
```





http://www.alistapart.com/articles/progressiveenhancementwithcss/

Niveau d'insertion recommandé

En théorie, utilisation de plusieurs feuilles externes (e.g., pour chaque média)

En pratique, utilisation d'**une seule feuille externe** :

- Bénéficie des avantages du CSS (séparation données/mise en forme, facilité de maintenance, etc.)
- Moins complexe que l'utilisation de plusieurs feuilles
- Possibilité de découper en plusieurs feuilles externes plus tard
- Limiter l'utilisation des niveaux d'insertion inline et interne

Syntaxe du CSS

Un style est un **ensemble de paires sous la forme *(propriété, valeur)***

En inline, le style s'applique à la balise courante

<balise style="propriété$_1$: valeur$_1$; ... ; propriété$_n$: valeur$_n$;">
...</balise>

```
<p style="background-color: #2277FF; margin: 1em;">...</p>
```

Syntaxe du CSS interne ou externe

En insertion interne ou externe, il faut spécifier sur quels éléments s'applique chaque style : c'est le rôle du sélecteur

sélecteur {
  propriété$_1$: valeur$_1$;
  ...
  propriété$_n$: valeur$_n$;
}

```
#id-paragraphe {
  background-color: #2277FF;
  margin-right: 1em;
}
```

Notion de cascades

Les styles sont "en cascades", i.e., les déclarations de styles externe, interne et/ou inline se combinent

- Priorité accordée à la déclaration la plus proche de l'élément à styliser, en cas de conflit sur une même propriété
- Ordre de priorité (liste par priorité décroissante) :
  [feuille utilisatrice,] inline, interne, externe

Exemple : si on déclare une balise *paragraphe* avec des propriétés de style *fond rouge, alignement centré*, et qu'une feuille externe contraint les balises paragraphes au style *fond vert, police taille 12*, quel est le style du paragraphe ?

Notion de cascades

Les styles sont "en cascades", i.e., les déclarations de styles externe, interne et/ou inline se combinent

- Priorité accordée à la déclaration la plus proche de l'élément à styliser, en cas de conflit sur une même propriété
- Ordre de priorité (liste par priorité décroissante) :
  [feuille utilisatrice,] inline, interne, externe

Exemple : si on déclare une balise *paragraphe* avec des propriétés de style *fond rouge, alignement centré*, et qu'une feuille externe contraint les balises paragraphes au style *fond vert, police taille 12*, quel est le style du paragraphe ?
⇒ fond rouge, alignement centré, police taille 12

Notion d'héritage

Imbrication des éléments HTML :

- La balise `<html>` contient les balises `<head>` et `<body>`, la balise `<body>` contient des balises `<main>`, `<header>`, etc.

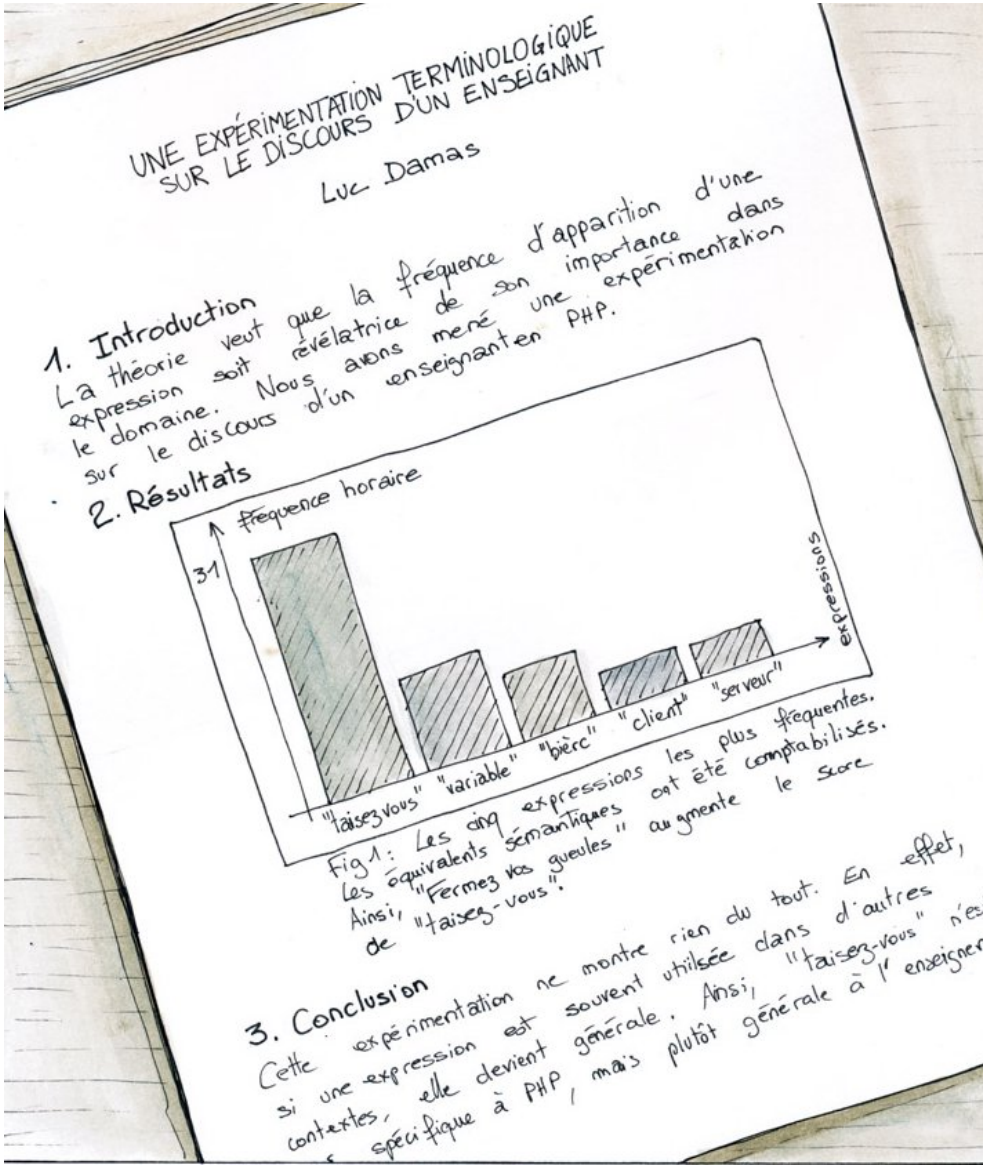
En CSS, un élément imbriqué **hérite des propriétés de ses ancêtres**, sauf les propriétés redéfinies pour cet élément

Exemple : si on déclare une balise `<body>` avec une *couleur de fond bleue*, un paragraphe sans style spécifique et dont le parent direct est la balise `<body>` aura une couleur de fond bleue

Commentaires en CSS

Commentaires multi-lignes avec la notation /* … */

```
/*
   Fichier de style styles.css
   Version septembre 2024
*/
p {
/* paragraphes en vert */
   color : green ;
}
```


UNE EXPÉRIMENTATION TERMINOLOGIQUE SUR LE DISCOURS D'UN ENSEIGNANT

Luc Damas

1. Introduction

La théorie veut que la fréquence d'apparition d'une expression soit révélatrice de son importance dans le domaine. Nous avons mené une expérimentation sur le discours d'un enseignant en PHP.

2. Résultats

Fig 1 : Les cinq expressions les plus fréquentes.

Les équivalents sémantiques ont été comptabilisés. Ainsi, "Fermez vos gueules" augmente le score de "taisez-vous".

3. Conclusion

Cette expérimentation ne montre rien du tout. En effet, si une expression est souvent utilisée dans d'autres contextes, elle devient générale. Ainsi, "taisez-vous" n'est spécifique à PHP, mais plutôt générale à l'enseigne

En résumé

- Utiliser une feuille de style externe
- Un style = des paires (propriété : valeur)
- Cascade : plusieurs styles peuvent s'appliquer à un élément
- Héritage : un élément hérite des propriétés de ses ancêtres

Plan

Le CSS

Concepts

Sélecteurs

Sélecteurs

Sélecteur : motif ("pattern") qui sélectionne le ou les éléments sur lesquels va s'appliquer un style

Quelques types de sélecteur :

- Une balise
- Un nom de classe (introduit par `.`)
- Un identifiant d'élement (introduit par #)
- Une combinaison d'éléments (e.g., les balises imbriquées dans une autre)
- Des éléments qui satisfont une condition (e.g., possédant un certain attribut)
- ...

Sélecteur de type balise

balise {
  propriété$_1$: valeur$_1$;
  ...
  propriété$_n$: valeur$_n$;
}

- Le style défini par les propriétés propriété$_1$, ..., propriété$_n$ est appliqué à tous les éléments dont le nom de balise est *balise*

Sélecteur de type classe (.)

.classe {
  propriété$_1$: valeur$_1$;
  ...
  propriété$_n$: valeur$_n$;
}

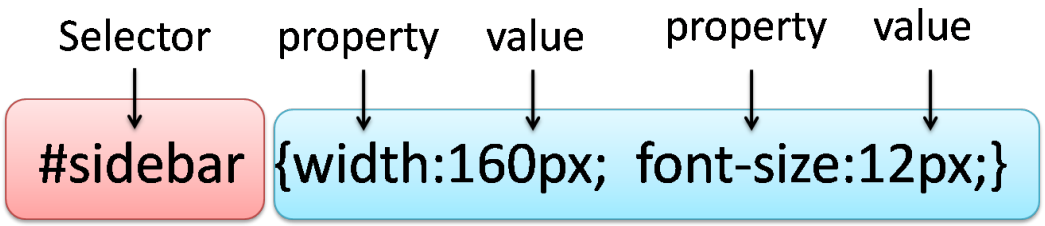
- Pour rappel, une balise HTML peut avoir un attribut `class` qui contient une ou plusieurs classes
- Le style défini par les propriétés propriété$_1$, ..., propriété$_n$ est appliqué à tous les éléments dont l'attribut `class` possède la valeur *classe*

```
.ninja {
   color: #000000;
   visibility:hidden;
}
```

Sélecteur de type identifiant (#)

#identifiant {
  propriété$_1$: valeur$_1$;
  ...
  propriété$_n$: valeur$_n$;
}

- Pour rappel, une balise HTML peut avoir un attribut `id` qui identifie l'élément sans ambiguïté dans la page
- Le style défini par les propriétés propriété$_1$, ..., propriété$_n$ est appliqué à tous les éléments dont l'attribut `id` a pour valeur *identifiant*



Sélecteurs basés sur la structure

/* tous les éléments <*balise*$_1$> et <*balise*$_2$> */
balise$_1$, balise$_2$ { ... }

/* tous les éléments *balise*$_2$ imbriqués dans un élément *balise*$_1$ */
balise$_1$ balise$_2$ { ... }

/* tous les éléments *balise*$_2$ dont le parent est un élément *balise*$_1$ */
balise$_1$ > balise$_2$ { ... }

/* tous les éléments possédant une classe *uneClasse* qui sont imbriqués dans un élément *balise*$_1$ */
balise$_1$ .uneClasse { ... }

Sélecteurs de type attribut

/* tous les éléments *balise* ayant l'attribut *attrib* */
balise [attrib] { ... }

/* tous les éléments *balise* ayant un attribut *attrib* dont la valeur vaut *val* */
balise [attrib = "val"] { ... }

/* tous les éléments *balise* ayant un attribut *attrib* dont la valeur contient la chaîne *sous-chaine* */
balise [attrib *= "sous-chaine"] { ... }

Priorités des styles en CSS

Plusieurs niveaux d'insertion, plusieurs types de sélecteurs... qui a la priorité ? Liste du plus prioritaire au moins prioritaire :

- Style dans la balise HTML (e.g., *<h1 style="color : black">*)
- Utilisation d'un identifiant (e.g., *h1#id {color : black ;}*)
- Utilisation d'une classe (e.g., *.noir {color : black ;}*)
- Imbrication d'éléments (e.g., *div h1 {color : black ;}*)
- Élément HTML (e.g., *h1 {color : black ;}*)

Exception avec **la mention** `!important` en fin de déclaration (qui devient la déclaration à appliquer) :

propriété : valeur !important ;

En résumé

- Déclaration CSS : `sélecteur {propriété: valeur;}`
- Nombreux types de sélecteurs, notamment les pseudo-classes (e.g., élément survolé, lien visité)
- Priorité des styles selon le niveau de CSS et le type de sélecteur

http://www.w3.org/Style/css3-selectors-updates/WD-css3-selectors-20010126.fr.html#selectors

http://www.w3schools.com/cssref/css_selectors.asp

http://wumo.com/wumo