

Les classes

Le sélecteur d'élément habille tous les paragraphes d'un coup. Pour n'en habiller qu'un seul, il faut lui donner une étiquette : c'est la classe.

4TTR

 Découverte

Tu veux mettre en évidence **un** paragraphe de ta page : celui qui contient le conseil important. Tu écris :

```
1 | p {  
2 |   background-color: #fff8e1;  
3 |   border-left: 4px solid #ffd15c;  
4 | }
```

Et tous les paragraphes de la page se retrouvent en jaune. Le sélecteur `p` ne fait pas de détail : il vise le type de balise, donc toutes les balises de ce type.

Il te manque un moyen de dire : « celui-là, et pas les autres ».



Objectifs

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

1. Poser une classe sur un élément HTML.
2. Écrire le sélecteur CSS correspondant.
3. Réutiliser la même classe sur des balises de types différents.
4. Choisir un nom de classe qui restera compréhensible dans six mois.

Une étiquette sur l'élément

La **classe** est un attribut, comme `href` ou `src`. Tu l'écris dans la balise ouvrante, et tu choisis toi-même sa valeur.

```
1 | <p class="conseil">Utilise un mot de passe différent pour chaque compte.</p>  
2 | <p>Ce paragraphe reste normal.</p>
```

Le premier paragraphe porte désormais l'étiquette `conseil`. Le second n'a rien.

En CSS, on vise cette étiquette avec un **point** devant son nom :

```
1 | .conseil {  
2 |   background-color: #fff8e1;  
3 |   border-left: 4px solid #ffd15c;
```

```
4 padding: 12px;
5 }
```

Seul le paragraphe étiqueté change d'apparence. L'autre reste tel quel.

⚠ **ATTENTION** : le point ne s'écrit **que** dans le CSS. Dans le HTML, tu écris `class="conseil"`, sans point. C'est l'erreur numéro un des débuts : `class=".conseil"` ne fonctionne pas.

Retiens la paire :

Dans le HTML	Dans le CSS
<code>class="conseil"</code>	<code>.conseil { ... }</code>
<code>class="alerte"</code>	<code>.alerte { ... }</code>

La même classe sur des balises différentes

C'est là que les classes deviennent vraiment utiles : une classe n'est **pas** liée à un type de balise. La même étiquette peut se poser sur un titre, sur un paragraphe, sur un élément de liste.

```
1 <h3 class="alerte">Ne clique jamais sur un lien inconnu</h3>
2
3 <p>Le reste du texte garde son apparence normale : il ne porte pas l'étiquette.</p>
4
5 <ul>
6   <li class="alerte">Vérifie l'adresse avant de cliquer</li>
7   <li>Installe les mises à jour</li>
8   <li>Active la vérification en deux étapes</li>
9 </ul>
```

```
1 .alerte {
2   color: #b3261e;
3   font-weight: bold;
4 }
```

Ne clique jamais sur un lien inconnu

Le reste du texte garde son apparence normale : il ne porte pas l'étiquette.

- **Vérifie l'adresse avant de cliquer**

- Installe les mises à jour
- Active la vérification en deux étapes

Le titre et le premier élément de la liste deviennent rouges. Pourtant, `<h3>` et `<li>` n'ont rien à voir l'un avec l'autre : c'est bien l'étiquette qui décide, pas le type de balise.

L'inverse est vrai aussi : un élément peut porter **plusieurs** classes, séparées par une espace.

```
1 <p class="conseil alerte">Change ce mot de passe aujourd'hui.</p>
```

Ce paragraphe reçoit à la fois le fond jaune de `.conseil` et le rouge gras de `.alerte`. Les deux règles s'appliquent, elles s'additionnent.

Bien nommer ses classes

Un nom de classe est libre : lettres, chiffres, tirets. Pas d'espace (l'espace sépare deux classes), pas d'accent, et jamais de chiffre en première position.

Mais le vrai enjeu n'est pas la syntaxe : c'est le **choix du mot**.

❌ À éviter	✅ Préférable	Pourquoi
<code>.rouge</code>	<code>.alerte</code>	Le jour où les alertes passent en orange, <code>.rouge</code> devient un mensonge
<code>.p2</code>	<code>.conseil</code>	<code>.p2</code> ne dit rien à personne, pas même à toi
<code>.grosTitreBleu</code>	<code>.titre-section</code>	Trois informations d'apparence figées dans un nom

🧠 **À RETENIR** : nomme la classe d'après le **rôle** de l'élément, pas d'après son apparence du moment. L'apparence, c'est justement ce que le CSS doit pouvoir changer sans que tu touches au HTML.

Et l' `id` ?

Tu croiseras un cousin de la classe : l'attribut `id`.

```
1 | <h1 id="titre-principal">Introduction à la cybersécurité</h1>
```

```
1 | #titre-principal {
2 |   color: navy;
3 | }
```

Même principe, deux différences :

- il se vise avec un **dièse** (`#`) au lieu d'un point ;
- il doit être **unique dans la page** : un seul élément peut porter un `id` donné.

Pour la mise en forme, on préfère largement les classes — elles sont réutilisables. L' `id` sert surtout à désigner un endroit précis de la page, par exemple comme destination d'un lien : `<a href="#titre-principal">`.

Quelques lignes pour finir proprement

Voici cinq déclarations qui, sans rien t'apprendre de nouveau sur les sélecteurs, transforment l'allure d'une page. Tu peux les recopier telles quelles.

```
1 body {  
2   max-width: 700px;  
3   margin: auto;  
4   line-height: 1.6;  
5   padding: 24px;  
6   background-color: #f5f5f5;  
7 }
```

- `max-width: 700px` empêche les lignes de s'étirer sur toute la largeur d'un grand écran. Au-delà d'une certaine longueur, l'œil perd le début de la ligne suivante.
- `margin: auto` centre ce bloc dans la fenêtre.
- `line-height: 1.6` aère l'espace entre les lignes. C'est le réglage qui, à lui seul, rend un texte agréable à lire.



`max-width` et `margin` sont des propriétés de **boîte** : elles pilotent la place que prend un élément. C'est tout le sujet de la partie suivante du parcours — pour l'instant, contente-toi d'en profiter.



Vérifications

1. Écris le HTML et le CSS pour qu'un seul paragraphe de la page ait un fond gris clair.
 2. Quelle est la différence d'écriture entre le nom de la classe dans le HTML et dans le CSS ?
 3. Peut-on poser la classe `.encadre` sur un `<h2>` et sur un `<li>` en même temps ?
 4. Pourquoi `.rouge` est-il un mauvais nom de classe ?
-



À retenir

- `class` est un **attribut** : `<p class="conseil">`, sans point.
 - Le sélecteur correspondant s'écrit avec un point : `.conseil { ... }`.
 - Une classe peut être posée sur **n'importe quelle** balise, et sur autant d'éléments que tu veux.
 - Un élément peut porter **plusieurs** classes, séparées par une espace.
 - Nomme tes classes d'après le **rôle**, jamais d'après la couleur.
-

Suite

Tu sais maintenant écrire une page complète : la structurer, la relier, l'habiller, et cibler précisément l'élément que tu veux modifier. C'est la fin de la première partie du parcours.

La suite s'attaque à ce que ce cours a effleuré à la dernière minute : les **boîtes**. Où commence un élément, où il finit, combien de place il prend, et comment les ranger les uns à côté des autres.