

# Découverte

## Découverte de CSS

Rédigé par

**David ROUMANET**  
Professeur BTS SIO



Changement

Date	Révision

## Sommaire

A Introduction.....	1
A.1 Présentation.....	1
A.2 Intégrer CSS dans un fichier HTML.....	2
A.2.1 Section <style></style>.....	2
A.2.2 Chargement d'un fichier CSS.....	2
A.2.3 Syntaxe CSS.....	2
B Nouvelles balises HTML.....	3
B.1 Création d'un bloc : <div>.....	3
B.1.1 Présentation div.....	3
B.1.2 Travail à faire.....	3
B.2 Création d'un bloc <span>.....	5
B.2.1 Présentation span.....	5
B.2.2 Travail à faire.....	5
C Les mises en formes CSS.....	6
C.1 Travail à faire.....	6
C.2 Conteneurs.....	6
C.2.1 Présentation des conteneurs.....	6
C.2.2 Travail à faire sur les conteneurs.....	7
C.3 Classes et identifiants.....	9
C.3.1 Présentation des propriétés class et id.....	9
C.3.2 Travail à faire.....	9
C.4 Les bordures.....	10
C.4.1 Présentation propriété border.....	10
C.4.2 Travail sur les bordures.....	10
C.4.3 Largeur et largeur maximum.....	12
D Cours à suivre.....	13
D.1 CSS dinner.....	13
D.2 Guess CSS.....	13
D.3 Grid garden.....	13
D.4 Flexbox Froggy.....	13
E Annexes.....	14
E.1 Sources.....	14
E.2 Autres.....	14
F Ce qu'il faut retenir.....	15

### Nomenclature :

- **Assimiler** : cours pur. Explication théorique et détaillée (globalement supérieur à 4 pages).
- **Décoder** : fiche de cours, généralement inférieure à 5 pages.

- **Découvrir** : Travaux dirigés. Faisable sans matériel.
- **Explorer** : Travaux pratiques. Nécessite du matériel ou des logiciels.
- **Mission** : Projet encadré ou partie d'un projet.
- **Voyager** : Projet en autonomie totale. Environnement ouvert : Vous êtes le capitaine !

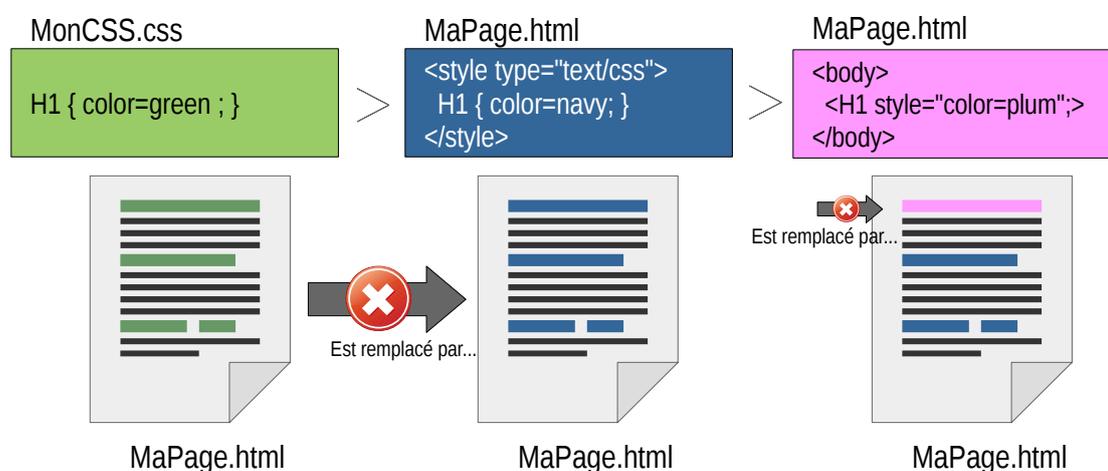
## A Introduction

Nous avons vu que le langage HTML fournit la structure d'un document. Ainsi, le navigateur peut appliquer la mise en forme par défaut.

Cependant, la volonté de personnaliser les couleurs et les propriétés de mise en forme, ont poussé la création d'un langage permettant de facilement changer les éléments.

### A.1 Présentation

Le CSS (Cascaded Style Sheet) peut se traduire par feuille de styles en cascade. L'idée est de proposer des styles, dont le plus proche de l'élément à décorer est prioritaire.



Ainsi, le style d'un élément défini dans un fichier CSS sera moins prioritaire qu'un autre style sur le même élément, défini dans la section `<head>` de la page. Ce dernier style sera modifié par la propriété `style` à l'intérieur de la balise de l'élément.

Le CSS permet de modifier toute la mise en forme de chaque élément : taille (hauteur et largeur), espacement, couleur, police, cadre, mais aussi le comportement.

En effet, CSS est devenu plus dynamique au fil du temps et permet désormais de gérer des animations : transitions, rotations, changement de couleurs ou de propriétés, au survol de la souris ou lors d'événements.

## A.2 Intégrer CSS dans un fichier HTML

Il existe plusieurs moyens d'intégrer du CSS dans une page. Cependant, Quel que soit le moyen, il est préférable que cette intégration se fasse à l'intérieur de la section `<head></head>`.

### A.2.1 Section `<style></style>`

Il existe une balise `<style>` et `</style>` qui permet de délimiter l'endroit où le code sera en CSS. Cela fonctionne de la même manière que la balise `<script>` pour JavaScript.

### A.2.2 Chargement d'un fichier CSS

L'autre méthode est la plus recommandée : le CSS se trouve dans un fichier séparé du HTML. Dans ce cas, il faut importer ce fichier, cela peut se faire de deux manières différentes :

Méthode 1

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

Méthode 2

```
<head>
  <style>
    @import url('css/style.css')
  </style>
</head>
```

Quelle que soit la méthode, on peut charger plusieurs fichiers CSS : c'est très pratique si on utilise un framework existant. La seule chose importante à se rappeler, est qu'en cas de conflit de mise en forme sur un élément, c'est la dernière commande qui s'applique.

### A.2.3 Syntaxe CSS

CSS utilise une syntaxe proche de JavaScript, avec des blocs utilisant `{` et `}`. Chaque ligne doit se terminer avec un `;` à la fin. Chaque propriété peut recevoir une valeur, après le symbole `:` collé à la propriété.

Il suffit de choisir une balise HTML (sans les `<` et `>`) et de créer un bloc après, pour configurer la mise en forme de cette balise. Pour modifier la mise en forme des titres `<h1>` il suffit d'écrire...

```
h1 {
  color: blue;
}
```

Ceci change la couleur des titres h1 en bleu.

## B Nouvelles balises HTML

---

Afin de bien comprendre le fonctionnement de CSS et de HTML, nous allons utiliser deux nouvelles balises.

### B.1 Création d'un bloc : <div>

#### B.1.1 Présentation div

Les balises `<div>` et `</div>` permettent de créer un bloc qui contiendra différents éléments HTML. La seule contrainte de ce bloc, est que par défaut, on ne peut rien mettre à côté (il occupe tout l'espace horizontal). Pour le reste, c'est l'outil de base des développeurs et des créateurs de pages web. CSS permet également de faire que plusieurs blocs puissent se placer les uns à côté des autres.

#### B.1.2 Travail à faire

Dans votre espace workspace, créer un dossier CSS, puis dans VSCode, ouvrir ce répertoire. Créer un nouveau fichier **CSS\_exemple1.html** qui contient le code suivant.

CSS\_exemple1.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <div>
    Lorem ipsum dolor sit amet consectetur adipisicing elit. Nesciunt iste architecto doloremque
    eveniet! Doloribus repellat veritatis consequatur deleniti rerum incidunt dolor deserunt eos numquam
    exercitationem optio aspernatur harum, architecto praesentium.
    Architecto quos repellendus odio commodi velit, dicta corrupti necessitatibus tempora ipsa
    temporibus ex consequuntur quaerat esse itaque voluptate. Corrupti.
  </div>
  <div>Lorem ipsum dolor sit amet consectetur adipisicing elit. Natus deserunt, atque in labore
  mollitia eius dolore? Maxime quis tempore, asperiores accusamus laborum quas pariatur odit itaque quae
  repellendus ab deserunt.
  Sunt molestias laudantium expedita voluptatem quos dicta impedit corrupti omnis, soluta error
  tempore aliquam distinctio dolor hic minima explicabo maiores veritatis, ea voluptas a illo adipisci
  eum mollitia cum. Officiis?
  </div>
  <div>
    Veniam quidem nostrum unde repudiandae ipsa vel rerum recusandae praesentium aspernatur facere
    quam aliquid sequi earum itaque, vitae inventore. Libero, voluptatem accusantium officiis totam culpa
    dicta numquam illo distinctio quis.
  </div>
  <div>
    Lorem ipsum dolor sit amet consectetur adipisicing elit. Inventore fugiat quaerat nam
    explicabo, vel labore adipisci doloribus voluptas eius! Praesentium voluptatibus quisquam natus cum
    maxime amet consequatur vero assumenda vitae.
  </div>
</body>
</html>
```

Afficher la page.

Ensuite, créer la section `<style>` suivante dans l'entête HTML :

```
<style>
  div {
    border: solid grey 1px;
  }
</style>
```

Sauvegarder et afficher la page. Que remarquez-vous ?

Changer la ligne `border: solid grey 1px ;` par

```
border-style: solid;
border-color: grey;
border-width: 1px;
```

Sauvegarder et afficher la page. Voyez-vous un changement ?



Règle : il existe généralement une propriété nommée comme `border-color`, `border-style`, mais il est possible d'appeler plusieurs propriétés liées en utilisant la propriété principale, ici `border` en séparant les valeurs par un espace.

Dernière action, remplacer la section `<style>` par celle-ci :

```
<style>
  div {
    border-style: solid;
    border-color: grey;
    border-width: 1px;
    display: inline-block;
    width: 250px;
  }
</style>
```

Que constatez-vous ?

Ajouter le code suivant pour aligner les blocs en haut et définir à la taille verticale. Cependant, certains textes devrait dépasser du cadre. CSS n'est pas magique...

```
height: 300px;
vertical-align: top;
```

## B.2 Création d'un bloc <span>

### B.2.1 Présentation span

La balise `<span>` est également un bloc invisible par défaut, mais elle s'intègre parfaitement dans un texte. Elle permet de créer une zone de texte qui peut avoir un comportement particulier.

### B.2.2 Travail à faire

Reprenez le fichier `CSS_exemple1.html` et enregistrez-le sous `CSS-exemple2.html`

Dans la section `<style>`, sous le bloc concernant les balises `div`, collez le texte suivant :

```
span {  
  color: goldenrod;  
}
```

Et avant la fin de la section `<body>`, mais après le dernier `</div>`, insérez le paragraphe suivant :

```
<p>“Je crois beaucoup en la <span>chance</span> ; et je constate que plus je <span>travaille</span>,  
plus la chance me sourit”<br>  
  Thomas Jefferson.</p>
```

Que constatez-vous ?

Ajoutez maintenant le style suivant :

```
span:hover {  
  background-color: goldenrod;  
  color: black;  
}
```

Vérifiez que le survol des mots dans les balises `<span>` changent.

Un dernier travail : Passez en mode outil de débogage de la page et cliquez sur l'outil avec un cadre et un curseur à l'intérieur, puis passez votre souris sur votre page web.



Désormais, vous pouvez connaître le formatage de toutes les pages web que votre navigateur affiche.

## C Les mises en formes CSS

Les balises <div> sont très fortement utilisées, même lorsqu'on ne les voit pas.

### C.1 Travail à faire

Dans un navigateur, allez sur <https://www.google.com/> et Affichez les outils de débogage. Choisissez "éditeur de style" pour Mozilla Firefox ou "éléments > Style" pour Chrome et observez le nom et le nombre de balises CSS pour cette simple page.

### C.2 Conteneurs

#### C.2.1 Présentation des conteneurs

L'idée dans un navigateur, est que chaque élément tient dans un conteneur. Ce conteneur propose des propriétés de mise en forme assez régulières :

- Marge autour du conteneur (rien ne peut se rapprocher à moins de cette distance) : propriété `margin`.
- Bordure autour du conteneur (la taille du trait) : propriété `border-width`.
- Espacement entre la bordure et le contenu du conteneur (le texte, par exemple) : propriété `padding`.

Voici à quoi cela ressemble.



Si on ajoute la marge, la bordure et l'espacement, on obtient la taille complète du conteneur.

## C.2.2 Travail à faire sur les conteneurs

Copier le code suivant dans le fichier **CSS\_exemple3.html**.

CSS\_exemple3.html

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>CSS Box Model Demo</title>
  <style>
    @import url("CSS_exemple3.css");
  </style>
</head>
<body>
  <div class="container">
    <div class="box">Bloc 1</div>
    <div class="box">Bloc 2</div>
    <div class="box">Bloc 3</div>
    <div class="box">Bloc 4</div>
    <div class="box center-box">
      Bloc Central
      <span class="label padding-label top">padding: 20px</span>
      <span class="label margin-label right">margin: 10px</span>
      <span class="label border-label bottom">border: 15px</span>
      <span class="label padding-label left">padding: 20px</span>
    </div>
    <div class="box">Bloc 6</div>
    <div class="box">Bloc 7</div>
    <div class="box">Bloc 8</div>
    <div class="box">Bloc 9</div>
  </div>
</body>
</html>
```

Puis créer un fichier **CSS\_exemple3.css** dans le même répertoire et y coller le code suivant.

CSS\_exemple3.css

```
.container {
  display: grid;
  grid-template-columns: repeat(3, 200px);
  grid-template-rows: repeat(3, 200px);
  gap: 10px;
  justify-content: center;
  margin-top: 20px;
}

.box {
  background-color: #e0e0e0;
  display: flex;
  justify-content: center;
  align-items: center;
  font-family: Arial, sans-serif;
}

/* Style spécial pour le bloc central */
.center-box {
  background-color: #fff;
  border: 15px solid #2196F3;
  padding: 20px;
  margin: 10px;
  position: relative;
}

/* Labels pour le bloc central */
```

```

.label {
  position: absolute;
  font-size: 12px;
  color: #333;
}

.padding-label {
  background-color: #FFE082;
  padding: 2px 5px;
}

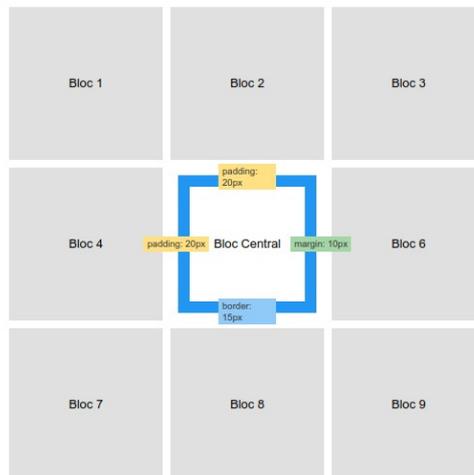
.margin-label {
  background-color: #A5D6A7;
  padding: 2px 5px;
}

.border-label {
  background-color: #90CAF9;
  padding: 2px 5px;
}

/* Positionnement des labels */
.top { top: -30px; left: 50%; transform: translateX(-50%); }
.right { right: -60px; top: 50%; transform: translateY(-50%); }
.bottom { bottom: -30px; left: 50%; transform: translateX(-50%); }
.left { left: -60px; top: 50%; transform: translateY(-50%); }

```

Sauvegarder les deux fichiers et afficher le fichier HTML. Vous devriez obtenir ceci :



Actions à faire :

- Changer la couleur de la bordure en rouge.
- Grossir le trait à 20 pixels (20px).
- Augmenter l'espace intérieur de 50 pixels
- Augmenter la marge extérieure de 20 pixels.

Que se passe-t-il pour le texte et pour les autres blocs ?

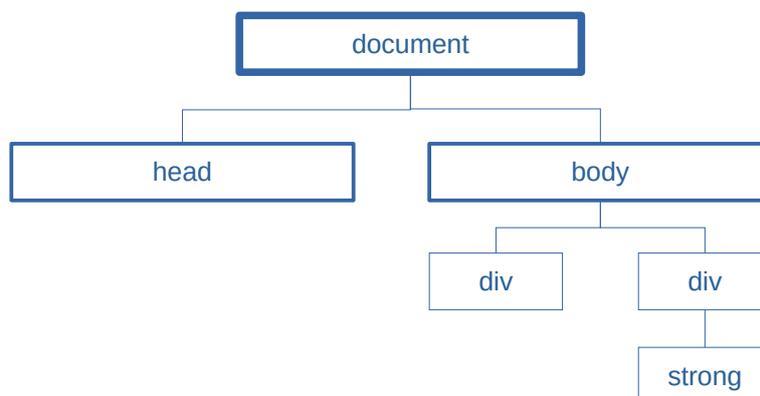
Ce code CSS mérite un peu d'analyse pour le comprendre, car il n'y a aucune mise en forme portant le nom `div`.

La raison est simple, nous utilisons ici la propriété `class`.

## C.3 Classes et identifiants

### C.3.1 Présentation des propriétés class et id

Le principe d'une page HTML est de contenir des éléments affichés sur le `document`. Il se forme alors une structure en forme d'arbre. Chaque nouvelle balise HTML devient un nœud et si on trouve une balise `div` qui contient deux paragraphes `p` (dont l'un ayant du texte en `strong`), on peut le dessiner comme ceci :



Dans ce document, on peut vouloir appliquer un style uniquement à certains éléments et pas à tous. Il existe alors deux propriétés utiles :

- `id` que nous avons déjà vu pour Javascript, et qui fonctionne pour CSS. Un élément ayant l'id "`nomUtilisateur`" pourra ainsi être traité en CSS avec la balise `#nomUtilisateur`.
- `class` est une propriété commune à plusieurs éléments. Tous les éléments ayant une propriété class "`sport`" recevront la mise en forme de CSS qui s'applique sur `.sport`.

### C.3.2 Travail à faire

Ainsi, dans le code `CSS_exemple.html`, identifiez les différentes classes existantes puis regardez dans le fichier CSS le style appliqué.

Changez les traits des boîtes par des tirets (voir [https://www.w3schools.com/css/css\\_border.asp](https://www.w3schools.com/css/css_border.asp) )

Modifiez tous les labels pour écrire en couleur bleue

## C.4 Les bordures

Nous avons déjà utilisé les balises `border` pour rendre visible les conteneurs. Pour rappel, toutes les balises HTML sont des conteneurs, ainsi, il est presque toujours possible d'afficher un cadre autour des balises comme `h1`, `div`, `span`, etc.

### C.4.1 Présentation propriété `border`

Les nombreuses propriétés de `border` sont disponibles dans la documentation de [https://www.w3schools.com/css/css\\_border.asp](https://www.w3schools.com/css/css_border.asp)

Quelques remarques cependant :

L'épaisseur de la bordure est généralement exprimée en pixel (ex : 40px) mais elle peut utiliser d'autres unités (voir exemple). Voici les différentes unités : <https://www.codeur.com/tuto/css/unite-de-mesure-taille-px-em-rem/>

Les bordures peuvent être gérées indépendamment (haut, droite, bas, gauche) ;

### C.4.2 Travail sur les bordures

Copier le code suivant dans le fichier `CSS_exemple4.html`

CSS\_exemple4.html

```
<!DOCTYPE html>
<html>
<head>
<style>
p.one {
  border-style: solid;
  border-width: 0px 1px 2vw 1px;
}
p.two {
  border-style: solid;
  border-radius: 10px;
}
p.three {
  border-style: dotted;
  border-width: 2px;
}
p.four {
  border-style: solid;
  border-width: 1px;
  color: black;
  background-color: cyan;
}
p.five {
  border-style: double;
  border-width: 5px;
  border-radius: 50px 5px 50% 0%;
}
</style>
</head>
<body>

<h2>Les bordures</h2>
<p>Faites bouger la largeur de la fenêtre pour voir le fonctionnement des tailles en pourcentage, em, rem, etc.</p>

<p class="one">Bordures différentes et variables</p>
```

```
<p class="two">Bordures arrondies</p>
<p class="three">Bordures en pointillés</p>
<p class="four">Bordures et remplissage</p>
<p class="five">Bordures différentes chaque angle</p>

</body>
</html>
```

Utilisez la propriété `padding-left: 50px;` pour décaler le texte, dans tous les conteneurs (une seule commande pour tous).

Utilisez la propriété `width: 400px;` pour réduire la largeur des conteneurs pour la classe 'three'.

Utilisez la propriété `margin-right: 50%;` sur le conteneur de classe 'five'.

Faites varier la largeur de la fenêtre de votre navigateur pour comprendre les interactions.

## Les bordures

Faites bouger la largeur de la fenêtre pour voir le fonctionnement des tailles en pourcentage, em, rem, etc.

Bordures différentes et variables

Bordures arrondies

Bordures en pointillés

Bordures et remplissage

Bordures différentes chaque angle

Cherchez maintenant comment centrer le texte dans le conteneur de classe 'two'.

💡 il existe une propriété pour l'alignement de texte, appelée `text-align`.

Changez la couleur de fond de toute la page.

💡 La propriété `background-color` peut s'appliquer à la balise HTML `<body>`.

### C.4.3 Largeur et largeur maximum

Créez un nouveau fichier CSS\_exemple5.html contenant le code suivant :

CSS\_exemple5.html

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  height: 50px;
  width: 400px;
  max-width: 80%;
  border: 1px solid #4CAF50;
}
</style>
</head>
<body>

<h2>Largeur et largeur maximum</h2>

<div>La hauteur de cette section est fixée à 50 pixels. Cependant, si la largeur est fixée à 400
pixels, elle peut être réduite pour respecter la largeur maximum.</div>

</body>
</html>
```

Faite varier la largeur de la fenêtre du navigateur pour comprendre la différence entre `width` et `max-width`.

## D Coursus à suivre

---

Il existe de nombreux sites qui présentent l'ensemble des possibilités en CSS.

### D.1 CSS dinner

Le site <https://flukeout.github.io/> vous entraine à la sélection d'objet en CSS.

La première épreuve est la sélection des assiettes (balises <plate>) avec le mot 'plate'

Rapidement, le jeu vous propose de sélectionner des sous-éléments. Pour cela, on utilise les sélecteurs.

- `element#id` permet de ne sélectionner QUE l'élément qui porte l'ID `id`
- `element.class` permet de sélectionner tous les éléments de type `élément` et portant la classe `class`

Dans le bandeau à droite, vous avez une aide pour chaque niveau.

### D.2 Guess CSS

Le site <https://www.guess-css.app/> est plus orienté sur l'usage des placements statiques et relatifs dans une page.

### D.3 Grid garden

Le site <https://cssgridgarden.com/#fr> propose d'utiliser les grilles en CSS. Cette notion est disponible dans les supports de cours sur <http://david.roumanet.free.fr/BTS-SIO/Bloc1-Dev/130%20CSS/>

L'idée est ici d'apprendre à placer des éléments sur une grille (ici un jardin avec une grille visible) tel que le font les designers web. Les grilles sont en effet une méthode de découpage de page web (généralement 12 colonnes ou 24 colonnes).

### D.4 Flexbox Froggy

Le site <http://flexboxfroggy.com/> apporte des défis sur l'usage des flexbox. C'est le design le plus efficace et très responsive, mais un peu plus difficile à apprendre. Un support est dédié à cette méthode sur <http://david.roumanet.free.fr/BTS-SIO/Bloc1-Dev/130%20CSS/>

## **E Annexes**

---

### **E.1 Sources**

### **E.2 Autres**

---

## F Ce qu'il faut retenir...

---

Insertion > Autres sauts > Saut Manuel...

Choisir le style de page « Résumé »

Attention : résumé s'enchaîne avec Style de page droite ► Mettre une page « Résumé » à la place d'une page Gauche.

Les [liens](#) sont sous et surlignés