Les conteneurs : div et span

Certaines balises ne veulent rien dire du tout. C'est justement à ça qu'elles servent : emballer un groupe d'éléments, ou attraper quelques mots au milieu d'une phrase.

4TTR Découverte

Toutes les balises vues jusqu'ici portent un sens. `<h1>` annonce un titre, `<p>` un paragraphe, `<ul>` une liste. Le navigateur, Google et les lecteurs d'écran s'appuient sur ce sens.

Mais il arrive qu'on ait besoin d'une balise qui ne dise **rien**. Juste un emballage, pour tenir plusieurs éléments ensemble et les décorer d'un coup.

🎯 Objectifs

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

1. Regrouper plusieurs éléments dans un `<div>`.
2. Attraper quelques mots d'une phrase avec un `<span>`.
3. Expliquer la différence entre un élément **bloc** et un élément **en ligne**.
4. Rendre une boîte visible pour comprendre où elle commence et où elle finit.

Le div, une boîte vide

`<div>` — pour GB ***division*** — n'a aucune signification propre. C'est un conteneur, et rien d'autre.

```
<div class="intro">
  <p>On appelle « virus » à peu près tout ce qui infecte un ordinateur.</p>
  <p>En réalité, il existe plusieurs familles.</p>
</div>
```

À l'écran, rien ne change. Les deux paragraphes s'affichent exactement comme avant.

Alors à quoi bon ? À ce que les deux paragraphes forment désormais **un seul objet**. Tu peux leur donner un fond commun, les encadrer ensemble, les déplacer ensemble. Sans le `<div>`, il faudrait répéter le même travail sur chacun.

> 💡 **ASTUCE :** un `<div>` porte presque toujours une classe. Un `<div>` sans classe est un emballage anonyme dont personne, toi compris, ne saura dire à quoi il sert.

Petit rappel au passage : `class="intro"` est une **étiquette** posée sur l'élément, et le CSS la vise avec un point devant son nom.

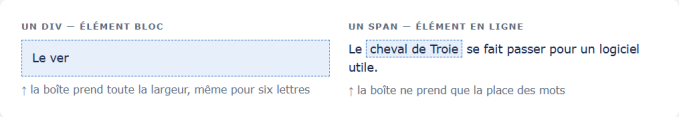
```
.intro {
  background-color: AliceBlue;
}
```

Bloc ou en ligne : la distinction qui explique tout

Applique ce fond bleu clair, et tu verras quelque chose d'inattendu : la boîte s'étale sur **toute la largeur** de la page, même si son texte tient en six lettres.

C'est normal. Les éléments HTML se répartissent en deux familles :

Famille	Exemples	Comportement
Bloc	`div` , `p` , `h1` , `ul` , `li`	Prend toute la largeur disponible, et pousse la suite à la ligne
En ligne	`span` , `a` , `b` , `strong` , `img`	Ne prend que la place de son contenu, et reste dans le fil du texte



> 🧠 **À RETENIR :** c'est la distinction la plus importante du CSS. Elle explique pourquoi une largeur s'applique à un `<div>` mais semble ignorée sur un `<b>` : un élément en ligne ne décide pas de sa largeur, il épouse son texte.

Le span, la version en ligne

`<span>` est le jumeau en ligne du `<div>` : aucune signification, mais il se glisse **au milieu d'une phrase**.

```
<p>Le <span class="terme">cheval de Troie</span> se fait passer pour un logiciel utile.</p>
```

```
.terme {
  color: #b3261e;
  font-weight: bold;
}
```

Il sert quand aucune balise existante ne convient. Pour un mot important, `<strong>` est meilleur, parce qu'il porte du sens. Mais pour colorer un terme de vocabulaire, mettre une date en évidence, marquer une unité — `<span>` est l'outil.

Tu veux…	Utilise
Signaler un mot **important**	`<strong>`
Marquer une **insistance**	`<em>`
Habiller quelques mots, sans autre intention	`<span>`
Regrouper plusieurs éléments entiers	`<div>`

Rendre les boîtes visibles

Quand une mise en page part de travers, le premier réflexe des développeurs est toujours le même : **colorer les boîtes** pour voir où elles sont.

```
.intro {
  background-color: AliceBlue;
}
```

Ou, plus radical, encadrer tout ce qui traîne :

```
div {
  border: 1px dashed red;
}
```

C'est laid, c'est temporaire, et c'est redoutablement efficace. Une boîte trop large, un élément qui déborde, un espacement inexplicable : tout devient visible d'un coup. Tu enlèves la règle quand tu as compris.

> ℹ️ Les navigateurs proposent aussi un outil intégré pour ça. Clic droit sur un élément, puis « Inspecter » : le navigateur surligne la boîte et affiche ses dimensions.

🧪 Vérifications

1. Que change un `<div>` à l'affichage, tant qu'on ne lui applique aucun style ?
2. Pourquoi un `<div>` contenant le mot « ver » occupe-t-il toute la largeur de la page ?
3. Tu veux colorer trois mots au milieu d'un paragraphe. `<div>` ou `<span>` ?
4. Cite deux balises de la famille « bloc » et deux de la famille « en ligne ».

✍️ À retenir

- `<div>` et `<span>` n'ont **aucune signification** : ce sont des conteneurs.
- `<div>` est un élément **bloc** : toute la largeur, et la suite passe à la ligne.
- `<span>` est un élément **en ligne** : il épouse son contenu et reste dans la phrase.
- Un conteneur sans classe ne sert à rien : donne-lui un nom.
- Colorer les boîtes est la méthode la plus rapide pour comprendre une mise en page.

Suite

Tu sais fabriquer des boîtes. Reste à savoir de quoi une boîte est faite : trois couches d'espace autour du contenu, et deux d'entre elles se confondent tout le temps.