

CSS : Introduction

CSS : Cascading Style Sheets



- Le CSS est un langage interprété par le navigateur qui sert à mettre en forme une page web.
- Le navigateur va « lire » les instructions et appliquer notre « style », une esthétique, à notre page.
- Avec le CSS, nous pouvons agir par exemple sur :
 - Les couleurs
 - Les tailles
 - Le placement des éléments sur la page
 - Les polices de caractères (font)
 - Etc...

CSS : Cascading Style Sheets

Que peut-on faire concrètement avec le CSS ?

CSS Zen Garden peut donner des idées de ce qu'il est possible de faire uniquement en CSS.

<https://www.csszengarden.com/>



CSS : Bases

Le code CSS peut être placé à **3 endroits différents** :

- **CSS « en ligne »** : Le code CSS est placé dans n'importe quelle balise html, au moyen de l'attribut « style ». Le principal problème que cela occasionne, est que ça engendre une énorme duplication de code et rend la maintenance plus compliquée.

```
<p style="color: red;">  
Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Mauris commodo lacus placerat, ultrices quam vitae, luctus odio.  
</p>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Mauris commodo lacus placerat, ultrices quam vitae, luctus odio.

CSS : Bases

Le code CSS peut être placé à **3 endroits différents** :

- **CSS « interne »** : Le code CSS est placé dans une balise html « style », enfant de la balise « head ». Le code n'est maintenant plus dupliqué sur la page html, en revanche si nous en avons plusieurs, le même problème de maintenabilité et duplication de code se pose.

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

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Mauris commodo lacus placerat, ultrices quam vitae, luctus odio.

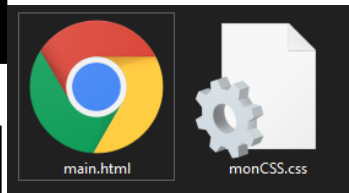
CSS : Bases

Le code CSS peut être placé à **3 endroits différents** :

- **CSS « externe »** : Le code CSS est placé dans des fichiers « .css » .
Le code peut être mutualisé pour la totalité de l'application.
Le style est aussi plus uniforme.

```
<head>  
    <link rel="stylesheet" href="monCSS.css" >  
</head>
```

```
p{  
    color: magenta;  
}
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Mauris commodo lacus placerat, ultrices quam vitae, luctus odio.

CSS : Bases

Que choisir ?

Dans un fichier .css ! Car cela a de nombreux avantages :

- + Permet une maintenance du code plus simple
- + Permet d'utiliser le même fichier sur plusieurs pages HTML

Comment ça marche ?

Il suffit de lier notre page CSS avec notre page HTML grâce à la balise <link> dans le <head> :

```
<head>  
    <link rel="stylesheet" href="monCSS.css" >  
</head>
```

CSS : Bases

Regardons notre code en détail :

```
p{  
    color: magenta;  
}
```

Nous avons :

- **En rouge** : Le sélecteur => Permet de choisir sur quels éléments s'applique le style
- **En jaune** : Les accolades => Délimitent les propriétés que l'on va assigner à un élément
- **En bleu** : La propriété => Les effets graphiques qui vont être appliqués
- **En orangé** : La valeur => La valeur (un nombre, une couleur, ...)
- **En rose** : Le séparateur => Le point-virgule « ; » sert à séparer les propriétés entre elles

Ici, nous appliquons donc la couleur « magenta » au texte de nos balises <p>.

CSS : Bases

Commentaires

Comme pour le HTML, nous pouvons mettre des commentaires dans notre code qui ne seront pas interprétés par le navigateur, et qui n'apparaîtront pas sur la page.

Attention, tout ce que vous écrivez en commentaire pourra aussi être lu par n'importe qui !

Pour écrire un commentaire, il faut le mettre entre « /* » ouvrante et « */ » fermante comme ceci :

```
p{  
    /*J'applique la couleur magenta*/  
    color: magenta;  
}
```

CSS : Cascade et héritage

Il y a plusieurs choses à savoir pour bien maîtriser l'art du CSS.

- **Héritage :**

Les enfants d'un élément HTML héritent des propriétés CSS de leur parent.

```
div{  
    color: magenta;  
}
```

```
<div>  
    <p>  
        Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
    </p>  
</div>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

CSS : Cascade et héritage

Il y a plusieurs choses à savoir pour bien maîtriser l'art du CSS.

- **Héritage :**

Les enfants d'un élément HTML héritent des propriétés CSS de leur parent.

```
div{  
    color: magenta;  
}
```

```
<div>  
    <p>  
        Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
    </p>  
</div>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

CSS : Cascade et héritage

Il y a plusieurs choses à savoir pour bien maîtriser l'art du CSS.

- **Cascade :**

Les feuilles CSS sont empilées en « cascade ».

Les feuilles de styles sont interprétées dans l'ordre de leur déclaration dans le code HTML.

Si un même élément est ciblé, les éventuels conflits dans les propriétés seront en faveur de la feuille déclarée en dernier.

```
<head>
    <link rel="stylesheet" href="monCSS.css">
    <link rel="stylesheet" href="monCSS2.css">
</head>
/* Les propriétés de monCSS2 écraseront celle de monCSS si elles sont
déclarées */
```

CSS : Cascade et héritage

Il y a plusieurs choses à savoir pour bien maîtriser l'art du CSS.

- **Cascade :**

C'est aussi le cas au sein d'un fichier. Il est interprété « de haut en bas ».

```
p{  
  color: blue;  
}  
  
p {  
  color: magenta;  
}  
/* La couleur du paragraphe sera donc magenta */
```

CSS : Cascade et héritage

Il y a plusieurs choses à savoir pour bien maîtriser l'art du CSS.

- **Spécificité :**
+ un sélecteur cible un élément avec précision, + sa priorité sera haute
L'instruction la plus précise sera donc prioritaire

```
<p id="monId">  
Lorem ipsum dolor sit amet, consectetur adi  
piscing elit.  
</p>
```

```
#monId{  
    color: red;  
}  
p{  
    color: blue;  
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

CSS : Récap

- On déclare de préférence notre CSS dans un fichier séparé que l'on lie à notre page grâce à la balise `<link>` qui se situe dans le `<head>` de la page.
- Notre fichier CSS contient des règles CSS.
- Chaque règle CSS contient une ou plusieurs déclarations qui sont composées d'une propriété et de sa valeur.
- On utilise le « : » pour séparer la propriété de sa valeur.
- On utilise le « ; » pour séparer les déclarations entre elles.
- On utilise les accolades « { } » pour séparer les règles entre elles.

```
h1{  
    color: blue;  
    font-size: large;  
}  
/*Je suis un commentaire*/  
p {  
    color: magenta;  
    background-color: black;  
}
```