**CSS**

CSS pour Cascading Style Sheets en anglais. On peut avoir plusieurs façons pour incorporer le CSS directement dans le HTML :

* En utilisant la balise <style> directement dans notre HTML.
* Soit en utilisant un attribut style dans une balise tels que : <h1 style= « color » >.
* Où, celui qu’on fait majoritairement, on crée un fichier de style.css et qu’on incorpore dans le HEAD de HTML avec la balise <link>.

Quand on commence à coder du CSS, celui-ci va lire notre code dans l’ordre qu’on a écrit. Par exemple, si on veut mettre notre paragraphe en rouge :

P {

Color : red ;

}

SI on avait mis avant ce code :

Body {

Color : green

}

P {

Color : red ;

}

Le CSS écrasera le green sur le paragraphe pour mettre le rouge étant en dernier.

**PRIORITER :**

Après il y a aussi un ordre de priorité dans la lecture du CSS. En effet, ID sera en premier dans l’ordre de lecture, puis la class et enfin la balise. Par exemple :

Body {

Color : green

}

#NomID {

Color : red ;

}

.NomCLASS {

Color : blue ;

}

Dans cette ligne de code, la couleur qui s’affichera à la fin sera le rouge car l’ID sera prioritaire.

**PASSAGE PAR UN PARENT**

Quand on veut que l’enfant d’un parent soit d’une couleur on peut écrire notre ligne de cette façon : UL LI {…}. A ce moment-là tous les enfants directs et indirects seront modifiés. Mais si on veut que ce soit que les enfants directs qu’ils soient modifié et non les enfants indirects on va écrire : UL > LI {…}.

Si on veut juste modifier, un enfant dans un parent nous allons ajouter « + » afin que la modification soit sur l’élément souhaité. Par exemple :

body > main > section > article + article + article {

background-color : blue ;

}

A ce moment-là, c’est le dernier article qui sera en bleu et les deux autres resteront en blanc.

Taille absolu : PX

Responsive : % (plutôt pour les blocs)

EM (emphase): dépend de la taille du parent => pour les textes => 1 valeur de base

REM (root): dépend de la taille de l’élément racine html

H2 ~UL => l’élément qui précédent l’autre élément

Second class tels que ul :hover {…}