Part 1

Bases en css

les fondamentaux du css : cascade et héritage : MDN

Les sigles « CSS » sont l'abréviation de « Cascading StyleSheets » ou « feuilles de styles en cascade » en français.

Le CSS vient résoudre un problème bien différent du HTML : en effet, le HTML sert à définir les différents éléments d'une page, à leur donner du sens. Le CSS, lui, va servir à mettre en forme les différents contenus définis par le HTML en leur appliquant des styles.

Si votre navigateur affiche par défaut les contenus textuels que vous avez déclaré comme des titres en HTML en grand et en gras et à l'inverse les paragraphes en plus petit c'est justement parce que chaque navigateur possède sa propre feuille de styles. En fournissant nos styles CSS pour les différents contenus de notre page, nous allons donc pouvoir définir l'apparence de ceux-ci puisque les styles que nous allons fournir vont être considérés comme prioritaires par rapport à ceux du navigateur. C'est le principe du style en cascade (CSS)

1.1 Une déclaration en CSS

Une règle typique est composée de trois parties :

- un **sélecteur**,
- une **propriété**,
- une **valeur**

selecteur { propriete : valeur ; }

La **valeur** de la propriété se met après le séparateur `:`.

Ces règles deviennent des *déclarations* lorsque celles-ci comprennent plusieurs couples propriété-valeur :Les règles sont alors séparées par un point-virgule `;`, dans la même accolade :

```
selecteur {
  propriete 1: valeur;
  propriete 2: valeur;
}
```

Le selecteur fait référence à un ou plusieurs éléments du document.

1.2 indexer les parties du document

Classes et identifiants L'un des attributs des éléments HTML peut servir à indexer cet élément :

- de manière unique : l'attribut `id="valeur"` : une seule valeur d'attribut `id` peut être utilisée dans tout le document HTML. L'attribut id sert, comme son nom l'indique, à identifier un élément HTML en particulier dans une page pour ensuite pouvoir le cibler précisément (pour lui appliquer des styles CSS, ou bien créer un lien vers cet élément).

- avec un attribut de classe : `class="valeur"` : ici, plusieurs éléments peuvent avoir la même *classe*. On pourra alors séléctionner (en css et en javascript) tous les éléments possédant cette classe, avec une seule instruction.

Remarque : Plusieurs classes peuvent être associées à un élément HTML en les séparant par un espace : `<h1 class="titre lexique">`

1.3 Selecteurs

Pour pouvoir appliquer un style à un contenu, il va déjà falloir le cibler, c'est-à-dire trouver un moyen d'indiquer qu'on souhaite appliquer tel style à un contenu en particulier.

Pour cela, nous allons utiliser des sélecteurs.

selecteur	effet
h1	Le nom d'une balise. Toutes les balises de ce nom contenues dans le document seront concernées.
h1,h2,h3	Plusieurs sélecteurs séparés par des virgules.Toutes les balises énumérées sont concernées.
#logo	Le caractère # indique un identifiant. La balise avec cet ID est concernée.Exemple img id="logo" src="..."/.
.bidule	Le point introduit une classe. Toutes les balises comportant un attribut class de ce nom sont concernées.Exemple : p class="bidule"
tab img	Plusieurs sélecteurs séparés par un espace : inclusion. Les balises img incluses dans un tableau sont concernées. Il peut exister des niveaux intermédiaires : ici il y aura probablement au moins des balises tr et td.

Part 2

Règles principales

2.1 Principe

Le CSS permet d'ajouter des propriétés aux éléments (html) de la page. Par exemple, pour realiser la ligne suivante, avec une partie du texte en gras, et une partie en caractères normaux, non gras :

Acteur : Daniel Crog

On peut, soit utiliser une balise inline `<strong>` placée avant le texte (propriété `font-weight: bold`), soit une balise inline générique, et placer une instruction CSS. Ces 2 choix vont donner le même résultat :

```
1 <!-- choix n°1-->
```

```
<strong>Acteur:</strong> Daniel Crog
<!-- choix n°2-->
<span style="font-weight: bold">Acteur:</span> Daniel Crog
```

2.2 Règles appliquées aux textes

Tutoriel complet : developer.mozilla.org

Quelques propriétés CSS relatives aux textes

paramètre	valeurs possibles	effet
color	blue, yellow, red, ...	colorie le texte selon le nom *anglais* de la couleur
color	rgb(0..255, 0..255, 0..255)	selon le code RGB sur un octet chacun
color	#0..F0..F0..F	selon le code RGB exprimé en hexadecimal, sans separateur
background-color	idem color	couleur de fond
font-family	arial, courier new...	choix de la police de caractères
font	italic 12 px sans-serif	informations complémentaires pour la police
text-align	center, left, right, justify	alignement du texte
font-weight	bold	mettre en gras

2.3 Règles appliquées aux boites

Tutoriel complet : developer.mozilla.org

En CSS, tout élément est inclus dans une boîte {{< img src="../images/boite.png" >}} | paramètre | valeurs possibles | effet | |— |— |— | | border | solid, dashed, none | bordure en trait plein, pointillé, sans bordure | | margin | 1px | marge exterieur, dimension en pixels | | padding | 1px | marge intérieure |

2.4 Rappel sur la propriété display

- Les balises s'affichant en "block" (bloc) : elles prennent toute la largeur disponible et s'affichent avec un saut de ligne avant et après. On peut les dimensionner avec les propriétés `width` et `height`. Elles n'accèptent pas `vertical-align`
- Les balises s'affichant en "inline" (dans la ligne) : elles prennent uniquement la largeur utile pour leur contenu, sans ajouter de saut de ligne. Elles acceptent `vertical-align`, mais pas `width` et `height`.

Part 3

Approfondir

3.1 héritage

Lien alsacreation

L'héritage des CSS est fondé sur le modèle Parent-Enfant(s) : chaque élément Enfant reçoit en héritage tous les styles de son élément Parent. Par exemple, la balise `<html>` est parent de `<body>`, et `<table>` est parent de `<tr>` qui lui-même est parent de `<td>`. Cet héritage est très pratique et permet d'éviter beaucoup de répétitions inutiles.

Précision : l'élément enfant héritera de toutes les propriétés de l'élément parent uniquement si ces propriétés s'héritent, car l'héritage ne fonctionne pas non plus sur toutes les propriétés css (margin, padding et autres propriétés de bloc).

Pour appliquer une propriété de style "par défaut" à un document, un auteur peut l'appliquer à la racine de l'arbre du document.

Par exemple ici, tous les descendants de l'élément BODY auront la valeur de couleur 'black', la propriété 'color' étant héritée :

```
BODY { color: black; }
```

Les valeurs de pourcentage spécifiées ne sont pas héritées, celles calculées le sont.

Par exemple, cette feuille de style :

```
BODY { font-size: 10pt }
H1 { font-size: 120% }
```

et cet extrait d'un document :

```
<BODY>
  <H1>Un <EM>grand</EM> titre</H1>
</BODY>
```

Ici, la propriété 'font-size' de l'élément H1 aura une valeur calculée de '12pt' (120% de la valeur de son parent). Et, comme la valeur de la propriété 'font-size' est héritée, la valeur calculée pour l'élément EM sera aussi '12pt'.

3.2 Selection, compléments

selecteur	effet
td > img	Plusieurs sélecteurs séparés par un caractère supérieur : inclusion directe.Les balises img directement incluses dans une cellule de tableau sont concernées. Il ne doit pas exister de niveaux intermédiaires.

selecteur	effet
td + img	Plusieurs sélecteurs séparés par un signe + : vient après.Désigne les balises img venant après une balise p.
p ~ img	Plusieurs sélecteurs séparés par un signe ~ : une balise qui suit une autre (directement ou pas)
:hover	Le caractère : (deux points) introduit une pseudo-classe. La pseudo-classe désigne un élément lorsqu'il se trouve dans une certaine situation : ici lorsqu'il est survolé par la souris.Voir la liste complète des pseudo-class.
img[width]	Désigne les balises img comportant un attribut width quelle que soit la valeur de ce dernier. On teste donc seulement la présence de l'attribut.
p[lang = "fr"]	Ici on limite la portée aux balises p qui comportent l'attribut lang avec la valeur "fr".p lang="fr"

Exemple : mise en couleur de fond gris pour une cellule sur 2 du tableau

```
tbody > tr:nth-child(odd) > td {
  background-color: lightGrey;
}
```