

Syntaxe et sélecteurs

Styles en ligne

Les styles CSS peuvent être directement ajoutés aux éléments HTML en utilisant l' `style` attribut dans la balise d'ouverture de l'élément. Chaque déclaration de style se termine par un point-virgule. Les styles ajoutés de cette manière sont appelés *styles en ligne*.

```
<h2 style="text-align: center;">Centered text</h2>
```

```
<p style="color: blue; font-size: 18px;">Blue, 18-point text</p>
```

Objectif du CSS

CSS, ou feuilles de style en cascade, est un langage utilisé en combinaison avec HTML qui personnalise l'apparence des éléments HTML. CSS peut définir des styles et modifier la mise en page et la conception d'une feuille.

Écrire CSS dans un fichier HTML

Le code CSS peut être écrit dans un fichier HTML en enfermant le code dans des `<style>` balises. Le code entouré de `<style>` balises sera interprété comme une syntaxe CSS.

```
<head>
  <style>
    h1 {
      color: blue;
    }
  </style>
</head>
```

Écrire CSS dans des fichiers séparés

Le code CSS peut être écrit dans ses propres fichiers pour le séparer du code HTML. L'extension des fichiers CSS est `.css`. Ceux-ci peuvent être liés à un fichier HTML à l'aide d'une `<link>` balise dans la `<head>` section.

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

Séparer le code HTML du code CSS

Il est courant de séparer le code de contenu dans les fichiers HTML du code de style dans les fichiers CSS. Cela peut aider à rendre le code plus facile à maintenir, en gardant la syntaxe de chaque fichier séparée, et toute modification du contenu ou du style peut être apportée dans leurs fichiers respectifs.

<link> Élément de lien

L' `<link>` élément est utilisé pour lier des documents HTML à des ressources externes telles que des fichiers CSS. Il utilise couramment :

- `href` attribut pour spécifier l'URL de la ressource externe
- `rel` attribut pour spécifier la relation du document lié au document courant
- `type` attribut pour définir le type de contenu lié

Sélecteurs de classe et d'ID

Les classes CSS peuvent être réutilisables et appliquées à de nombreux éléments. Les sélecteurs de classe sont indiqués par un point `.` suivi du nom de la classe. Les sélecteurs d'ID CSS doivent être uniques et utilisés pour styliser un seul élément. Les sélecteurs d'ID sont indiqués par un signe dièse `#` suivi du nom d'ID.

```
<!-- How to link an external
stylesheet with href, rel, and type
attributes -->
```

```
<link
href="./path/to/stylesheet/style.css"
rel="stylesheet" type="text/css">
```

```
/* Selects all elements with
class="column" */
.column {
```

```
/* Selects element with id="first-
item" */
#first-item {
```

Groupe de sélecteurs CSS

Faites correspondre plusieurs sélecteurs à la même règle CSS, en utilisant une liste séparée par des virgules. Dans cet exemple, le texte pour les deux `h1` et `h2` est défini sur rouge.

```
h1, h2 {
  color: red;
}
```

Chaînage de sélecteur

Les *sélecteurs* CSS définissent l'ensemble d'éléments auxquels s'applique un ensemble de règles CSS. Par exemple, pour sélectionner tous les `<p>` éléments, le `p` sélecteur peut être utilisé pour créer des règles de style.

Les sélecteurs CSS peuvent être chaînés afin que les ensembles de règles s'appliquent uniquement aux éléments qui correspondent à tous les critères. Par exemple, pour sélectionner des `<h3>` éléments qui ont également la `section-heading` classe, le sélecteur `h3.section-heading` peut être utilisé.

```
/* Select h3 elements with the
section-heading class */
h3.section-heading {
  color: blue;
}

/* Select elements with the section-
heading and button class */
.section-heading.button {
  cursor: pointer;
}
```

Sélecteurs de type CSS

Les *sélecteurs de type* CSS sont utilisés pour faire correspondre tous les éléments d'un type ou d'un nom de balise donné. Contrairement à la syntaxe HTML, nous n'incluons pas les crochets angulaires lors de l'utilisation de sélecteurs de type pour les noms de balises. Lors de l'utilisation de sélecteurs de type, les éléments sont mis en correspondance quel que soit leur niveau d'imbrication dans le code HTML.

```
/* Selects all <p> tags */
p {
}
```

sélecteurs de classe CSS

Le sélecteur de classe CSS fait correspondre les éléments en fonction du contenu de leur `class` attribut. Pour sélectionner des éléments ayant `calendar-cell` comme valeur de l' `class` attribut, `.calendar-cell` doit être préfixé.

```
.calendar-cell {
  color: #fff;
}
```

Attributs HTML avec plusieurs valeurs

Certains attributs HTML peuvent avoir plusieurs valeurs d'attribut. Plusieurs valeurs d'attribut sont séparées par un espace entre chaque attribut.

```
<div class="value1 value2 value3">
</div>
```

Spécificité du sélecteur

La spécificité est un système de classement utilisé lorsqu'il existe plusieurs valeurs de propriété en conflit qui pointent vers le même élément. Lors de la détermination de la règle à appliquer, le sélecteur avec la spécificité la plus élevée l'emporte. Le type de sélecteur le plus spécifique est le sélecteur d'ID, suivi des sélecteurs de classe, suivis des sélecteurs de type. Dans cet exemple, seul `color: blue` sera implémenté car il a un sélecteur d'ID alors qu'il `color: red` a un sélecteur de type.

```
h1#header {
  color: blue;
} /* implemented */

h1 {
  color: red;
} /* Not implemented */
```

Sélecteurs d'ID CSS

Le sélecteur d'ID CSS fait correspondre les éléments en fonction du contenu de leur `id` attribut. Les valeurs de l' `id` attribut doivent être uniques dans l'ensemble du DOM. Pour sélectionner l'élément ayant `job-title` comme valeur de l' `id` attribut, un `#` doit être préfixé.

```
#job-title {  
  font-weight: bold;  
}
```

Sélecteur de descendants CSS

Le combinateur de sélecteur *descendant* CSS est utilisé pour faire correspondre les éléments qui sont les descendants d'un autre sélecteur correspondant. Ils sont désignés par un seul espace entre chaque sélecteur et le sélecteur descendant. Tous les éléments correspondants sont sélectionnés quel que soit le niveau d'imbrication dans le code HTML.

```
div p { }
```

```
section ol li { }
```