Les balises génériques

Elles permettent de **regrouper des éléments** afin de leur appliquer du **CSS commun**, sans ajouter de valeur sémantique.

Elles sont de deux types :

- Les balises de type **block** : `<div>`, qui prennent toute la largeur disponible dans la page et provoquent un retour à la ligne,

- Les balises de type **inline** : `<span>`, qui ne prennent que la largeur nécessaire à leur contenu, et ne provoquent pas de retour à la ligne.

Attention : on peut mettre une balise de type inline *à l'intérieur* d'une balise de type block, mais pas l'inverse.

Styler en CSS

Notions de base

La séparation des préoccupations

La séparation du fond - en HTML - et de la présentation - en CSS - facilite l'**adaptation de l'affichage des documents web** en fonction de plusieurs paramètres, notamment :

- Les **préférences utilisateur** (ex. thème foncé, contrastes, tailles de polices…),
- Le **type de terminal client** (ex. ordinateur, mobile…),
- La **résolution d'écran**,
- La compatibilité entre les navigateurs…

Notions de base

La séparation des préoccupations

Il est possible d'ajouter du CSS à un document HTML de plusieurs façons :

- via un **attribut** `style` sur les balises HTML dont on souhaite modifier la présentation,
- contenues dans un **élément** `<style>...</style>`, en général lui-même contenu dans l'élément `<head>` du document,
- dans un **fichier externe** portant l'**extension de fichier .css**, importé dans le document HTML via l'élément `<link>` :

```
<link ref="stylesheet" href="styles.css" />
```

Notions de base

La séparation des préoccupations

La première méthode est verbeuse peu pratique à maintenir, et la seconde ne permet d'agir que sur un seul document à la fois - alors qu'un site web comporte en général plusieurs pages.

C'est pourquoi en dehors de certains cas d'usage, on préférera généralement la troisième méthode, qui permet d'appliquer les mêmes feuilles de styles à plusieurs documents, ce qui garantit une **cohérence de design** à l'échelle du site web.

Notions de base

Structure d’une règle CSS

Un **sélecteur** permet le ciblage d'un ou plusieurs éléments (ou parties d'élément) dans le document.

Une **propriété** est un paramètre de style appliqué aux éléments sélectionnés, qui peut prendre une valeur.

La combinaison d'une propriété et d'une valeur est appelée **déclaration.**

Notions de base

Structure d'une règle CSS

Une **règle** consiste en une ou plusieurs déclarations appliqués à un ou plusieurs sélecteurs suivant la syntaxe suivante :

```
selecteur
(, selecteur) {
  propriété: valeur;
  (propriété: valeur;)
}
```

Notions de base

La cascade

Il s'agit de l'**algorithme** au coeur de CSS, qui calcule quelles déclarations de styles s'appliquent réellement aux éléments HTML.

Différents critères modifient le "**poids**" d'une déclaration :

- L'**importance** : l'ajout de `!important` à la fin d'une déclaration en augmente le poids ;
- L'**origine** :
    - le **site web** (les seuls sur lesquels vous pouvez intervenir),
    - la **feuille de styles de l'utilisateur**,
    - les **styles par défaut du navigateur** ;
- La **spécificité du sélecteur** (nous y reviendrons) ;
- La **position** de la règle : à score égal pour les critères précédents, c'est la **dernière déclaration définie** qui s'applique.

Notions de base

L’héritage

Certaines propriétés dont la valeur est déclarée pour un élément donné peuvent être **héritées** par les éléments **descendants** (ou enfants) de celui-ci, tandis que d'autres ne le sont pas.

Par exemple, les propriétés de couleurs ou de polices seront hérités - afin de réduire la quantité de CSS nécessaire à l'affichage - tandis que celles de largeur ou de hauteur ne le seront pas - car cela engendrerait au contraire une complexification inutile.

Notions de base

Flux de rendu et modèle de boîtes

Le modèle évoqué pour `div` et `span` concerne en réalité **tous les éléments HTML**.

Les élément de type **block** :

- occupent toute la largeur disponible et provoquent un retour à la ligne,
- les propriétés de largeur et de hauteur sont appliquées,
- les propriétés de marges et de bordures repoussent les éléments alentours,
- ex. `<p>, <article>`…

Notions de base

Flux de rendu et modèle de boîtes

Les éléments de type **inline** :

- n'occupent que la largeur nécessaire à l'affichage de leur contenu et ne provoquent pas de retour à la ligne,
- les propriétés de largeur et de hauteur ne sont pas appliquées,
- les propriétés de marges et de bordures sont appliquées mais ne repoussent pas les éléments alentours;
- ex. `<a>, <input>`…

Notions de base

Flux de rendu et modèle de boîtes

Il est cependant possible de changer le type d'un élément via la propriété **`display`**, qui peut donc prendre comme valeur **`block`** ou `inline`, mais également **`inline-block`**, afin de créer un **type hybride**, qui possède des caractéristiques des deux types :

- comme **`inline`**, il n'occupe que la largeur nécessaire à son contenu et ne provoque pas de retour à la ligne,

- comme **`block`**, les propriétés de largeur et de hauteur seront appliquées, et les propriétés de marges et de bordures auront un effet sur les éléments voisins.