CSS

Cascading StyleSheets, feuilles de styles en cascade

Rôle de CSS

Définir l’apparence des éléments d’une page web

Les outils que l’on va utiliser

- Comment activer l’inspecteur dans Safari;
- Comment fonctionne l’inspecteur de Firefox;
- Chrome *devtools*;
- Web Developper, un ensemble d’outils pour la conception web;
- Visbug pour Chrome, Visbug pour Firefox et même pour Safari

Les avantages

- **Découplage** : HTML gère le contenu, CSS l'apparence;
- **Maintenabilité** : Une seule page HTML peut avoir des formes visuelles différentes;
- **Ré-utilisabilité** : un seul document CSS permet de gérer toutes les pages d'un site mais aussi d'autres sites;
- Standard et utilisable sur une large palette de *devices*.

Syntaxe

```CSS
selecteur {
  propriété: valeur
}
```

- **Sélecteur**, les éléments HTML sur lesquels les déclarations s'appliquent;
- **Propriété**, caractéristique visuelle (la police, la couleur du text, le fond…);
- **Valeur**, la configuration de la propriété, par exemple `serif`, `red`, `#cacaca`;
- La propriété et la valeur sont séparées par un `:`.

Cet ensemble se nomme un *bloc de déclarations*.

Syntaxe, déclarations, commentaires

CSS

```css
/*
  Chaque bloc de déclarations
  commence par { et se finit par }
*/

p {
  /* Chaque déclaration est séparée par un `;` */
  color: blue;
  font-weight: bold;
}
.selecteur {
  color: #fc0;
}
```

Et comment l'utiliser ?

1. Styles externes dans un fichier lié au document `html` ;
2. Styles *inline* ;
3. Styles *internes*.

Fichier de styles externes

Pratique idéale, permet de séparer le contenu et l'aspect visuel et d'utiliser un seul document CSS pour plusieurs pages.

HTML

```html
<head>
  <link rel="stylesheet" href="monfichier.css">
</head>
<body>
  <p>Test</p>
</body>
```

CSS

```css
/* monfichier.css */
p {color: orange}
```

Autres méthodes

Méthodes existantes mais **à éviter autant que possible**.

HTML

```html
<!-- Styles inline-->
<p style="color: orange">Exemple de styles inline</p>
```

HTML

```html
<!-- Styles internes -->
<p>Exemple de styles internes</p>
<style>
  p {color: orange}
</style>
```

Essayons toutes ces méthodes

Dans un nouveau projet, faites trois pages html et pour chacune essayez d’appliquer des styles avec ces trois méthodes.

La sélection en CSS

Les sélecteurs basiques

- Par nom d'élément (`body,p,h1,h2…,section`);
- Par `class` (`.class`);
- Par `id` (`#id`);

La sélection par nom d'élément

CSS

```css
body {
  margin: 0;
  background-color: white;
}
p {
  color: orange
}
```

- Le `body` sera sans marge et son fond sera blanc;
- Tous les `p` seront oranges.

La sélection par `class`

CSS

```css
.big {
  font-size: 3rem
}
.small {
  font-size: 1rem
}
```

HTML

```html
<p class="big">Ce texte sera grand</p>
<p class="small">Ce texte sera plus petit</p>
<section class="small"><p>Ce texte sera plus petit aussi</p></section>
```

Permet de sélectionner des ensembles d'éléments portant un même attribut `class`.

La sélection par `id`

CSS

```css
#header-logo { color: blue }
#footer-logo { color: orange }
```

HTML

```html
<!-- L'id est unique : un seul élément avec un même id par page -->
<h2 id="header-logo">Logo entête</h2>
<h2 id="footer-logo">Logo pied de page</h2>
```

À propos de la sélection par `id`

La sélection par `id` est aujourd'hui considérée comme une **mauvaise pratique** mais s'utilise en JavaScript.

Organisation d’un fichier CSS

Sur les projets que l’on abordera ensemble, un seul fichier CSS sera suffisant pour tout un projet (et c’est le cas aussi dans le monde réel).

Le fichier est organisé selon deux grands niveaux

1. Le début du fichier contient des styles génériques qui s’appliquent à une large sélection d’éléments;
2. Ensuite les éléments plus spécifiques, les composants et sous-composants qui font votre interface.

🤑 Héritage

Principe par lequel des propriétés sont naturellement ***transmises*** d'un élément parent vers un élément enfant.

- Des propriétés se propagent, comme la couleur du texte, l'interlignage, l'interlettrage et d'autres ne seront jamais héritées par défaut (marges, fonds);
- Les liens et les formulaires n'héritent pas d'autant de propriétés que les autres éléments (police pour les formulaire et couleur pour les liens.)

CSS

```css
/* Pour forcer l'héritage */
a {color: inherit}
input {font-family: inherit}
```

Exemple d’héritage

HTML

```
<p>Un simple paragraphe</p>
<section>
  <p>Un autre paragraphe, enfant de l’élément section</p>
</section>
```

CSS

```
section { color: orange }
```

Le second paragraphe aura une couleur orange dont il *hérite* de son parent l’élément `section`.

🌊 Cascade, le *C* de CSS

L'aspect d'un élément est déterminé par plusieurs règles qui le concernent :

- Spécificité (force) des sélecteurs ou héritages;
- L'ordre d'apparence des règles qui le concernent;
- L'importance forcée (`!important`, **mauvaise pratique**);

Cascadons

Mettez en place le code de l’exemple qui suit dans un nouveau projet ou récupérez le code proposé sur mon espace

https://developer.mozilla.org/fr/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance

La sélection hiérarchique

CSS

```css
/* Sélecteur descendant simples */

/* p présents dans des sections */
section p {color: orange}

/* .small présents dans des sections */
section .small {color: blue}

/* Les h2 enfants directs de body */
body > h2 { color: red }
```

D’autres sélecteurs

CSS

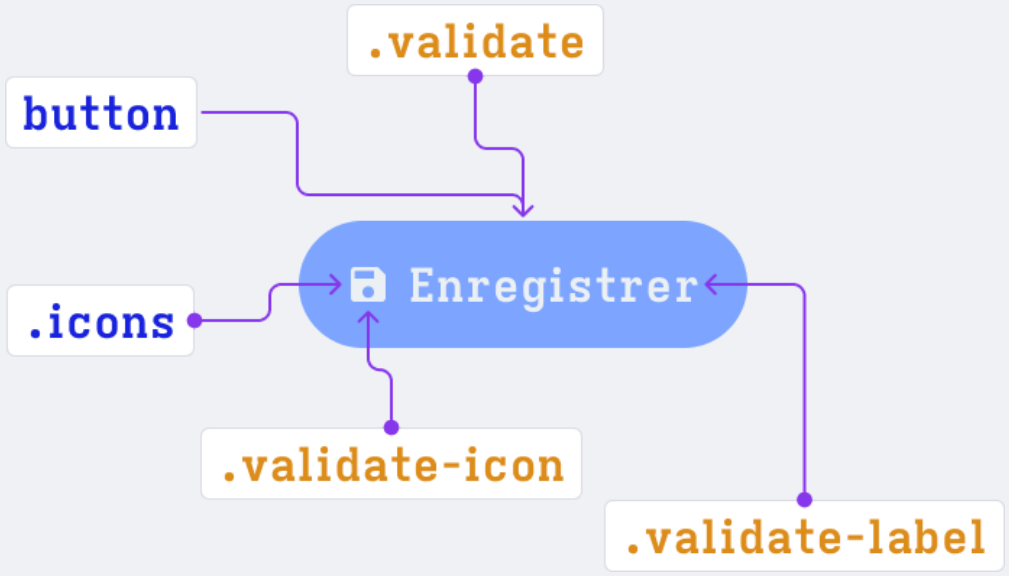
```css
/* Sélecteur universel (*), tout les éléments seront soulignés */
* {text-decoration: underline}

/* Regroupement, p et .small auront les mêmes règles */
p, .small {font-family: serif}

/* Sélecteur de class combiné, les éléments de type p avec une class small */
p.small {color: green}

/* Sélecteur d’état survolé */
.item:hover {color: blue}

/* Sélecteur de sélection :) */
body::selection {background-color: yellow}
```

Quand utiliser quel sélecteur ?

Pour résumer, la sélection par élément et par class seront les modes que vous allez utiliser.

- Pas de sélection par `id` en CSS, sauf si vous avez plus de 35 ans;
- La sélection par élément (balise ou *tag*) est utilisée pour *initialiser* des styles génériques sur des familles d'éléments. Régler la typographie sur tout un document, le fond de la page, apparence des liens;
- La sélection par `class` sera votre alliée pour la vie;

Exemple simple



CSS

```css
button {
  border: 0;
  border-radius: 3rem;
}
.icons {
  width: 1rem;
  height: 1rem;
}
.validate {
  color: white;
  background-color: #7EA9FF;
}
.validate-icon {}
.validate-label {}
```

Quelques références sur la sélection en CSS

- Un cheatsheet interactif des sélecteurs CSS (en);
- Un autre cheatsheet (en);
- Comment fonctionne la sélection CSS (en);
- Une liste complète des sélecteurs existants (en);
- CSS Diner, jeu de sélection en CSS (langues au choix)

Propriétés, valeurs, box model

Propriétés et valeurs

- Les couleurs, notation hexadécimales, `rgb, rgba, hsl, hsla…` ;
- Unités de mesure : `px, em, rem, %, vh, vw` ;
- Typographie : `font-family, font-size, font-weight, line-height` ;
- Les bordures, marges et espacements (`padding, margin, border`);
- Box model pour les * noobs*;

Couleurs

Le web utilise un espace RVB (sRGB).

Les couleurs peuvent se noter de différentes manières, avec des couleurs nommées, en `#rrggbb`, `rgb`, `hsl`, `oklab`, `oklch`, `lab`, `oklab` et leur dérivées avec une valeur *alpha* (la transparence) `#rrggbbaa`, `rgba`, `hsla`.

- Un petit exercice ?
- Un guide complet sur les couleurs en CSS

Unités de mesure

Unités relatives et absolues, nous disposons d’énormément d’unités en CSS

- Les unités *absolues* conserve leur dimension sans prendre en compte le contexte;
- Les unités *relatives* sont dépendantes d’un autre élément, un élément parent, la dimension de la fenêtre.

Unités absolues

`q, mm, cm, in, pc, pt et px`, des dimensions ancrées dans le monde réel, immuable. Un centimètre sera toujours égal à lui-même.

Les unités absolues ne sont presque plus utilisées dans un contexte ou la conception web *responsive*, oubliez les pixels.

Une liste de ces unités

Unités relatives

`em, ex, ch, rem, lh, vw, vh, vmin, vmax, %`, les dimensions relatives sont dépendantes d'un contexte, la taille d'un élément parent, de la fenêtre du navigateur.

Même si ces valeurs semblent plus complexes, elles se révèlent être une réponse idéale quand on ne maîtrise pas la taille d'un écran, d'une fenêtre ou la taille d'une police

Exercice sur les dimensions

Téléchargez ce projet et suivez ce qui est demandé dans le fichier html.

Typographie

Le module de typographie de CSS est un outil puissant permettant de contrôler tous les aspects d'un texte.

`font-family`, `font-size`, `font-weight`, `line-height`, `letter-spacing`, `text-decoration`, `font-style`, `font-variant` `text-transform` pour ne citer que les principales fonctionnalités liées à la gestion de la typographie.

Essayons-en quelques-une. Téléchargez ce projet et suivez ce qui est demandé dans le document HTML

Bordures, marges et espacement

- `margin`, la marge autour d'un élément;
- `padding` représente la marge interne à un élément;
- `border` , la bordure physique d'un élément;
- `outline` une bordure alternative, non physique.

Un petit exercice pour tester ces propriétés.

box model, le modèle de boîte CSS

En CSS tout est une boîte. Mais elle a un fonctionnement qu'il faut connaître.

Un fois rendue à l'écran quelle est la largeur de cette boîte en pixels ?

CSS

```css
.la-boite {
  width: 20rem;
  height: 20rem;
  margin: 1rem;
  padding: 1rem;
  border: 2rem dashed solid;
}
```

Réponse

416 pixels = 20rem + (2 × 1rem) + (2 × 2rem)

La largeur + (padding gauche + droite) + (border gauche + droite)

Intuitif ? Pour mieux gérer cet aspect du langage il existe un propriété `box-sizing`.

CSS

```css
*,*:after,*:before {
  box-sizing: border-box
}
```

Références

- Un outil de conversion de couleurs
- CSS Diner, un jeu sur la sélection CSS;
- CSS Battle, si tu aimes jouer pas le meilleur endroit pour apprendre les bonnes pratiques;
- Un document complet à propos des sélecteurs CSS (en);
- Un speedrun ?;
- Des cours complet, (fr-ca)
- À lire et à relire (en);
- À propos de pixels