

Les titres et les paragraphes

Structurer un texte en HTML, c'est dessiner son plan : des titres qui se hiérarchisent, des paragraphes qui découpent, et quelques mots mis en valeur.

4TTR

 Découverte

Un texte sans structure ne se lit pas. En HTML, on ne choisit pas une taille de police : on annonce ce que chaque morceau **est**, et le navigateur s'occupe de l'apparence.



Objectifs

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

1. Choisir le bon niveau de titre pour construire le plan d'une page.
2. Découper un texte en paragraphes.
3. Forcer un retour à la ligne au bon endroit.
4. Mettre un mot en valeur et expliquer pourquoi `<strong>` vaut mieux que `<b>`.

Six niveaux de titre

Le HTML propose six balises de titre, de `<h1>` à `<h6>`. Le **h** vient de GB *heading*, en-tête.

```
1 | <h1>Introduction à la cybersécurité</h1>
2 | <h2>C'est quoi, au juste ?</h2>
3 | <h3>Les trois choses à protéger</h3>
```

Elles ne servent **pas** à choisir une taille de texte. Elles dessinent le plan du document, comme les chapitres et les sections d'un livre :

- `<h1>` : le sujet de la page. **Un seul par page.**
- `<h2>` : les grandes parties.
- `<h3>` : les subdivisions d'un `<h2>`.

On descend d'un niveau à la fois. Après un `<h2>` vient un `<h3>`, jamais un `<h5>` — même si sa taille te plaît davantage. Pour la taille, il y aura le CSS.

🧠 **À RETENIR** : ce plan n'est pas décoratif. Google s'en sert pour comprendre de quoi parle ta page, et un lecteur d'écran l'utilise pour permettre à une personne aveugle de sauter de section en section.

Le paragraphe

`<p>` — pour GB *paragraph* — découpe le texte en blocs. Chaque paragraphe s'affiche sur ses propres lignes, avec un peu d'espace au-dessus et en dessous.

```
1 | <p>Rome, trois jours en mai.</p>
2 | <p>Lisbonne, deux jours en juillet.</p>
```

Voici le point qui déroute tout le monde au début : **le navigateur ignore complètement les retours à la ligne que tu tapes dans ton code**. Tu peux appuyer dix fois sur Entrée, écrire ton paragraphe sur douze lignes ou sur une seule : le résultat à l'écran est identique.

```
1 | <p>Rome,
2 |     trois jours
3 |     en mai.</p>
```

s'affiche exactement comme :

```
1 | <p>Rome, trois jours en mai.</p>
```

Le navigateur remplace toute suite d'espaces et de retours à la ligne par une simple espace. Si tu veux un vrai saut de ligne, il faut le lui demander.

Le retour à la ligne, et les balises orphelines

`<br>` fait passer à la ligne suivante, à l'intérieur d'un paragraphe.

```
1 | <p>Rome,<br>trois jours en mai.</p>
```

Cette balise a une particularité : elle n'encadre rien du tout. Il n'y a pas de contenu à délimiter, donc **pas de balise fermante**. On l'appelle une balise **orpheline** (ou auto-fermante).

Les plus courantes :

Balise	Ce qu'elle fait
<code>
</code>	Passe à la ligne

Balise	Ce qu'elle fait
<code><hr></code>	Trace un trait de séparation horizontal
<code></code>	Affiche une image
<code><meta></code>	Donne une information sur la page, dans le <code><head></code>

⚠ **ATTENTION** : écrire `</br>` est une erreur. Cette balise n'a rien à fermer.

N'abuse pas de `<br>`. Pour séparer deux idées, c'est un nouveau paragraphe qu'il faut, pas un saut de ligne. Le `<br>` sert aux cas où la coupure fait partie du texte lui-même : une adresse, un vers de poème.

Mettre un mot en valeur

Jusqu'ici, tes balises créaient des **blocs** : un titre, un paragraphe occupent toute la largeur et poussent la suite plus bas. D'autres balises s'utilisent **au milieu d'une phrase**, sur quelques mots seulement.

```
1 | <p>Départ le <b>12 mai</b>, retour le 15.</p>
```

Quatre balises se ressemblent, deux à deux :

Balise	Rendu	Sens
<code></code>	gras	Fait ressortir visuellement, sans plus
<code></code>	gras	Signale que le mot est important
<code><i></code>	italique	Met en italique, sans plus
<code></code>	italique	Signale une insistance dans la phrase

À l'écran, `<b>` et `<strong>` donnent le même résultat. La différence est invisible pour toi, mais pas pour tout le monde : un lecteur d'écran change d'intonation sur un `<strong>`, et l'ignore sur un `<b>`.

💡 **ASTUCE** : dans le doute, préfère `<strong>` et `<em>`. Même rendu, plus de sens.

Ce que ça donne

Voici les balises de ce cours réunies dans une page, avec un peu de mise en forme :

Introduction à la cybersécurité

Tu passes plusieurs heures par jour connecté : téléphone, console, ordinateur de l'école.
La cybersécurité, c'est l'ensemble des moyens qui empêchent quelqu'un d'autre d'y accéder.

C'est quoi, au juste ?

Protéger un système, ce n'est pas seulement l'empêcher d'être **piraté**. C'est aussi faire en sorte qu'il continue à fonctionner.

Les trois choses à protéger

La confidentialité, l'intégrité, la disponibilité.

Le titre principal, les deux niveaux de sous-titres, les paragraphes et le mot en gras : cinq balises, et le texte est déjà lisible.



Vérifications

1. Combien de `<h1>` doit contenir une page ? Pourquoi ?
2. Tu tapes trois retours à la ligne dans ton code, entre deux phrases d'un même paragraphe. Que voit le visiteur ?
3. Quelle est la différence entre couper avec `<br>` et ouvrir un nouveau `<p>` ?
4. Pourquoi `<strong>` est-il préférable à `<b>` ?



À retenir

- Les titres `<h1>` à `<h6>` dessinent le **plan** de la page, pas sa mise en forme.
- Un seul `<h1>` par page, et on descend d'un niveau à la fois.
- Le navigateur **ignore** les espaces et les retours à la ligne de ton code.
- `<br>` est une balise **orpheline** : pas de contenu, donc pas de fermeture.
- `<strong>` et `<em>` portent un sens, là où `<b>` et `<i>` ne font que décorer.

Suite

Ton texte a un plan. Il lui manque de quoi énumérer : les listes à puces et les listes numérotées.

