Les premières règles CSS

Le HTML dit ce que les choses sont. Le CSS dit à quoi elles ressemblent. Une règle, six morceaux, et une page change complètement d'allure.

4TTR Découverte

Ta page est correcte : le plan est là, les paragraphes sont découpés, le lien fonctionne. Mais elle est noire sur blanc, dans la police par défaut du navigateur. Elle ressemble à un document des années 1990.

Ce n'est pas un défaut du HTML. C'est simplement que le HTML n'a **jamais** eu pour métier de décider de l'apparence. Ce métier-là, c'est celui du CSS.

🎯 Objectifs

À la fin de ce cours, tu seras capable de :

1. Expliquer la différence de rôle entre HTML et CSS.
2. Placer une zone de style au bon endroit dans ta page.
3. Écrire une règle CSS complète et nommer chacune de ses parties.
4. Changer les couleurs, la police, les bordures et les marges intérieures.

Deux langages, deux métiers

Le **CSS** — GB ***Cascading Style Sheets***, feuilles de style en cascade — est un second langage, avec sa propre syntaxe. Il ne remplace pas le HTML : il vient s'y greffer.

Langage	Question à laquelle il répond
HTML	Qu'est-ce que c'est ? Un titre, un paragraphe, une liste…
CSS	À quoi ça ressemble ? Quelle couleur, quelle taille, quel espacement…

Voici la même page, exactement le même HTML, sans puis avec une quinzaine de lignes de CSS :

Rien n'a changé dans le contenu. Pas une balise ajoutée, pas un mot réécrit.

Où écrire le CSS

Le CSS se range dans une balise `<style>`, placée dans le `<head>` — donc dans la partie invisible de la page.

```
<head>
  <meta charset="utf-8">
  <title>Carnet de voyage</title>
<style>

    h1 {
      color: navy;
    }

</style>
</head>
```

C'est logique : le CSS n'est pas du contenu à afficher, c'est une consigne d'apparence. Il n'a rien à faire dans le `<body>`.

> ⚠ **ATTENTION** : tout le CSS s'écrit **entre** `<style>` et `</style>`. Une règle qui traîne dans le `<body>` s'affichera comme du texte ordinaire dans la page — c'est l'erreur qui surprend le plus au premier essai.

L'anatomie d'une règle

```
h1 {
  color: navy;
}
```

Cette règle se lit : « pour tous les `<h1>` de la page, la couleur du texte est bleu marine ».

Elle contient six morceaux, et chacun compte :

Morceau	Ici	Rôle
Le **sélecteur**	`h1`	Qui est concerné
Les **accolades**	`{ }`	Elles délimitent la règle
La **propriété**	`color`	Ce qu'on modifie
Les **deux-points**	`:`	Ils séparent la propriété de sa valeur
La **valeur**	`navy`	La nouvelle apparence
Le **point-virgule**	`;`	Il termine la déclaration

Une règle peut contenir autant de déclarations que tu veux, chacune terminée par son point-virgule :

```css
h1 {
  color: navy;
  font-size: 32px;
  text-align: center;
}
```

> 💡 **ASTUCE :** le point-virgule de la **dernière** déclaration est facultatif — mais mets-le quand même. Le jour où tu ajoutes une ligne en dessous, tu ne te feras pas piéger.

Le sélecteur `h1` désigne **toutes** les balises `<h1>` de la page. Si tu en avais dix, les dix deviendraient bleu marine. C'est ce qu'on appelle un **sélecteur d'élément** : il vise un type de balise.

Les couleurs

Deux propriétés, à ne pas confondre :

- `color` : la couleur du **texte** ;
- `background-color` : la couleur du **fond**.

```css
body {
  background-color: #f5f5f5;
}

h1 {
  color: navy;
}
```

Une couleur s'écrit de deux façons :

Écriture	Exemple	Remarque
Par son nom anglais	`navy`, `crimson`, `teal`	Simple, mais choix limité
En hexadécimal	`#16244a`	N'importe quelle couleur, précise au ton près

L'écriture hexadécimale commence par un `#` suivi de six caractères : deux pour le rouge, deux pour le vert, deux pour le bleu. Tu n'as pas à les calculer — n'importe quel sélecteur de couleur, dans VSCode ou en ligne, te donne le code à copier.

La police et la taille

```css
body {
  font-family: Verdana, Arial, sans-serif;
  font-size: 16px;
}
```

`font-family` reçoit **plusieurs** polices, séparées par des virgules. Ce n'est pas un caprice : c'est une liste de secours. Le navigateur essaie Verdana ; si l'ordinateur ne l'a pas, il essaie Arial ; en dernier recours, il prend n'importe quelle police sans empattement (`sans-serif`).

> ℹ️ Termine toujours ta liste par une famille générique : `sans-serif` (sans empattement, comme Arial), `serif` (avec empattements, comme Times) ou `monospace` (chasse fixe, comme dans un éditeur de code).

`font-size` se donne en **pixels**. Une valeur de 16 px correspond à peu près à la taille de lecture confortable par défaut.

L'héritage : la propriété qui descend

Pourquoi écrire ces deux propriétés sur `body` plutôt que sur chaque balise ?

Parce que certaines propriétés se transmettent des parents vers les enfants. On appelle ça l'**héritage**. Tout ce qui se trouve dans le `<body>` — donc toute la page — reçoit sa police et sa taille de texte, sans qu'on ait à le répéter.

```css
body {
  font-family: Verdana, Arial, sans-serif;
  color: #333333;
}
```

Trois lignes, et toute la page change de police et de couleur de texte.

Cela ne vaut pas pour tout, cependant. Les propriétés liées au texte s'héritent : `color`, `font-family`, `font-size`. Celles qui concernent la boîte — bordures, fonds, espacements — ne s'héritent pas. Un `background-color` posé sur le `body` colore le fond de la page, mais ne colore pas chaque paragraphe individuellement.

© 2025 Mr. Lambrechts https://v2.ttrinfo.be 4 | 6

Encadrer et aérer

```css
p {
  border: 2px solid navy;
  padding: 12px;
}
```

`border` demande trois valeurs, dans cet ordre : **l'épaisseur**, **le style**, **la couleur**.

Style	Effet
`solid`	Trait continu
`dashed`	Tirets
`dotted`	Pointillés

`padding` ajoute de l'espace **à l'intérieur** de la bordure, entre le trait et le texte. Sans lui, un encadré serre son texte contre le cadre et devient désagréable à lire.

> 💡 **ASTUCE :** dès que tu poses une bordure, pose un `padding` dans la foulée. Les deux vont presque toujours ensemble.

Le cas du lien

Par défaut, un lien est bleu et souligné. Pour retirer le soulignement :

```css
a {
  color: #3aa86a;
  text-decoration: none;
}
```

C'est courant, mais ce n'est pas anodin. Le soulignement est le signal qui dit « ceci est cliquable ». Si tu l'enlèves, il faut que **la couleur** rende le lien franchement reconnaissable au milieu du texte.

> ⚠️ **ATTENTION :** un lien vert sombre au milieu d'un texte gris sombre, sans soulignement, devient invisible — en particulier pour une personne daltonienne. Ne retire le soulignement que si le contraste de couleur fait clairement le travail.

🧪 Vérifications

1. Dans quelle balise, et dans quelle partie de la page, écrit-on le CSS ?
2. Nomme les quatre parties de la déclaration `color: navy;`.
3. Pourquoi écrit-on plusieurs polices dans `font-family` ?
4. Tu écris `color: crimson;` sur le `body`. Tes paragraphes deviennent-ils rouges ? Et si tu écris `border: 1px solid crimson;` ?

✍️ À retenir

- Le HTML dit **ce que c'est**, le CSS dit **à quoi ça ressemble**.
- Le CSS s'écrit entre `<style>` et `</style>`, dans le `<head>`.
- Une règle, c'est un **sélecteur**, des **accolades**, et des déclarations `propriété: valeur;`.
- Un sélecteur d'élément comme `p` vise **toutes** les balises de ce type.
- Les propriétés de texte s'**héritent** : posées sur `body`, elles descendent dans toute la page.

Suite

Un problème se pose déjà : le sélecteur `p` touche **tous** les paragraphes. Comment n'en habiller qu'un seul ? C'est le rôle des classes.