Créez votre site web avec HTML5 et CSS3

Zotrim Uka

Contents

Tableau de bord :

Séances	Date	Début	Fin	Total
1ère	8 avril 2023	12h50	17h00	4h10
2ème	12 avril 2023	00h00	04h30	4h30
3ème	12 avril 2023	13h00	17h00	4h00
4ème	12 avril 2023	20h40	22H15	1H35
5ème	13 avril 2023	1h15	3h30	2h15
				16h30

Avant-propos :

- en **HTML**, vous utiliserez des **balises qui permettent de décrire votre contenu** - vous les écrirez entre chevrons `< >` ;
- en **CSS**, vous utiliserez des **propriétés qui permettent d'appliquer du style à des éléments HTML** - vous les écrirez entre accolades `{ }`.

1. **un fichier de contenu en HTML** - qui aura l'extension `.html` ;
2. et **un autre fichier de style en CSS** - qui aura l'extension `.css`.

Deux fichiers différents, mais il suffit d'ajouter une seul ligne de code, pour que les 2 fonctionnent en meme temps.

Objectifs pédagogiques

À la fin de ce cours, vous serez capable de :

- Maîtriser les bases de HTML5.
- Faire de la mise en forme avec CSS3.
- Agencer le contenu des pages.
- Utiliser des fonctionnalités avancées de HTML et CSS.

1. Tirez un maximum de ce cours

Professeurs :

- Mathieu Nebra
- Alexia Toulmet

Fil rouge du cours :

- se présenter ;
- mettre son travail en valeur grâce à un portfolio ;
- être contactée plus simplement, en ligne.

Les outils :

- vsCode / webstrom (jetbrains)
- github
- CodePen

Les modalités pédagogiques du cours :

- Le cours est divisé en **4 parties** de **plusieurs chapitres**, qu'il est préférable de suivre dans l'ordre.
- Chaque chapitre commence par une **vidéo**. Vous y trouverez des démonstrations de code essentielles, et des explications.
- Sous chaque vidéo, vous trouverez du **texte**, qui vous permettra d'approfondir les notions importantes, et un exercice "À vous de jouer" ainsi que son corrigé, ce qui vous permettra de construire un site web, chapitre après chapitre.
- Pour vous aider à pratiquer, à comprendre et à vous familiariser avec des structures et syntaxes de code, vous trouverez des **bacs à sable de code** via des liens CodePen que l'on vous donnera.
- Enfin, chaque partie se conclut par un **quiz** où vous pourrez vérifier vos acquis, et déterminer les points sur lesquels vous devez revenir.

Fiche de cours :

Fiche récap : Créez votre site web avec HTML5 et CSS3 — Développement

Syntaxe HTML

```html
<!DOCTYPE html>
  <html lang="fr">
     <head>
         <meta charset="utf-8">
         <title>Le titre de ma page web</title>
         <link href="style.css" rel="stylesheet">
     </head>
     <body>
         <h1 class="gros-titre">Bienvenue !</h1>
         <p>Mon paragraphe de texte</p>
     </body>
  </html>
```

Syntaxe CSS

```css
body {
  font-family: Arial;
  color: black;
}

.gros-titre {
  font-size: 30px;
}

p:hover {
  text-decoration: underline;
}
```

Définitions

Balise HTML : `<balise>`
Élément de structure en HTML qui indique la nature d'un contenu, encadré par des chevrons : `< >`.
→ Une **balise en paire** s'ouvre avant et se ferme après le contenu de l'élément. Ex. : `<body>...</body>` ;
→ Une **balise orpheline** est auto-fermante et contient l'élément dans la même ligne. Ex. : `<input>` ou `<img>`.

Attribut : `attribut="valeur"`
Option d'une balise donnant une indication complémentaire. Situé dans la balise ouvrante, l'attribut est suivi du signe égal et précise entre guillemets la **valeur** à prendre en compte. Ex. : `<html lang="fr">`.

Propriété CSS : `sélecteur {propriété: valeur;}`
Instruction en CSS, située entre accolades : `{ }`. Elle se termine toujours par un point-virgule : `;`.
Selon la valeur qu'on lui associe, une propriété CSS caractérise le style d'un élément HTML. Cet élément HTML est appelé en amont de la propriété via un **sélecteur** (il s'agit du nom de la balise, sans les chevrons).
On peut appliquer plusieurs propriétés à un élément HTML et une propriété à plusieurs éléments HTML.

Classe : `.classe {propriété: valeur;}`
Marquage créé avec l'attribut `class` pour isoler un élément HTML. On l'appelle dans le CSS avec un point : `.`.
Cela permet d'appliquer un style à cet élément HTML en particulier : un seul titre de niveau 1 par exemple.

Pseudo-classe : `sélecteur:pseudo-classe {propriété: valeur;}`
Option en CSS qu'on ajoute à la suite d'un sélecteur pour appliquer un style dynamique à un élément HTML.
Ex. : `:hover` désigne le survol de la souris. On peut lister plusieurs pseudo-classes pour un sélecteur.

Bonnes pratiques

- ✅ Commenter des lignes de code pour donner une indication sur leur fonctionnement, si besoin.
  Un commentaire en HTML : `<!-- commentaire -->`
  Un commentaire en CSS : `/* commentaire */`
- ✅ Structurer du contenu de texte avec des balises de titre, en commençant toujours par le niveau 1 : `<h1> </h1>`
- ✅ Rédiger le texte alternatif d'une image avec l'attribut `alt` : `<img src="….jpg" alt="…">`

Erreurs classiques

- ❌ Utiliser des balises universelles `<span>` (inline) et `<div>` (block) quand d'autres balises, plus adaptées, existent :
  `<span class="important">`
  ...au lieu de `<strong>` ;
  `<div class="titre">`
  ...au lieu de `<h1>` par exemple.
- ❌ Utiliser systématiquement des `position: absolute;` pour positionner des éléments pose des problèmes d'affichage : le contenu ne sera pas responsive.

2. Comprenez la différence entre HTML et CSS

Base du fonctionnement de tous les sites webs (langage web)

HTML :

- balisage
- Le HTML (HyperText Markup Language) a fait son apparition dès 1991 lors du lancement du Web.
- Décrire, structurer le contenu de la page
- Ajouter : des textes, des liens, des images

CSS :

- Description
- Agencement contenu
- Positionnement des images
- Choix couleurs
- Taille du texte
- Le CSS (Cascading Style Sheets, aussi appelées feuilles de style) a pour rôle de gérer l'apparence de la page web (agencement, positionnement, décoration, couleurs, taille du texte…). Ce langage est venu compléter le HTML en 1996, et il est toujours au fondement même du style du Web de nos jours.
- Le CSS a besoin d'une base HTML pour fonctionner

PAS UN PROGRAMME DE PROGRAMMATION

Lorsque je veux accèder à un site, le navigateur envoie une requête au serveur Et envoyer les données HTML de la page WEB. Le navigateur va interpreter le code HTML pour afficher le contenu.

A RETENIR :

Les différents navigateurs n'affichent pas toujours le même site exactement de la même façon ! Il faudra vous y faire, et prendre l'habitude de vérifier régulièrement que votre site fonctionne correctement sur les navigateurs les plus utilisés.

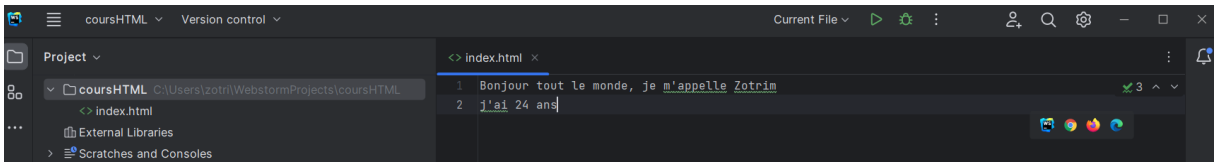
Tous les navigateurs embarquent des outils de développement particulièrment sophistiqués, notamment l'outil d'inspection d'une page web et faire des changements en temps réel.

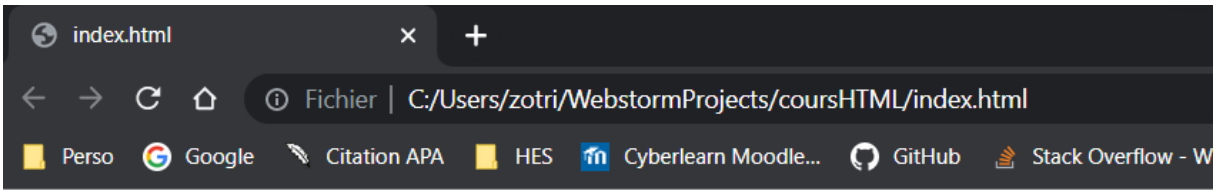
EN RÉSUMÉ

- Le HTML constitue la structure d'une page web.
- Le CSS permet d'ajouter du style.
- Les deux langages se complètent avec un rôle bien défini pour chacun.
- Le navigateur est un logiciel qui permet de lire les langages du Web : HTML et CSS.
- Tous les navigateurs embarquent des outils de développement, dont l'outil d'inspection qui permet d'accéder au HTML et au CSS d'une page.

3. Créez votre première page web en HTML

Rédiger une ligne de code :





Bonjour tout le monde, je m'appelle Zotrim j'ai 24 ans

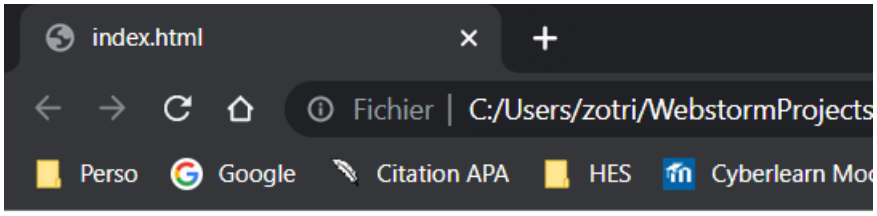
La phrase est sur la même ligne, alors que j'ai écrit sur 2 lignes

Pour créer une page web, il ne suffit pas de taper du texte. Il faut donner des instructions au navigateur : aller à la ligne, afficher une image, etc.

Pour ça, il faut utiliser des balises

1. Baliser la ligne de code :
   Des mots clés qu'il faut délimiter par des chevrons ouvrants et fermant:
   , <html>, <p>, <body>

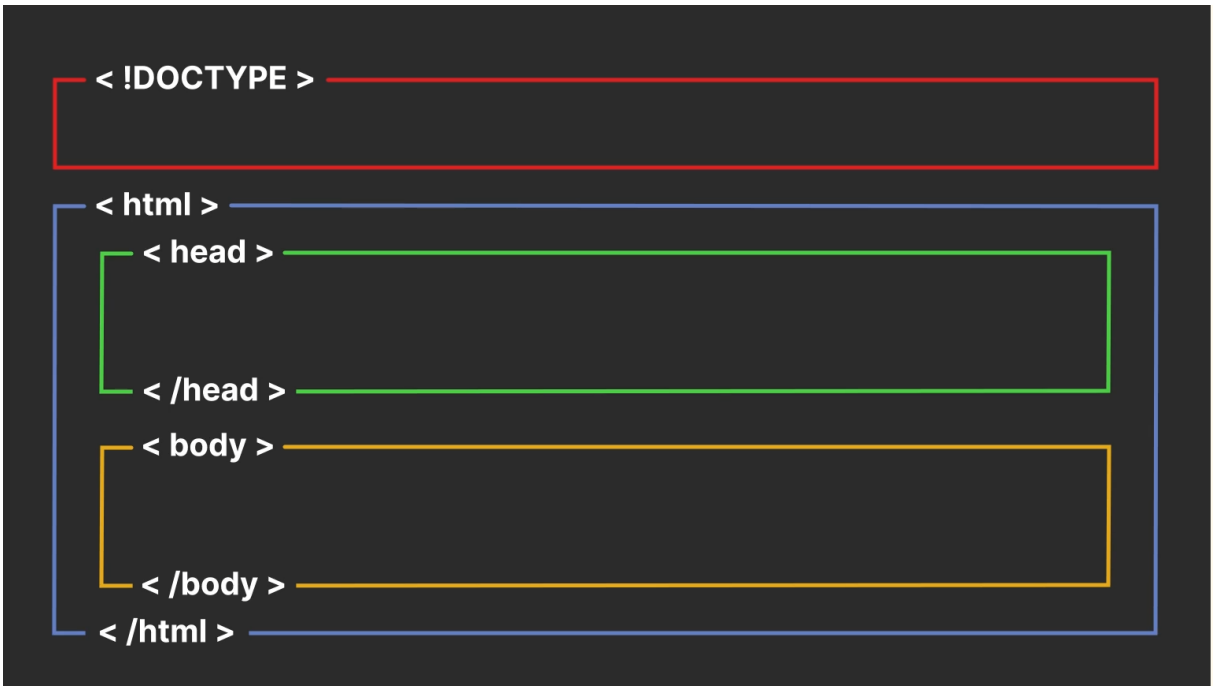
   j'utilise la balise <p> pour paragraphe :



Bonjour tout le monde, je m'appelle Zotrim

j'ai 24 ans

Connaître la structure de base HTML



HTML fonctionne avec des balises dans des balises

Elle s'ouvre et se ferme dans un ordre précis

La balise <html> est la première qu'on ouvre mais aussi la dernière qu'on ferme </html>

On appelle ça **une balise en pair** car elle a une balise ouvrante et fermante

Indentation n'est pas obligatoire mais nécessaire pour avoir un meilleur visuel

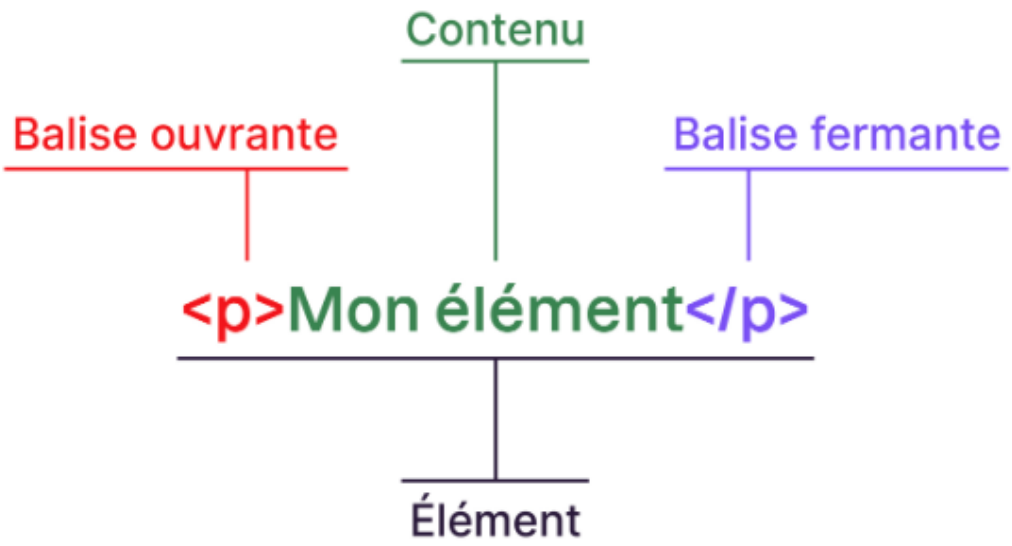
index.html

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="UTF-8">
5      <title>Title de ma page</title>
6  </head>
7  <body>
8  <p>Bonjour tout le monde, je m'appelle Zotrim</p>
9  <p>j'ai 24 ans</p>
10 </body>
11 </html>
```

Pour créer une page web en HTML, il suffit de créer un fichier ayant l'extension `.html`. Ce fichier s'ouvre dans le navigateur web simplement en faisant un double-clic dessus.

- **Balises HTML**

Le langage HTML utilise ce qu'on appelle des **balises**. On les écrit entre chevrons < et > :



Et elles servent à quoi les balises concrètement ?

Les balises indiquent la nature du texte qu'elles encadrent. Elles permettent au navigateur de comprendre ce qu'il faut afficher à l'écran pour les visiteurs d'un site web.

Si elles pouvaient parler, elles diraient :

<title> </title> : “Ceci est le titre de la page”,

<img>: “Ceci est une image”,

 : “Ceci est un paragraphe de texte”, etc.

Ah tiens, pourquoi certaines balises sont doublées et prennent un / dans leur syntaxe ?

On distingue deux types de balises :

Les balises en paires (une balise ouvrante et une balise fermante)

Et les balises orphelines (une seule balise).

Les balises en paires
Elles s'ouvrent, contiennent du texte, et se ferment plus loin. Si on prend la balise title qui correspond au titre de la page, voilà ce que ça nous donne :

<title>Ceci est le titre de ma page</title>
On a donc :

Une balise ouvrante : <title> ;

Et une balise fermante : </title> .

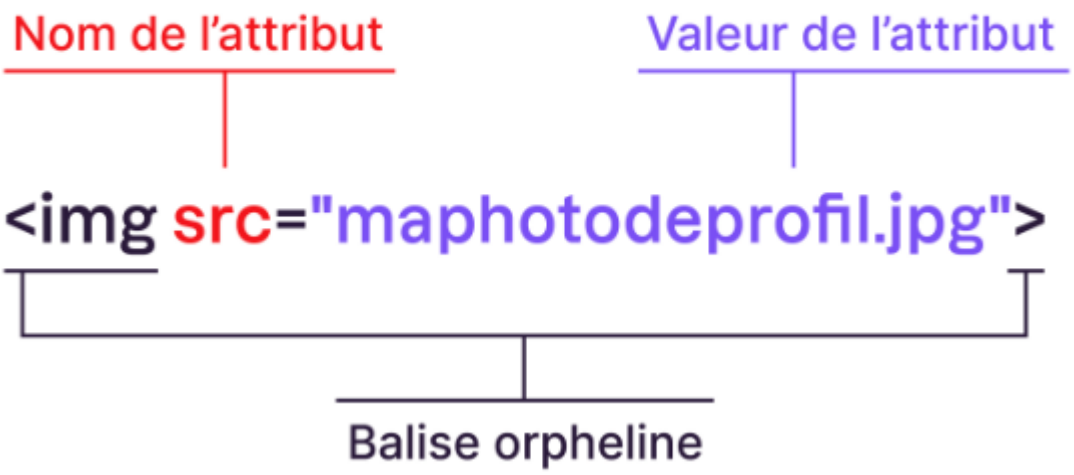
Cela délimite ce qui sera traduit par un titre. Pour l'ordinateur, tout ce qui n'est pas entre ces deux balises n'est pas un titre.

Les balises orphelines
Ce sont des balises qui servent le plus souvent à insérer un élément à un endroit précis (par exemple une image). Il n'est pas nécessaire de délimiter le début et la fin de l'image, on veut juste dire à l'ordinateur “Insère une image ici”. Il n'y a donc pas besoin de faire une balise ouvrante et une fermante, d'où l'appellation "balise orpheline".

Une balise orpheline s'écrit comme ceci : <img>

- **Paramétrez vos balises avec des attributs**
  Les attributs sont un peu les options des balises. Ils viennent les compléter pour donner des informations supplémentaires.

Un attribut est situé dans la balise ouvrante d'une balise en paire, ou directement dans une balise orpheline, comme c'est le cas ci dessous avec la balise `<img>` :



- **Utilisez la structure de base d'une page HTML**

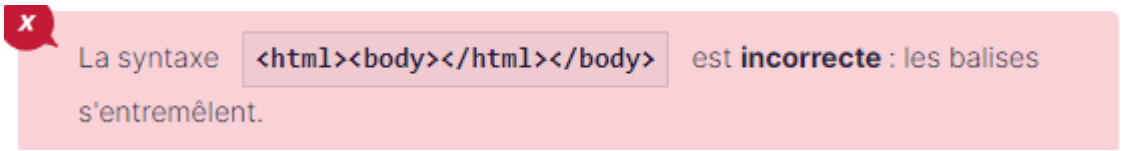
```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Le titre de ma page</title>
  </head>
  <body>
  </body>
</html>
```

C'est quoi : lang="fr"

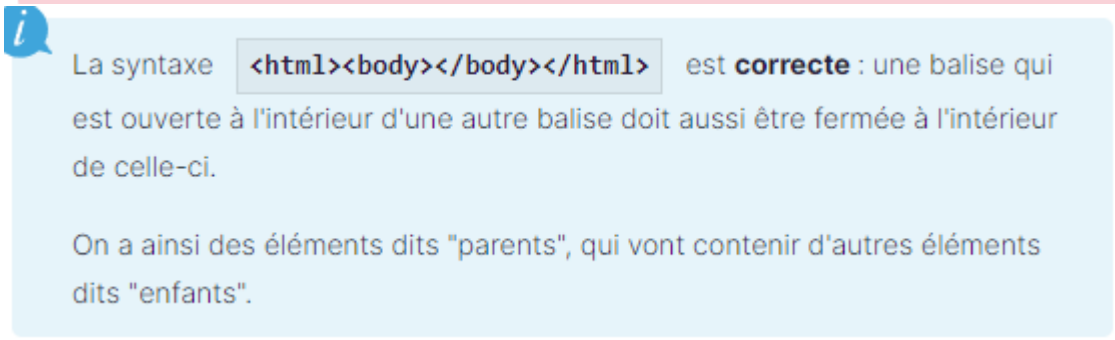
Il s'agit d'un attribut pour préciser la langue du site, ici notamment en français

Est-ce que l'ordre des balises est importantes ?

OUI ! Les balises s'ouvrent puis se ferment, et elles s'emboitent les unes dans les autres dans un ordre précis.



La syntaxe `<html><body></html></body>` est **incorrecte** : les balises s'entremêlent.



La syntaxe `<html><body></body></html>` est **correcte** : une balise qui est ouverte à l'intérieur d'une autre balise doit aussi être fermée à l'intérieur de celle-ci.

On a ainsi des éléments dits "parents", qui vont contenir d'autres éléments dits "enfants".

Voyons à quoi servent toutes ces balises.

La première ligne <!DOCTYPE html> est une balise orpheline indispensable : elle indique qu'il s'agit d'une page HTML.

La balise en paire <html> </html> englobe tout le contenu de la page web. A l'intérieur, il ya les balises en paire <head> </head> et <body> </body>

Pour rappel, c'est dans cette balise qu'on peut préciser la langue par défaut du site web :

- `<html lang="fr">` pour le français ;
- `<html lang="en">` pour l'anglais ;
- `<html lang="es">` pour l'espagnol...

Si besoin, voici **la liste complète des codes de langue**.

La balise en paire <head> </head> contient deux balises qui donnent des informations au navigateur : l'encodage et le titre de la page.

La balise orpheline <meta charset="utf-8"> indique l'encodage utilisé dans le fichier .html : cela détermine comment les caractères spéciaux s'affichent (accents, idéogrammes chinois et japonais, etc.). Si les accents ne s'affichent mal, c'est qu'il y a un problème d'encodage. Bien vérifier la balise meta indique bien UTF-8 et que le fichier est enregistré en UTF-8

La balise en paire <title> </title> indique au navigateur le titre de la page web. Toute page doit avoir un titre qui décrit ce qu'elle contient, il s'affichera dans l'onglet du navigateur, et apparaîtra dans les résultats de recherche, comme sur Google. Autant vous dire que bien choisir son titre est important !

Bonjour OpenClassrooms

- La balise en paire `<body> </body>` contient tout ce qui sera affiché à sur la page web.

La balise en paire <body> </body> contient tout ce qui sera affiché à l'écran sur la page web

- **Commentez votre code HTML**
  Un commentaire en HTML est un texte qui sert simplement de mémo. Il n'est pas affiché, il n'est pas lu par l'ordinateur, cela ne change rien à l'affichage de la page.

  Vous pouvez utiliser les commentaires pour laisser des indications sur le fonctionnement de votre page. Cela vous permettra de vous rappeler comment fonctionne votre page si vous revenez sur votre code source après un long moment d'absence.

  <!-- Ceci est un commentaire -->

  ATTENTION : tout le monde peut voir le code HTML Source, une fois mise en ligne. Les commentaires seront donc visibles. Ne pas mettre des infos sensibles comme des mot de passes dans les commentaires

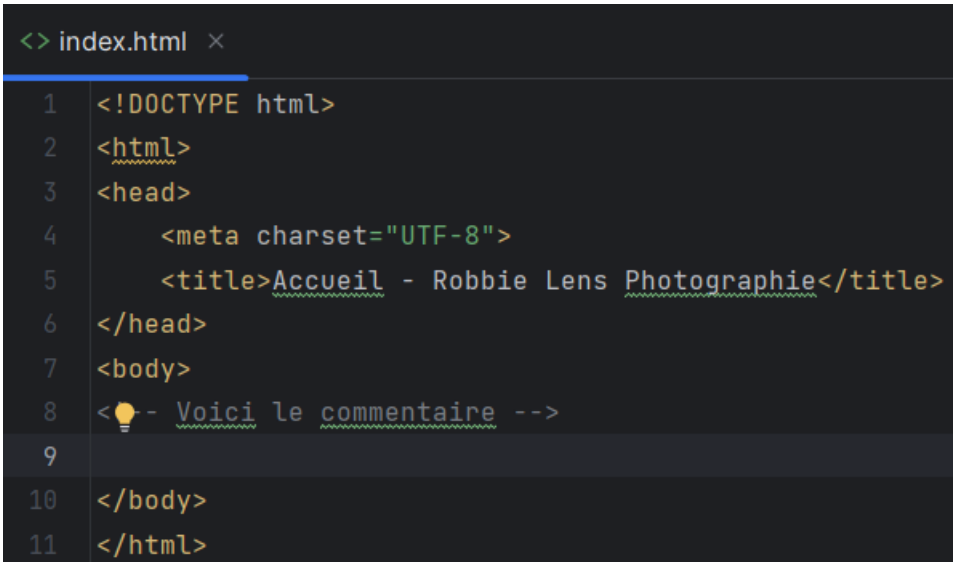
- **A vous de jouer !**
  Exercice :

  Pour cet exercice, vous allez devoir partir de votre fichier index.html que vous venez de créer pour :

  y insérer la structure de base HTML ;

  changer le contenu de la balise <title> </title> pour avoir “Accueil – Robbie Lens Photographie” ;

  écrire un commentaire dans <body> </body> .

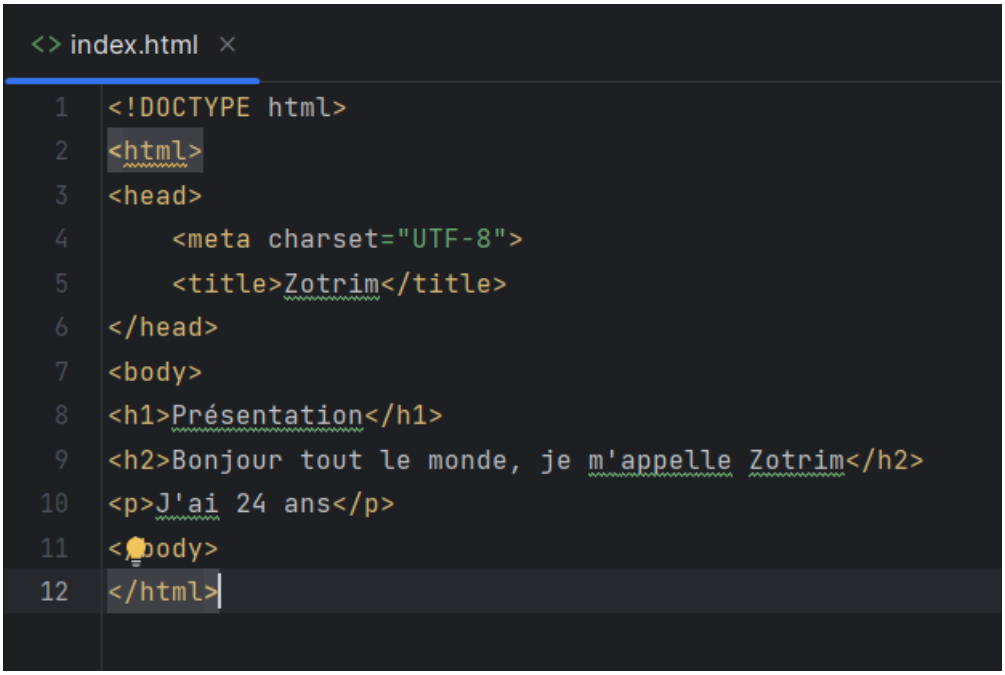


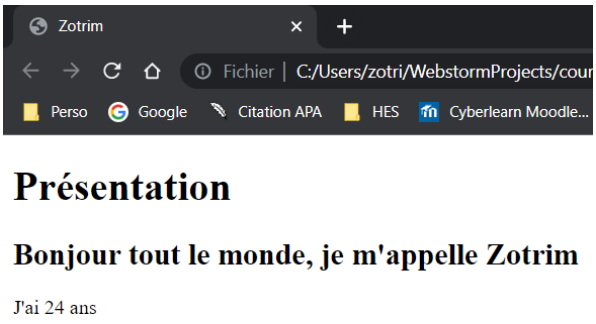
EN RÉSUMÉ :

- Pour créer une page web, on crée un fichier ayant l'extension .html , qui pourra être ouvert dans le navigateur web simplement en faisant un double-clic dessus.
- Chaque fichier HTML est constitué de balises.
- Les balises peuvent avoir plusieurs formes :
    - <balise> </balise> : **balises en paires**, elles s'ouvrent et se ferment pour délimiter le contenu (début et fin d'un titre, par exemple) ;
    - <balise> : **balises orphelines** (on ne les insère qu'en un seul exemplaire), elles permettent d'insérer un élément à un endroit précis (par exemple une image).
- Les balises sont parfois accompagnées d'attributs pour donner des indications supplémentaires, ou paramétrer un élément (exemple : <img src="photo.jpg">).
- Une page web est constituée de deux sections principales : l'en-tête<head> </head> dont le contenu n'apparaît pas dans l'affichage de la page et le corps <body> </body> qui, lui, apparaît.

4. Organisez votre texte

1. Corps de texte
   Les balises <p> je les place dans les balises <body>
2. Titres
   - Il faut structurer les paragraphes avec des titres
   - On peut faire plusieurs niveaux de titres
     - Niveau principal <h1> </h1>
     - Sous-titres <h2> </h2>
     - <h3> </h3>
     - <h4> </h4>
     - <h5> </h5>
     - <h6> </h6>

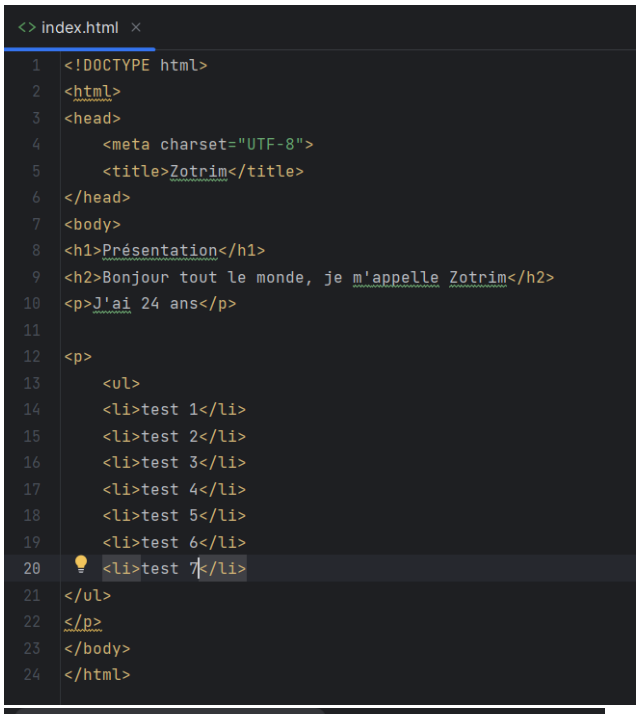


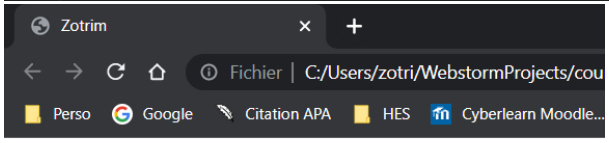


3. Listes

   Il existe de deux types de listes

   - Non-Ordonnée
     - Créer une liste non-ordonnée est très simple
     - Ajouter la balise
     - Ajouter autant de balises
     - Doit être dans la balise <body></body>





Présentation

Bonjour tout le monde, je m'appelle Zotrim

J'ai 24 ans

- test 1
- test 2
- test 3
- test 4
- test 5
- test 6
- test 7

   - Ordonnée
     - L'ordre est important
     - La même chose que non-ordonnée mais il faut changer la balise ul par

```
<p>
    <ol>
        <li>test 1</li>
        <li>test 2</li>
        <li>test 3</li>
        <li>test 4</li>
        <li>test 5</li>
        <li>test 6</li>
        <li>test 7</li>
</ol>
</p>
</body>
</html>
```

Le texte affiché sur une page web est compris entre les balises <body> </body>.

- **Créez des paragraphes avec les balises **
  Les balises ** **permettent de délimiter les paragraphes en HTML

  <body>
  Ceci est le contenu de mon premier paragraphe
  Ceci est le contenu de mon deuxième paragraphe
  </body>

- **Revenez à la ligne avec la balise orpheline
**
  Pour revenir à la ligne, on utilise la balise orpheline
 (pour break), on n'a donc pas besoin de la fermer :

  <body>
  Ceci est le contenu de mon premier paragraphe,
dont le contenu est particulièrement long.
  Ceci est le contenu de mon deuxième paragraphe

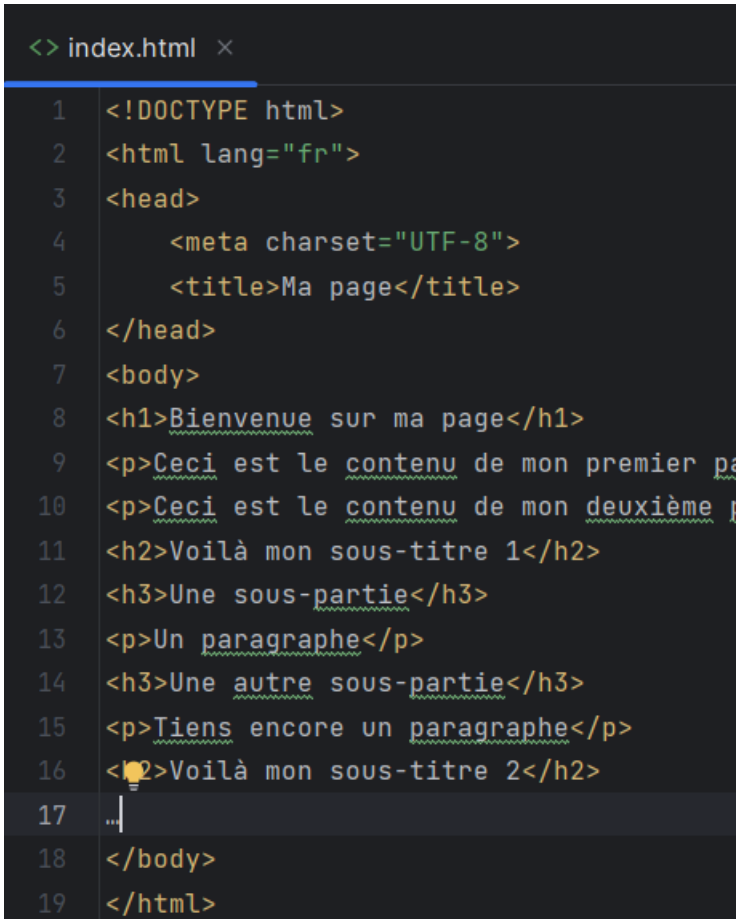
En théorie on peut mettre plusieurs balises
 d'affilée mais c'est une mauvaise pratique qui rend le code délicat à maintenir.

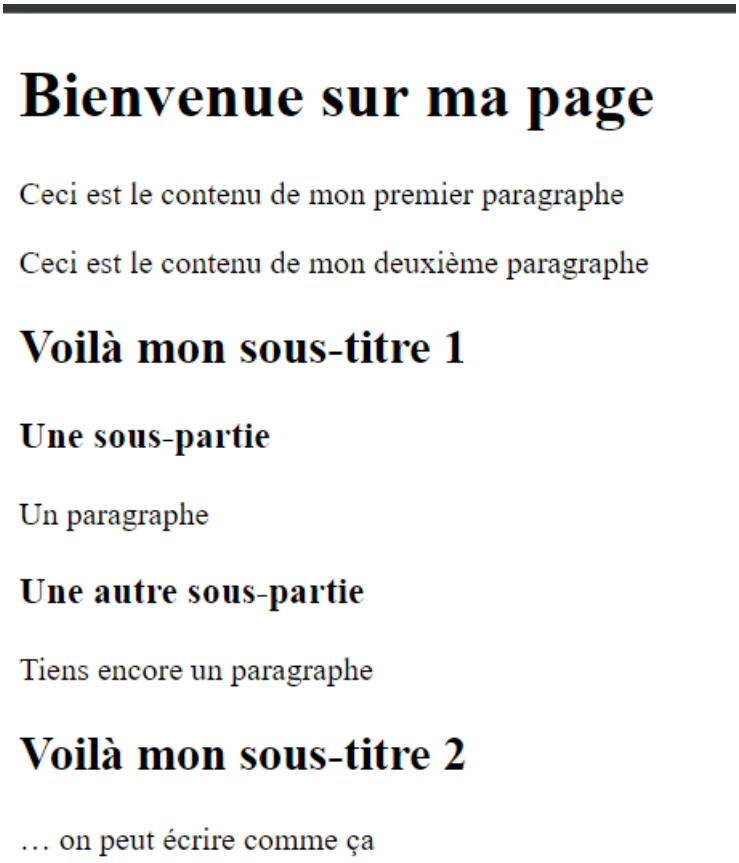
- **Créez des titres avec les balises <h1>, <h2>, etc**
  Les balises de titres vont de **<h1> </h1>** jusqu'à **<h6> </h6>**, ce qui permet de hiérarchiser et structurer le texte dans différentes sections, du niveau le plus grand, au niveau le plus petit.

  Toujours structurer sa page en commencant par h1 puis structurer à l'intérieur avec des titres de niv2 « h2 » et ainsi de suite. **Il ne devrait pas y avoir de sous-titre sans titre principal !**

  Ne pas confondre :

- La balise **<h1>** sert à créer un titre de niveau 1 qui sera affiché sur la page web.
- La balise **<title>**n'affiche rien sur la page web, elle affiche le titre de la page dans l'onglet du navigateur.





Pour modifier la taille du texte, on le fera en CSS

- **Créez des listes avec les balises <li> et <ul> ou **
  Etape 1 : balisez les éléments d''une liste avec pour « listed item » et on les insére dans une balise
  Etape 2 : insérez la liste de balises <ul> ou
  Ul = non ordonnée => liste à puces
  OL = ordonnée => liste numérotées donc ordre important

```
1  <h1>Les fruits rouges</h1>
2  <ul>
3      <li>Fraises</li>
4      <li>Framboises</li>
5      <li>Groseilles</li>
6  </ul>
7
8  <h1>Ma journée</h1>
9  <ol>
10     <li>Je me lève.</li>
11     <li>Je mange et je bois de l'eau.</li>
12     <li>Je retourne me coucher.</li>
13 </ol>
14
```

Les fruits rouges

- Fraises
- Framboises
- Groseilles

Ma journée

1. Je me lève.
2. Je mange et je bois de l'eau.
3. Je retourne me coucher.

- **Mettez en valeur du texte important**
  Différents moyens de mettre en valeur le texte
- Surligner le texte : <mark></mark>
- Mettre le texte en italique :
- Mettre le texte en gras :

Les balises **** et **** ne signifient pas respectivement “mettre en italique" ou "mettre en gras” mais seulement que le texte est “important”. On pourra décider plus tard, en CSS, d'afficher les mots “importants” d'une autre façon que le gras, si on le souhaite.

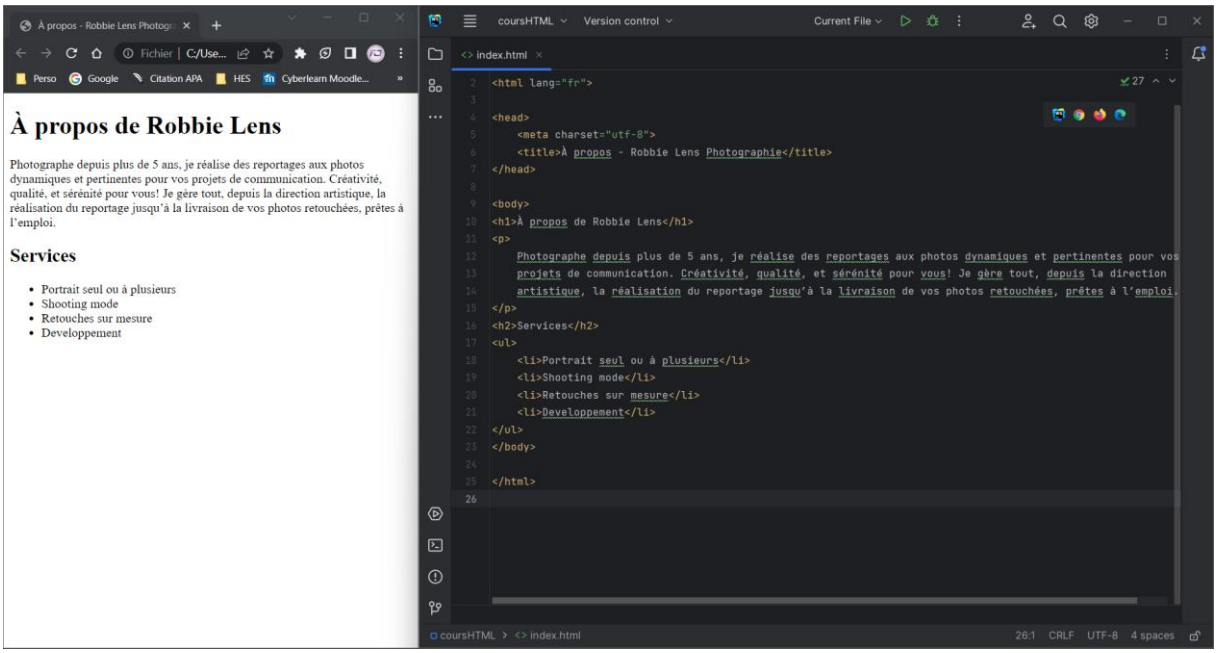
Soit, et si je n'ai pas envie de faire ressortir du texte important, c'est grave ?

Les robots de moteurs de recherche parcourent le Web en lisant le code HTML de tous les sites. Les mots-clés “importants”, mais aussi les titres (headings) hiérarchisés auront tendance à avoir plus de valeur à leurs yeux ; donc si quelqu'un fait une recherche sur ces mots, il a plus de chances de tomber sur votre site (votre site aura un meilleur référencement). Mais attention tout de même : abuser de ces balises n'aura pas l'effet escompté sur votre référencement.

Une nuance tout de même : abuser de ces balises n'aura pas l'effet escompté sur votre référencement. Si quasiment tout le texte est signalé comme étant important, rien en particulier ne va vraiment pouvoir "ressortir"... Pensez-y !

- **A vous de jouer :**
  Exercice :

1. créer le titre dans la page **a-propos.html**: "À propos de Robbie Lens" ;
2. créer le paragraphe associé : "Photographe depuis plus de 5 ans, je réalise des reportages aux photos dynamiques et pertinentes pour vos projets de communication. Créativité, qualité, et sérénité pour vous ! Je gère tout, depuis la direction artistique, la réalisation du reportage, jusqu'à la livraison de vos photos retouchées, prêtes à l'emploi." ;
3. créer un titre de niveau 2 pour ajouter une section nommée : "Services" ;
4. créer une liste non ordonnée pour lister les spécialités de Robbie Lens :
   - Portrait seul ou à plusieurs,
   - Shooting mode,
   - Retouches sur mesure,
   - Développement.



EN RÉSUMÉ :

- Le HTML comporte de nombreuses balises qui nous permettent d'organiser le texte de notre page. Ces balises donnent des indications comme “Ceci est un paragraphe”, “Ceci est un titre”, etc.
- Les paragraphes sont définis par la balise   , et les sauts de ligne par la balise orpheline
.
- Il existe six niveaux de titre, de <h1> </h1> à <h6> </h6>, à utiliser selon l'importance du titre.
- On peut mettre en valeur certains mots avec les balises , et <mark>.
- Pour créer des listes, on doit utiliser la balise <ul> (liste à puces, non ordonnée) ou <ol> (liste ordonnée). À l'intérieur, on insère les éléments avec une balise <li> pour chaque item.

5. Créez un lien hypertexte en HTML

Lien hypertexte : lien qu'on peut cliquer pour nous rédiger vers une autre page

1. Lien externe au site

```
<a href="https://www.google.ch/">Page de google</a>
```

   Page de google => le texte que verra le user pour cliquer sur le lien
2. Lien vers des pages internes au site
   Ça va marcher car les 2 fichiers sont en local, donc dans le même dossier
   Donc pour cela :
   Je créer un nouveau fichier : page2.html

Dans index.html je me le lien suivant :

```
<a href="page2.html">Ma 2ème page</a>
```

Maintenant, notre fichier page2.html et dans un dossier qui se nomme : contenu

```
<a href="contenu/page2.html">Ma 2ème page</a>
```

S'il est dans un dossier parent :

```
<a href="../page2.html">Ma 2ème page</a>
```

3. Lien vers des endroits précis du site
   Exemple sur un paragraphe, milieu de la page, etc.

   Ce type de lien s'appelle une « ancre »

   - D'abord définir le mot ou la phrase comme point de repère (un titre, sous-titre, ça peut être n'importe quoi)
   - Ajouter l'attribut => < h2> id=''mon_ancre'' Salut </h2>
   - Ensuite : <a href= ''#mon_ancre'' lien vers mon ancre

3.1 lien qui ouvre une nouvelle page qui amène directement à une ancre situer plus bas

- page2.html => fixer le repère et mettre id=''mon_ancre_finale''
- ensuite dans la page index.html : créer le lien :
  -
    lien vers mon ancre finale

On reconnaît facilement un **lien hypertexte** (ou **hyperlien**) sur une page web : par défaut, il est en bleu et souligné dans le navigateur et un curseur en forme de main apparaît lorsqu'on pointe dessus ; mais on peut modifier ce style en CSS

- **Créez un lien hypertexte avec la balise <a> et l'attribut href**
    1. on utilise la balise<a>(pour "anchor") pour indiquer qu'on va rediriger vers un autre endroit,
    2. puis, on ajoute l'attribut href suivi de=pour annoncer l'endroit vers lequel on veut rediriger,
    3. on indique explicitement entre " " l'endroit vers lequel le lien doit rediriger,
    4. enfin, on écrit le texte qui doit s'afficher sur l'hyperlien.
- **Créez un lien hypertexte vers l'URL d'une page disponible sur internet**
  un lien vers un site existant : Accédez à OpenClassrooms => il s'agit d'un lien absolu, il indique une adresse complète

- **Créez un lien hypertexte d'une page à une autre sur votre site**
  On a pas d'url disponible car le site n'est pas encore en ligne. On va créer des liens entre nos pages en utilisant leur nom et en indiquant leur arborescence dans notre dossier local.

  Ce type de lien hypertexte s'appelle un **lien relatif** : il indique où trouver notre fichier HTML.

  CAS 1 :
  Les pages dans le même dossier
  Si on veut aller de la page 1 à la page 2, voici ce que nous écrirons dans le fichier de **page1.html :**
  Page 2

  CAS 2 :
  Les 2 pages sont situées dans deux dossiers différents en local

  Si on veut créer un fichier page 3 et le déplacer dans un autre dossier, par exemple un dossier /contenu , on va donc indiquer le chemin à suivre pour trouver ce fichier :
  Page 3
  Et s'il y a plusieurs sous-dossiers, on écrira ceci :
  Page 3

  CAS 3 :
  Se trouve dans un dossier « parent »
  Page 3

- **Créez une ancre avec les attributs id et href**

Une ancre est un repère que l'on peut mettre dans une page HTML si elle est très longue, cela aide à la navigation et rend un contenu plus facile à parcourir. Cela permet par exemple aux visiteurs d'un site web d'aller directement à la partie qui les intéresse.

Ce comportement est typique d'un **site web "one page"** où tout se situe sur la même page.

Pour créer une ancre rajouter l'attribut « id » à une balise qui va servir de repère

La première étape consiste à ajouter l'attribut id suivi de = pour donner un nom à l'ancre entre " " :

Exemple : <h2 id="mon_ancre">Titre</h2>

L'attribut id sert à donner un nom "unique" à une balise, pour s'en servir de repère. Et, croyez-moi, vous n'avez pas fini d'entendre parler de cet attribut. Ici, on s'en sert pour faire un lien vers une ancre mais, en CSS, il pourra nous être utile pour repérer une balise précise, vous verrez.

Evitez de créer des id avec des espaces ou caractère spéciaux. La valeur doit être reconnue par tous les navigateurs.

La seconde étape consiste à indiquer où se situe l'ancre. La méthode pour se faire varie selon que :

1. l'ancre est plus bas sur la même page

   Dans ce premier cas, on crée un lien avec l'attribut href (il contient un dièse # suivi du nom de l'ancre) : Aller vers l'ancre

```
<h1>Ma grande page</h1>
<p>
Découvrez nos conseils d'aménagement pour :<br>
    <a href="#cuisine">La cuisine</a><br>
    <a href="#jardin">Le jardin</a><br>
    <a href="#salon">Le salon</a><br>
</p>
<h2 id="cuisine">La cuisine</h2>
<p>... (beaucoup de texte) ...</p>
<h2 id="jardin">Le jardin</h2>
<p>... (beaucoup de texte) ...</p>
<h2 id="salon">Le salon</h2>
<p>... (beaucoup de texte) ...</p>
```

2. l'ancre est située sur une autre page

   Dans ce cas-là, on tape le nom de la page cible avant le dièse # et enfin le nom de l'ancre : Le jardin

```
<h1>Le Mégamix</h1>
<p>
Rendez-vous quelque part sur la page :<br>
    <a href="index.html#cuisine">La cuisine</a><br>
    <a href="index.html#jardin">Le jardin</a><br>
    <a href="index.html#salon">Le salon</a><br>
</p>
```

Sachez qu'il est possible de configurer un lien pour qu'il ait un comportement un peu particulier.

`target="_blank"` fait en sorte que le lien hypertexte ouvre un nouvel onglet :

```
<p>Bonjour. Souhaitez-vous apprendre sur <a href="https://openclassrooms.com"
target="_blank">OpenClassrooms</a> ?</p>
```

`href="mailto:NOMDUMAIL@MAIL.COM"` crée un lien hypertexte qui ouvre la boîte mail avec un nouveau message vide.

`href="NOMDEFICHIER.EXTENSION"` crée un lien hypertexte qui permet de télécharger un fichier que vous avez placé au préalable dans le même dossier que votre page web.

EN RÉSUMÉ :

- Un lien hypertexte (ou hyperlien) permet de changer de page. Par défaut, il est en bleu et souligné dans le navigateur mais on peut modifier ce style en CSS.
- Pour faire un lien hypertexte vers un site web existant, on utilise la balise <a> avec l'attribut href pour indiquer **l'adresse de la page web cible**. Il s'agit d'un **lien absolu**. Exemple : <a href="https://openclassrooms.com"> .
- Pour faire un lien hypertexte vers une autre page de son site, on utilise la balise <a> avec l'attribut href pour indiquer **le nom du fichier en local**. Il s'agit d'un **lien relatif**. Exemple : <a href="page2.html"> .
- Un lien hypertexte peut aussi permettre d'amener vers un endroit précis d'une page. Il faut créer une ancre avec l'attribut id pour “marquer” cet endroit dans la page, puis faire un lien vers l'ancre comme ceci : <a href="#ancre"> .

Insérez des images

- ## Insérez une image avec la balise orpheline <img>

La balise qui permet d'insérer une image est une balise orpheline : <img>

Pour fonctionner correctement, elle a besoin de 2 attributs :

- src : cet attribut permet d'indiquer la source de l'image
- alt : cet attribut permet de donner à l'image une description alternative

On utilise un chemin absolu pour indiquer la source d'une image lorsqu'elle est en ligne (attention à bien vérifier que ça soit l'URL de l'image et non du site)

On utilise un chemin relatif si l'image est en local sur notre PC, on utilise le nom et l'arborescence du fichier ex : notre image et dans *images* => **src=''images/logo.png''**

/ ! \ éviter les accents, les maj, les espaces dans le nom du fichier

Idéalement :

- supprimer les espaces ou les remplacer par des « _ » et les accents
- tout mettre en minuscule => **images_du_site/image_toute_bete.jpg**

si l'image ne s'affiche pas = le chemin est incorrect

alt : c'est quoi ?

Elle permet de donner une description alternative à l'image (une bonne pratique)

Une description alternative est un court texte qui décrit ce que contient l'image. Ce texte alternatif sera :

- inscrit à la place de l'image si elle ne peut pas être affichée au moment du chargement de la page web (cela arrive) ;
- audio-décrit par les navigateurs de personnes présentant un handicap (trouble de la vue, non-voyants).
- Cela contribue donc à améliorer ce que l'on appelle l'accessibilité d'un site web. En plus, cela aide les robots des moteurs de recherche pour trouver des images.

- ## Choisissez le bon format d'image

Quel format adopter en fonction de l'image :

- Photo : utiliser un JPEG
- N'importe quelle image avec un peu de couleurs : moins de 256 (utiliser un png 8 bits ou un GIF)
- N'importe quelle image avec beaucoup de couleurs : utiliser un PNG 24 bits
- Une image animée : utiliser un GIF
- Un logo vectoriel : utiliser un SVG

- Ajouter une infobulle avec l'attribut => title

Afin d'afficher une bulle d'aide sur vos images, vous pouvez utiliser l'attribut title ; (à ne pas confondre avec la balise title qui permet d'indiquer au navigateur le titre d'une page web).

L'attribut title est facultatif, contrairement à alt.

Exemple :

```
<img src="montagnes.png" title="Alors, envie de vous balader n'est-ce pas ?"
alt="Chemin de randonnée au milieu des montagnes">
```

- Créez une miniature cliquable

Si image trop grosse, mettre miniature cliquable car le site mettra moins de temos à charger.

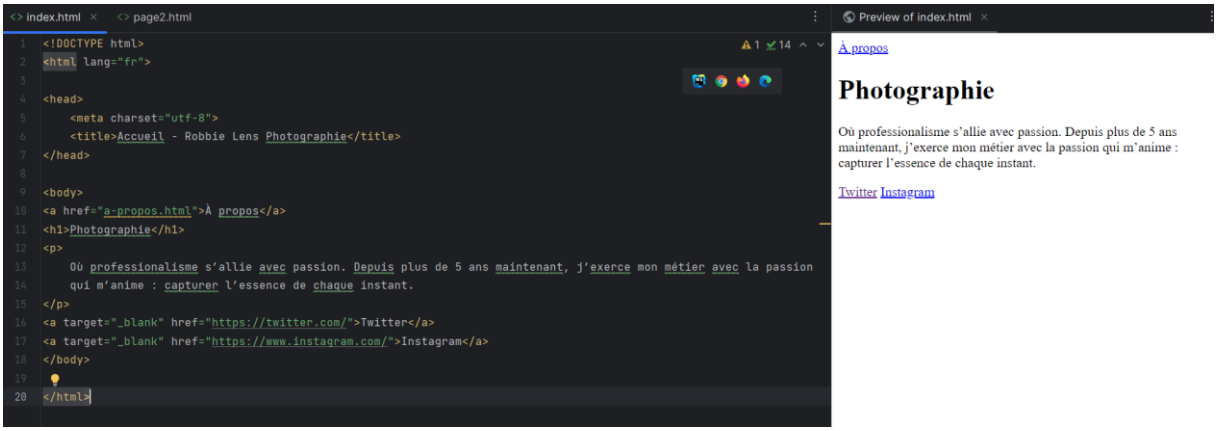
Il faut 2x la même image : la version originale et une version plus petite

1. placez vos deux images dans un dossier appelé par exemple **images** .
2. Faites afficher la version **montagne_mini.jpg** sur une page,
3. et faites un lien vers la version **montagne.jpg**.

   exemple :

```
<p>Vous souhaitez voir l'image dans sa taille d'origine ? Cliquez dessus !<br>
    <a href="images/montagne.jpg"><img src="images/montagne_mini.jpg" alt="Chemi
de randonnée au milieu des montagnes" title="Cliquez pour agrandir" ></a>
</p>
```

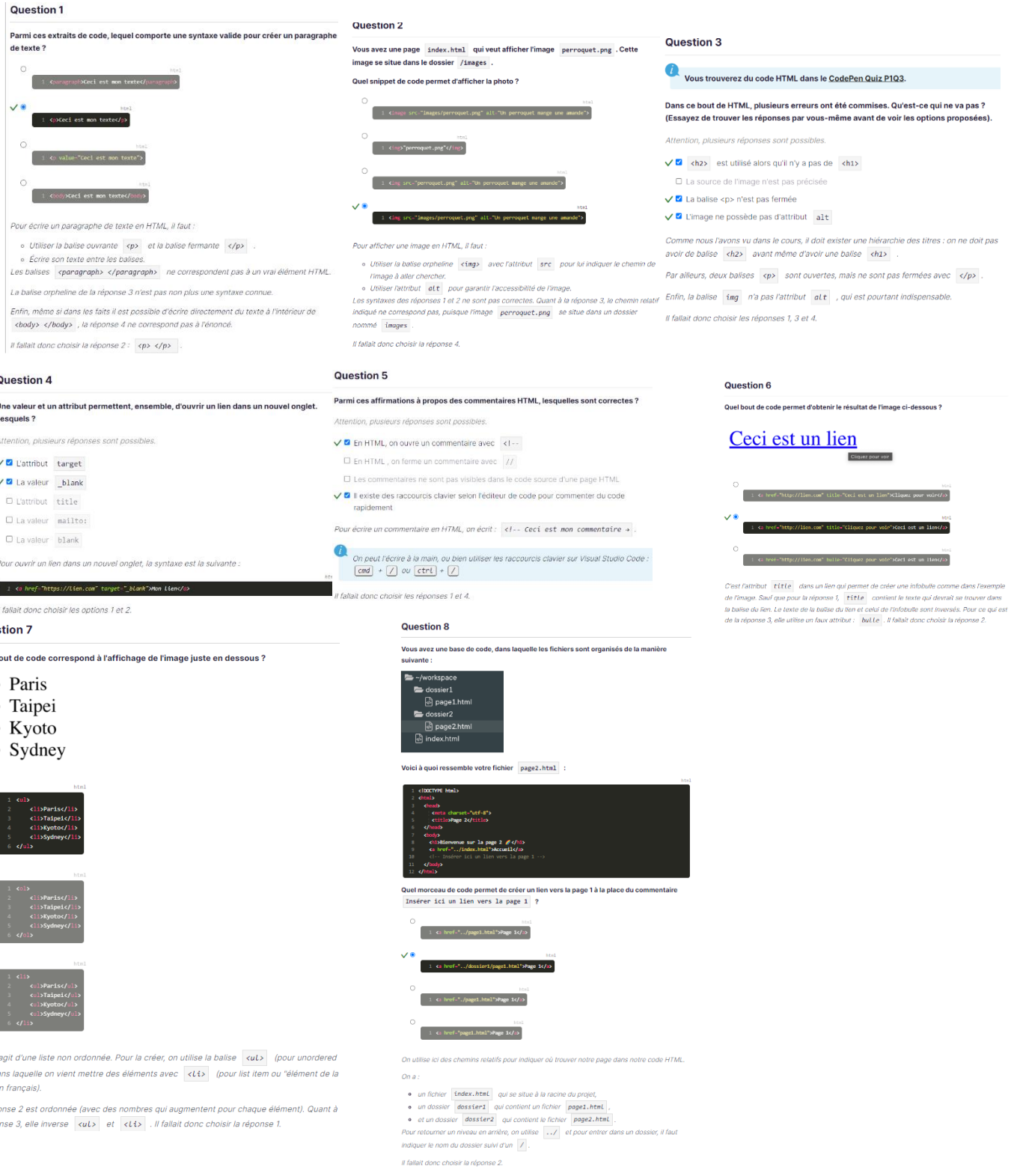
- A vous de jouer



- EN RÉSUMÉ

1. Il existe plusieurs formats d'images adaptés au Web : PNG, JPG…
2. On insère une image avec la balise <img> .
3. <img> doit obligatoirement comporter au moins ces deux attributs : src (source de l'image) et alt (courte description de l'image).
4. Il est possible d'afficher une autre version d'une image grâce à un lien qui entoure l'image.

QUIZZ : Maîtriser les bases de HTML5



Intégrez le CSS dans la page HTML

Liez le fichier CSS au fichier HTML

À partir du moment où on crée un fichier .css pour appliquer du sytle à notre page web (écrite dans un fichier .html)

Il faut lier les deux fichiers, pour ça il suffit d'ajouter une ligne dans le fichier .html

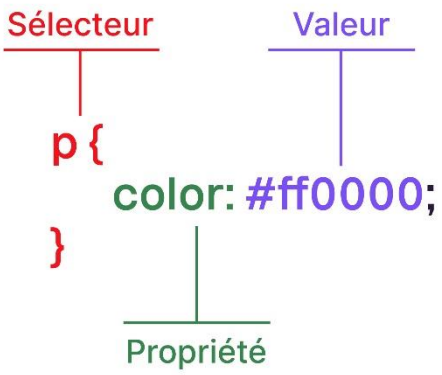
Cette ligne à rajouter dans le fichier .html s'ouvre avec la balise orpheline <link> et on la place à l'intérieur de la balise <head> </head> :

```html
<head>
    <meta charset="utf-8">
    <title>Ma page</title>
    <link href="style.css" rel="stylesheet">
</head>
```

Appliquez une propriété CSS à une balise HTML

h1{
color : blue ;
}

1. **Le sélecteur** : ici on écrit les noms des balises HTML dont on veut modifier l'apparence. Par exemple, si je veux modifier l'apparence de tous les paragraphes <p> , je dois écrire p (sans les chevrons).
2. **La ou les propriétés CSS** : les effets de style sont listés via des propriétés. Par exemple, color permet d'indiquer la couleur du texte, font-size permet d'indiquer la taille du texte, etc. Il existe BEAUCOUP de propriétés CSS ! Mais rassurez-vous, vous n'avez pas besoin de les connaître toutes par cœur.
3. **…et leurs valeurs** : pour chaque propriété CSS, on doit indiquer une valeur. Par exemple, pour la propriété color , il faut indiquer le nom de la couleur. Pour font-size, il faut indiquer quelle taille on veut, etc.



On écrit le nom de la balise. Puis entre accolades `{ }` on écrit les propriétés et leurs valeurs. (On peut mettre autant de propriétés que l'on veut à l'intérieur des accolades).

Chaque propriété est suivie du symbole `:` puis de la valeur correspondante.

Chaque ligne se termine par `;` .

Pour changer tous les paragraphes :

```
p {
   color: blue;
}
```

On peut faire des commentaires en CSS :

```
/*blablablabla
Blablalbla */
```

Appliquez une propriété CSS à plusieurs balises HTML à la fois

Nos titres niv 1 et nos paragraphes doivent être en bleu :

```
h1 {
   color: blue;
}

p {
   color: blue;
}
```

- ⇨ Répétitif

On peut combiner s'ils ont le même style:

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

Comment faire pour que certains paragraphes soient écrits d'une couleur différente ?

Pour appliquer un style à un seul élément (par exemple à un seul paragraphe parmi tous les paragraphes de notre code), on peut - **en théorie** - utiliser deux attributs :

1. l'attribut **class** ; => en pratique on utilise class
2. ou l'attribut **id**.

Les attributs class et id fonctionnent selon la même méthode mais on ne les utilise pas pour les mêmes raisons :

- pour appliquer un style à un seul élément parmi d'autres, on utilise un attribut class.
- pour appliquer un style à un élément unique en son genre, on utilise un attribut id.

En CSS, on peut appliquer du style à un élément (ou plus) avec l'attribut class. Par contre id ne peut s'utiliser que pour un seul élément, pas plus.

Appliquez un style à un élément isolé avec l'attribut : class

1. **Marquer un élément avec l'attribut class dans le fichier .html**
   Class est un attribut qu'on peut mettre dans sur n'importe quelle balise html
   Exemple :
   <h1 class="voici"> </h1>

   <img class="classes">

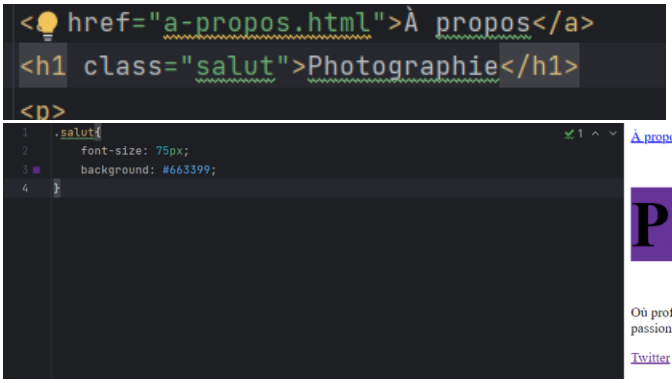
Oui, mais que met-on comme valeur à l'attribut **class**?

On doit écrire un nom qui sert à identifier la balise. C'est un moyen de sélectionner notre élément pour ensuite lui appliquer du style

```
<body>

  <h1>Titre principal</h1>

  <p>Ceci est le contenu de mon premier paragraphe</p>

  <p class="ma-classe">Ceci est le contenu de mon deuxième paragraphe</p>

  <h2>Voilà mon sous-titre h2</h2>

</body>
```

On vient de créer un attribut class nommé ma-classe dans le fichier .html pour marquer un élément auquel on veut appliquer un style en particulier. Maintenant, il faut appeler l'élément que l'on a marqué dans le fichier .css pour pouvoir lui appliquer un style.

2. **Appelez le nom de l'attribut class via un . dans le fichier .css**
   Dans le fichier .css, on va indiquer le nom de notre classe précédée d'un point « . »
   Exemple :
   .ma-classe {
   color: #663399;
   }
   Mon exemple :



   On peut utiliser la notation d'hexadécimal pour indiquer la couleur, pour être plus précis

   Appliquez une propriété CSS à *un élément unique* avec l'attribut id
   L'attribut id fonctionne comme la méthode classe mais ne peut être utilisé qu'une fois dans le code

Quel intérêt alors ?

En CSS, il n'y en a pas vraiment ; cela vous sera surtout utile si vous faites du **JavaScript** plus tard, pour reconnaître certaines balises.

En HTML, il y en a un car souvenez-vous on l'avait utilisé dans le chapitre sur les liens hypertextes pour créer des ancres.

En pratique donc, on n'utilisera en CSS un **id** que sur un élément qui est unique dans la page, comme le logo par exemple.

1. **Marquez un élément unique avec l'attribut id dans le fichier .html**
   On va marquer le logo grâce à l'attribut id dans le file .html
   <img src="images/logo.png" alt="Logo du site" id="logo">

2. **appelez l'attribut id via un # dans le fichier .css**
   Avec l'attribut **id** , on définit les propriétés de style de cet élément unique dans le fichier **.css** en l'appelant par un dièse **#**:
   Exemple :
   #logo {
   /* Indiquez les propriétés CSS ici */
   }

Appliquez plusieurs propriétés CSS d'un coup à un élément
En CSS il est possible de cumuler les styles

Font-size : permet d'indiquer la taille d'un texte

On peut ajouter plusieurs propriétés sur un même sélecteur :

```
balise1
{
   propriete1: valeur1;
   propriete2: valeur2;
/* … */
}
```

Exploitez les balises universelles
je souhaite modifier que 990 espèces d'oiseaux :
Il existe plus de 990 espèces d'oiseaux répertoriées dans toute l'Europe.

Il existe 2 balises universelles qui n'ont aucune signification particulière (elles n'indiquent pas que le mot est important, par exemple). Il y a une petite différence (mais significative) entre ces deux balises : :

-

  C'est une balise de type "**inline**", c'est-à-dire une balise que l'on place au sein d'un paragraphe de texte pour sélectionner certains mots uniquement. Les balises **** et **** sont de la même famille. Cette balise s'utilise donc au milieu d'un paragraphe, et c'est celle dont nous allons nous servir pour colorer "990 espèces d'oiseaux".

```
<span class="important">990 espèces d'oiseaux  </span>
```

-

  C'est une balise de type "**block**", qui entoure un bloc de texte. Les balises ** **, **<h1>** , etc., sont de la même famille. Ces balises ont quelque chose en commun : elles créent un nouveau "bloc", dans la page, et provoquent donc obligatoirement un retour à la ligne.** ** est une balise fréquemment utilisée dans la construction d'une mise en page, comme nous le verrons plus tard.

EN RÉSUMÉ

- CSS est un autre langage qui vient compléter le HTML. Son rôle est de mettre en forme votre page web.
- Pour écrire le code CSS, on crée un fichier séparé portant l'extension .css comme style.css.
- Pour lier les fichiers CSS et HTML, on rajoute une ligne dans la balise <head> </head> du fichier HTML : <link href="style.css" rel="stylesheet">
- En CSS, on sélectionne les portions de la page HTML qu'on veut modifier, et on change leur présentation avec des propriétés CSS :

```
balise1
{
   propriete1: valeur1;
   propriete2: valeur2;
}
```

- Il existe plusieurs façons de sélectionner la portion de page que l'on veut mettre en forme. Par exemple, on peut viser :
    - toutes les balises d'un même type, en écrivant simplement leur nom (h1par exemple) ;
    - certaines balises spécifiques, auxquelles on a donné des noms à l'aide des attributs class ouid(.nom-classe ou #nom-id) ;
    - uniquement les balises qui se trouvent à l'intérieur d'autres balises (h3,em).

Changez la taille du texte avec la propriété CSS font-size

Pour modifier la taille du texte, on utilise la propriété CSS **font-size** et ensuite, on indique :

- une **taille absolue** ;
- ou une **taille relative**.

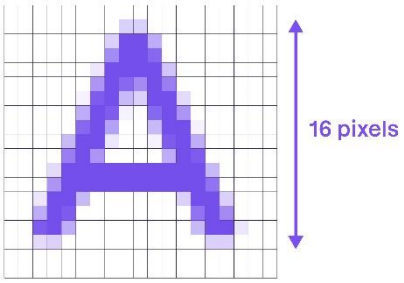
Indiquer une **taille absolue**, c'est très précis, mais il est conseillé de n'utiliser cette méthode que si on sait sur quelle taille d'écran ou dimension d'impression une personne verra le contenu de la page, car on risque d'indiquer une taille trop petite pour certains lecteurs.

Indiquer une **taille relative** a l'avantage d'être plus souple, elle s'adapte, selon les tailles d'écrans, plus facilement aux préférences des visiteurs.

Donnez une taille au texte avec une valeur absolue en px

Pour indiquer une taille absolue on utilise des pixels. Pour avoir un texte de 16 pixels on doit écrire :

font-size: 16px;



…ou optez pour une valeur relative en em (recommandé)

Il y a plusieurs moyens d'indiquer une valeur relative, une des plus fréquentes : em

- Le texte a une taille normale avec 1em .
- Le texte est plus grand avec une valeur supérieure à 1, comme 1.3em .
- Le texte est plus petit avec une valeur inférieure à 1, comme 0.8em.

/ ! \ Attention tout de même à la syntaxe : pour les nombres décimaux, il faut mettre un point et non une virgule. Vous devez donc écrire 1.4em et non pas 1,4em !

Exemple :

Si on le fait en HTML :

```
<p class="elem1">Élément 1 : 1em</p>
<p class="elem2">Élément 2 : 1.3em</p>
<p class="elem3">Élément 3 : 2em</p>
```

... et qu'on y associe le CSS ci-dessous :

```
.elem1 {
font-size: 1em;
}

.elem2 {
font-size: 1.3em;
}

.elem3 {
font-size: 2em;
}
```

... on obtient alors le résultat suivant :

Élément 1 : 1em

Élément 2 : 1.3em

Élément 3 : 2em

À quoi ça sert une taille relative ?

Pixel : une taille fixe

Em : une taille qui s'adapte à n'importe quelle écran

Choisissez une police avec la propriété CSS font-family

La propriété CSS qui permet d'indiquer la police à utiliser est **font-family** :

```
balise
{
   font-family: nom-police;
}
```

Liste de police sans sérif :

- Arial Black ;
- Futura ;
- Helvetica ;
- Impact ;
- Trebuchet MS ;
- Verdana.

Des caractères sans sérif est plus judicieux en termes d'accessibilité, car c'est plus facile à lire.

Liens pour police : google fonts

Intégrer une police :

1. Copiez les balises **<link>** dans le <head> </head> du fichier HTML.
2. Utilisez la propriété **font-family** dans le fichier CSS pour déclarer que vous voulez utiliser cette police.

On peut télécharger les polices

Mettez du texte en italique avec la propriété CSS font-style

En CSS, on donne une valeur à **font-style** pour dire si on veut que du texte soit en italique ou non :

- italic : le texte sera mis en italique ;
- normal : le texte sera normal (par défaut). Cela vous permet d'annuler une mise en italique. Par exemple, si vous voulez que les textes entre ne soient plus en italique, vous devrez écrire :
- exemple :

```
em
{
   font-style: normal;
}
```

Ceci est mon texte italique

Ceci est mon texte normal

Mettez du texte en gras avec la propriété CSS font-weight

Pour varier la graisse d'un texte, on utilise la propriété CSS **font-weight** à laquelle on associe une valeur, au choix :

1. bold : le texte sera en gras ;
2. normal : le texte sera écrit normalement (par défaut) ;
3. thin: le texte est plus fin.
4. Choisir la taille de l'épaisseur : 100 – 900

Thin (100)
Extra-Light (200)
Light (300)
Regular (400)
Medium (500)
SemiBold (600)
Bold (700)
ExtraBold / Heavy (800)
Black (900)

Pour appliquer les différents styles de texte (épaisseur et italique) pour les polices importées, il faut bien s'assurer d'avoir importé les styles de polices correspondants. Ainsi, pour utiliser la police Roboto en italique et bold, il faudra bien avoir importé dans votre code

Soulignez du texte avec la propriété CSS text-decoration

La propriété CSS **text-decoration** permet, entre autres, de souligner le texte, mais pas seulement. Voici quelques-unes des différentes valeurs qu'elle peut prendre :

- underline: souligné ;
- line-through: barré ;
- none: normal (par défaut, sauf dans le cas des liens).

Ceci est mon texte souligné

~~Ceci est mon texte barré~~

Ceci est mon texte normal

Alignez du texte avec la propriété CSS text-align

Le propriété CSS **text-align** permet d'aligner du texte selon la valeur qu'on lui donne :

- left : le texte sera aligné à gauche (c'est le réglage par défaut) ;
- center : le texte sera centré ;
- right : le texte sera aligné à droite ;

Texte aligné à gauche

Texte centré

Texte aligné à droite

- justify : le texte sera “justifié”.

  Justifier le texte permet de faire en sorte qu'il prenne toute la largeur possible sans laisser d'espace blanc à la fin des lignes. Les textes des journaux, par exemple, sont toujours justifiés.

L'alignement ne fonctionne que sur des balises de type block : <p>, , <h1>…

Aligner le texte d'une balise inline, comme , <a>,   ,   n'est pas possible. Et c'est logique quand on y pense : on ne peut pas modifier l'alignement de quelques mots au milieu d'un paragraphe.

EN RÉSUMÉ

- On modifie la taille du texte avec la propriété CSS font-size.
- On peut indiquer la taille en pixels, comme16px ; ou encore en “em”, comme 1.3em.
- On indique la police du texte avec la propriété CSS font-family .
- De nombreuses propriétés de mise en forme du texte existent : font-style pour l'italique, font-weight pour la mise en gras, text-decoration pour le soulignement.
- Le texte peut être aligné avec la propriété CSS text-align .

Ajoutez de la couleur et un fond

La propriété qui permet de modifier la couleur du texte : color

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

Différentes formes :

- Hexadécimal : (6 chiffres précédés d'un #). Exemple : color: #FFC8D3
- Notation RGB : color: rgb(250,25,118)
- Rgba : ajouter la notion d'opacité (transparence) : color: rgba(250,25,118, 0.5)

Sites internet : coolors ; html color picker

Appliquez une couleur d'arrière-plan avec background-color

Proprité css : background-color

Elle s'utilise de la même manière que : color

Càd : écrire nom de la couleur, hexadécimal, RGB

Il faut l'appliquer à la balise <body> car correspond à l'ensemble de la page

Ex :

```
body {
   background-color: black; /* Le fond de la page sera noir */
   color: white; /* Le texte de la page sera blanc */
}
```

On a demandé à ce que le texte de la balise <body> soit écrit en blanc, et tous les paragraphes <p> et titres <h1> ont pris cette couleur. Comment ça se fait ?

La balise `<body>` contient, entre autres, les balises de paragraphe `<p>` et de titre `<h1>`. Si on applique à la balise `<body>` une couleur de fond noire et une couleur de texte blanche, tous les titres et paragraphes seront eux aussi en blanc sur fond noir.

On appelle ça : l'héritage

Donc ma page web sera écrit en blanc ?

Non, car par la suite on peut modifier :

```
body {
   background-color: black; /* Le fond de la page sera noir */
   color: white; /* Le texte de la page sera blanc */
}
h1 {
   color: purple;
}
```

Ajoutez une image de fond avec background-image

Une "image de fond" ne s'applique pas forcément à la page entière. On peut aussi mettre une image de fond derrière des titres, ou des paragraphes uniquement.

Appliquez une image de fond derrière un élément HTML

Propriété permettant d'indiquer une image de fond : background-image

- En absolu : http://.....
- En relatif : fond.png
    - Attention lorsque vous écrivez une adresse en relatif dans le fichier CSS : l'adresse de l'image doit être indiquée *par rapport au fichier .css* et non pas par rapport au fichier .html . Pour simplifier les choses, je vous conseille de placer l'image de fond dans le même dossier que le fichier .css (ou dans un sous-dossier).

HTML :

```html
<body>
    <div class="banniere">
        <h1>Mon blog</h1>
    </div>
    <p>Ceci est le contenu de mon premier paragraphe</p>
    <p>Ceci est le contenu de mon deuxième paragraphe</p>
</body>
```

CSS :

```css
body {
    font-family: sans-serif;
    margin: 0; /* Vous ne connaissez pas encore cette propriété mais elle permet de s'assurer que nos éléments prennent bien toute la largeur de la page */
}

.banniere {
    padding: 100px; /* Vous ne connaissez pas encore cette propriété mais elle permet d'ajouter de l'espace dans notre élément*/
    background-image: url('paysage.jpg');
}

h1 {
    color: white;
    text-align: center;
}
```

Modifiez le comportement d'une image de fond

Pour changer le comportement d'une image de fond, il existe plusieurs propriétés CSS :

1. La propriété CSS **background-attachment** associée à la valeur fixed permet de rendre l'image de fond fixe lorsqu'on déroule la page web : background-attachment: fixed;
2. La propriété CSS **background-size** associée à la valeur cover permet de redimensionner l'image afin qu'elle s'adapte à la taille de l'élément qui la contient (elle garde ses proportions, en rognant la largeur ou la hauteur en fonction de la taille de l'élément qui la contient) : background-size: cover;
3. La propriété CSS **background-position** associée aux valeurs top , bottom , left , center ou right permet d'indiquer où doit se trouver l'image de fond, par exemple : background-position: top right;

Combinez ces propriétés CSS avec la "super-propriété" background

Si vous utilisez beaucoup de propriétés en rapport avec le fond, vous pouvez utiliser une sorte de “super-propriété” appelée background dont la valeur peut combiner plusieurs des propriétés :

- background-image
- background-repeat
- background-attachment
- background-size
- background-position

C'est la première “super-propriété” que je vous montre, il y en aura d'autres.

On peut donc tout simplement écrire :

```css
.banniere
{
    background: url("paysage.jpg") cover center;
}
```

Créez des dégradés avec linear-gradient

Pour créer un dégrader propriété CSS => background

background: linear-gradient(90deg, #8360c3, #2ebf91); => en fr = "J'applique un dégradé linéaire, à 90 degrés, en partant de la couleur #8360c3 pour arriver à la couleur #2ebf91.

Ici, on utilise **linear-gradient** mais il existe d'autres manières de créer des dégradés. Si vous voulez de l'inspiration en termes de dégradés, rendez-vous sur **UI Gradients**, et si vous voulez construire vos propres dégradés, je vous conseille **CSS Gradient**.

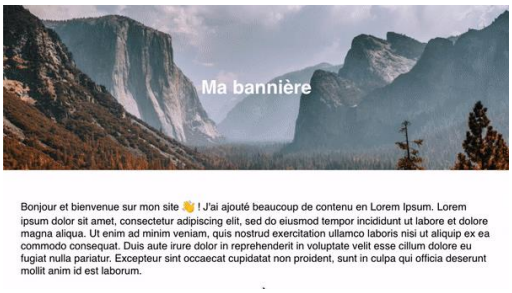
Jouez sur la transparence avec la propriété CSS opacity

La propriété CSS opacity permet d'indiquer le niveau d'opacité (c'est l'inverse de la transparence).

- Avec une valeur de 1, l'élément sera totalement opaque : c'est le comportement par défaut.
- Avec une valeur de 0, l'élément sera totalement transparent.

Il faut donc choisir une valeur comprise entre 0 et 1. Ainsi, avec une valeur de 0.6 , votre élément sera opaque à 60 %… et on verra donc à travers !

Jetez un œil à ce GIF auquel on applique 0.6 , puis 0.4 et enfin 0.2 :



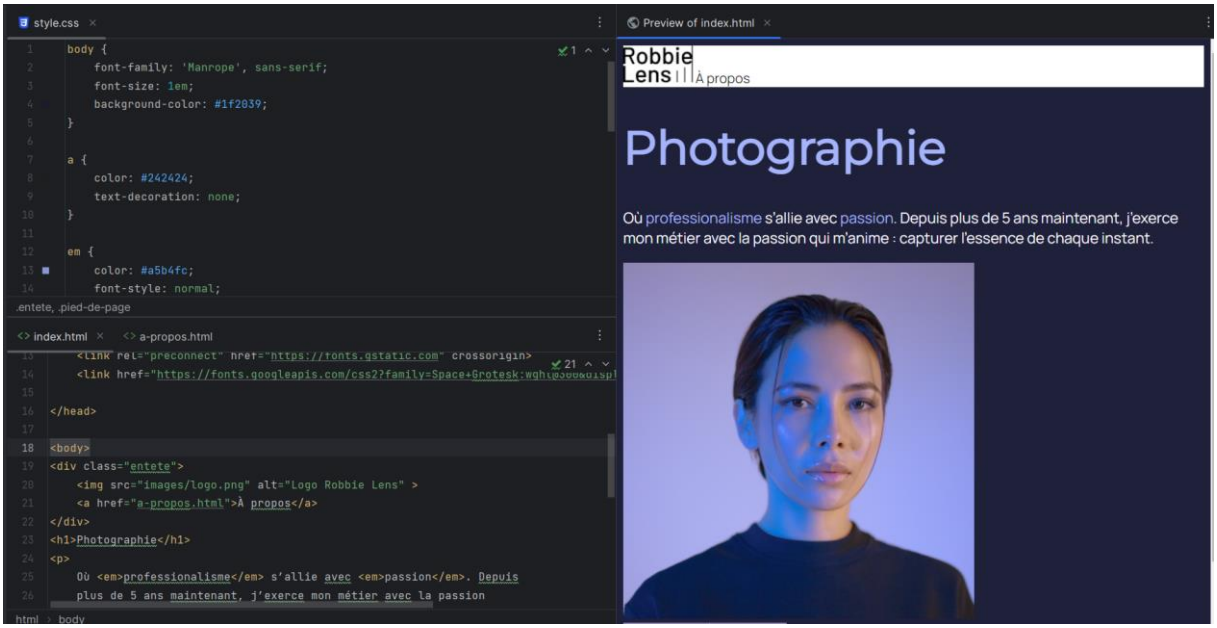
Comme vous pouvez le voir, si vous appliquez la propriété opacity à un élément de la page, tout le contenu de cet élément sera rendu transparent (même les images, les autres blocs à l'intérieur, etc.). Si vous voulez juste rendre la couleur de fond transparente, utilisez plutôt la notation RGBA que vous avez vue dans la vidéo d'introduction du chapitre.

À vous de jouer !

- changer les couleurs de fond pour que :
    - la div dans laquelle on a les liens en haut des pages, et le pied de page, aient un fond blanc,
    - le cœur de la page ait un fond correspondant à #1F2039 ;
- … et les couleurs de texte pour que :
    - les liens en tête et pied de page soient de couleur #242424 ,
    - les paragraphes, les listes et les titres H2 soient de la couleur #F9F8FF (sur fond bleu foncé).

Vous trouverez également des liens hypertextes qui ont été ajoutés : "Un projet ? Écrivez-moi" et "Voir mon portfolio". Vous devrez leur ajouter un dégradé qui passe de la couleur #8E86B5 à la couleur #ACAEED et les mettre dans la police Montserrat, en blanc.

C'est normal si votre couleur de fond a une petite marge, nous corrigerons cela dans quelques chapitres.



EN RÉSUMÉ :

- On change la couleur du texte avec la propriété color et la couleur de fond avec la propriété background-color.
- On peut indiquer une couleur en écrivant son nom en anglais, black par exemple, sous forme hexadécimale, comme #FFC8D3, ou en notation RGB, comme rgb(250,25,118).
- On peut ajouter une image de fond avec la propriété background-image. On peut choisir de fixer l'image de fond, ou encore de la positionner où on veut sur la page.
- On peut rendre une portion de la page transparente avec la propriété opacity ou avec la notation RGBA (une extension de la notation RGB, où la quatrième valeur indique le niveau de transparence).

Créez des bordures avec la propriété CSS border

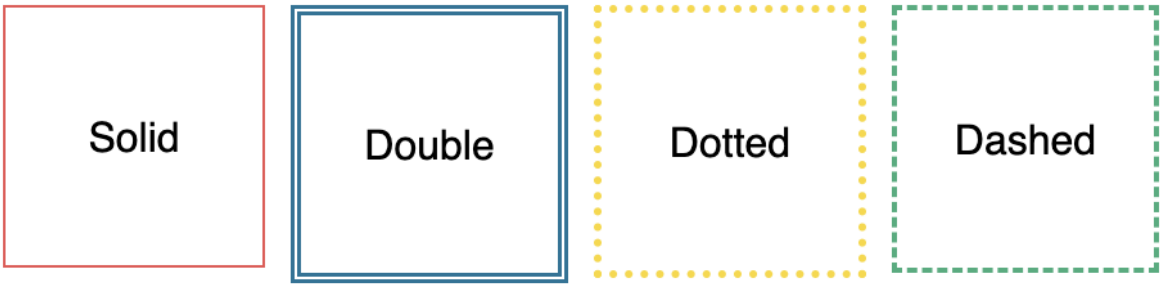
Le css offre un large choix de bordures :

- Border-witdth
- Border-color
- Border-style
- Etc

Super-propriété : border => même chose que background

Avec **border**, on peut utiliser jusqu'à trois valeurs pour modifier l'apparence de la bordure :

1. **La largeur** que l'on définit avec une valeur en pixels (comme 2px).
2. **La couleur** que l'on indique avec un nom de couleur, une valeur hexadécimale, ou une valeur RGB.
3. **Le type de bordure** qui peut être solid (un trait simple), double (un double trait), dotted (un trait en pointillés), dashed (un trait en tirets), ou autre. Vous avez un large panel d'options

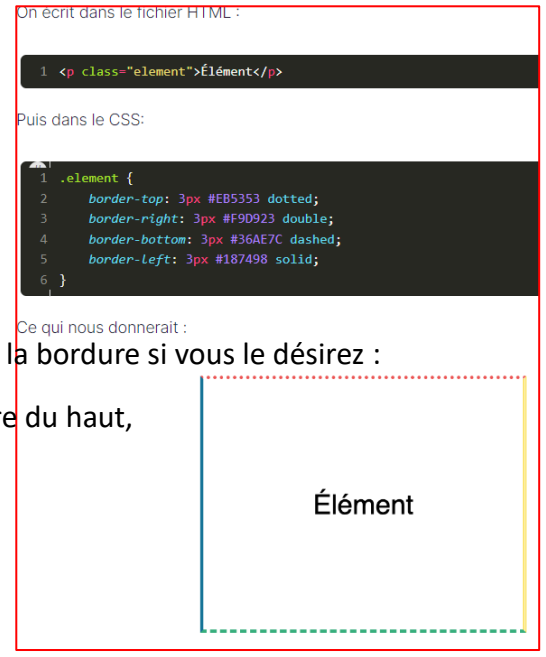


On peut choisir les côtés qu'on souhaite modifier :

1. border-top : bordure du haut
2. border-bottom : bordure du bas
3. border-left : bordure de gauche
4. border-right : bordure de droite

Il existe aussi des équivalents pour paramétrer chaque détail de la bordure si vous le désirez :

- border-top-width pour modifier l'épaisseur de la bordure du haut,
- border-top-color pour la couleur du haut, etc.



On peut avoir une bordure que d’un seul côté :

```css
.element {
    font-size: 25px;
    background-color: skyblue;
    border-left: 5px #187498 solid;
    padding: 100px;
}
```

Ce qui donne :

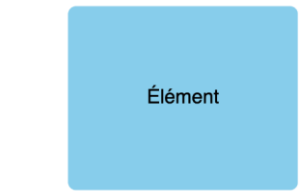
L'élément a une bordure à gauche

Arrondissez vos angles avec border-radius

La propriété CSS **border-radius** permet d'arrondir les angles de n'importe quel élément

```css
.element {
    font-size: 25px;
    background-color: skyblue;
    border-radius: 10px;
    padding: 100px;
}
```

Ce qui donne :



L'élément a des coins arrondis

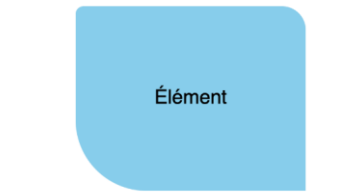
Comme pour les bordures, on peut avoir des angles différents

Border-radius :

1. En haut à gauche.
2. En haut à droite.
3. En bas à droite.
4. En bas à gauche.

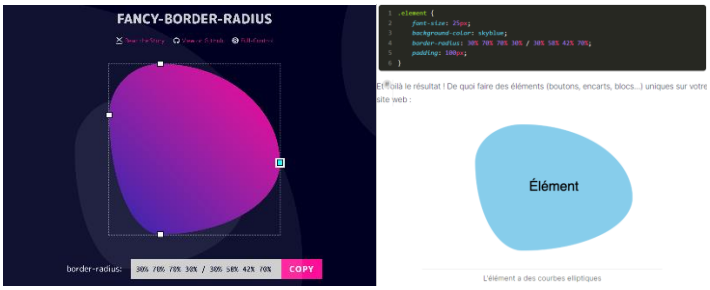
```css
.element {
    font-size: 25px;
    background-color: skyblue;
    border-radius: 10px 30px 0px 90px;
    padding: 100px;
}
```

Ce qui donne :



L'élément a des radius de différentes valeurs pour chaque angle

Il est même possible de créer des courbes elliptiques avec border-radius .Pour cela, il faut indiquer deux valeurs séparées par une barre oblique (*slash*) : /. Mais le rendu est assez dur à anticiper. Il vaut mieux utiliser des outils de visualisation comme [Fancy Border Radius] du rendu final :



Ajoutez une ombre portée avec la propriété CSS box-shadow

Propriété CSS : box-shadow

1. Le décalage horizontal de l'ombre.
2. Le décalage vertical de l'ombre.
3. L'adoucissement du dégradé.
4. La couleur de l'ombre.

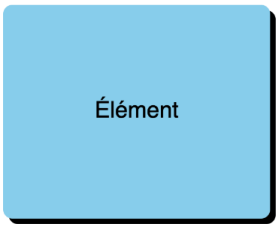
L'adoucissement peut être :

- **faible** (si on lui donne une **valeur inférieure** à celle du décalage),
- **normal** (si on lui donne une **valeur égale** à celle du décalage)
- ou **élevé** (si on lui donne une **valeur supérieure** à celle du décalage

Par exemple :

```
.element {
    font-size: 25px;
    background-color: skyblue;
    border-radius: 10px;
    box-shadow: 6px 6px 0px rgba(0, 0, 0);
    padding: 100px;
```

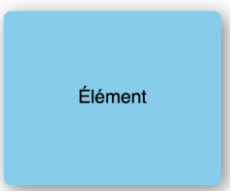
Ce qui donne :



Box-shadow :

```css
.element {
    font-size: 25px;
    background-color: skyblue;
    border-radius: 10px;
    box-shadow: 6px 6px 25px rgba(0, 0, 0, 0.5);
    padding: 100px;
}
```

Ce qui donne :



Il n'est pas toujours simple de trouver l'effet d'ombre désiré à tâtons. Pour vous aider, vous pouvez :

- utiliser l'outil **Shadows Brumm** ;
- vous inspirer des exemples listés sur **CSS Scan**.

Ajoutez une ombre à un texte avec text-shadow

Text-shadow permet d'ajouter une ombre sur les lettres

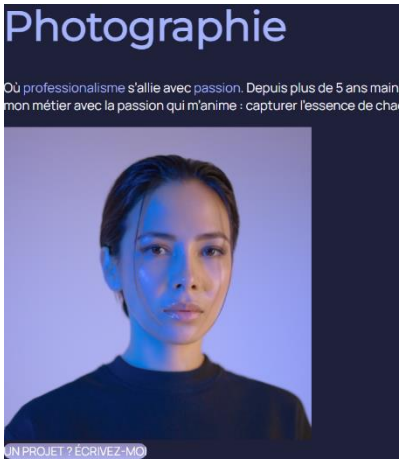
Les valeurs fonctionnent exactement de la même façon que box-shadow :

1. décalage horizontal,
2. décalage vertical,
3. adoucissement,
4. et couleur.

```css
h1 {
    font-size: 50px;
    text-shadow: 3px 3px 0px rgba(0,0,0,0.2);
    padding: 100px;
}
```

Et voyez ce que ça nous donne :

A vous de jouer ! :



EN RÉSUMÉ :

- On peut appliquer une bordure à un élément avec la super-propriété CSS border. Il faut indiquer la largeur de la bordure, sa couleur et son type (simple, double, pointillés, tirets).
- On peut arrondir les bordures avec la propriété CSS border-radius.
- On peut ajouter une ombre aux blocs de texte avec box-shadow. On doit indiquer le décalage vertical et horizontal de l'ombre, son niveau d'adoucissement et sa couleur.
- Le texte peut lui aussi avoir une ombre avec text-shadow.

Créez des apparences dynamiques

Les **pseudo-classes** viennent s'ajouter à un sélecteur CSS pour décrire à quel moment un style en particulier doit s'appliquer.

Stylisez un élément au survol de la souris avec :hover

La pseudo classe « :hover » s'applique lorsque le user survole un élément avec sa souris

Comment écrire :

a:hover {

// Insérer ici les propriétés CSS à appliquer

}

Balise a pour lien hypertexte mais si ma classe s'appelle ma-classe on écrirait .ma-classe :hover

```html
<body>
    <div class="mon-element">
    <h1>Ma page de pseudo-classes</h1>
    <p>Les pseudo-classes sont très importantes afin de créer des apparences dynamiques en fonction des interactions avec les utilisateurs.</p>
    </div>
</body>
```

..et le code suivant en CSS :

```css
.mon-element:hover {
    background-color: #BEAEE2;
    color: white;
}

h1:hover {
    color: #CDF0EA;
}

p:hover {
    color: #F9CEEE;
}
```

.. ce qui nous donne :

Stylisez un élément au moment du clic avec :active

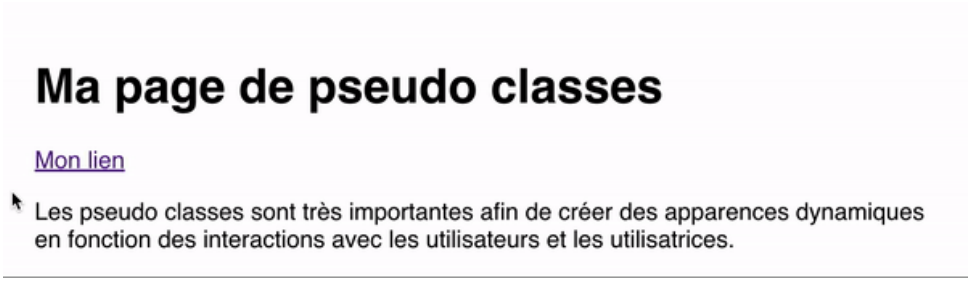
La pseudo-classe :active permet d'appliquer un style particulier au moment du clic

> En pratique, elle n'est utilisée que sur les liens hypertexte car le style ne se maintient que très peu de temps : en fait, le changement intervient lorsque le bouton de la souris est enfoncé. Ce n'est donc pas forcément toujours bien visible.

a:active {

background-color: #C1FFD7;

}



Stylisez un élément sélectionné par le visiteur avec :focus

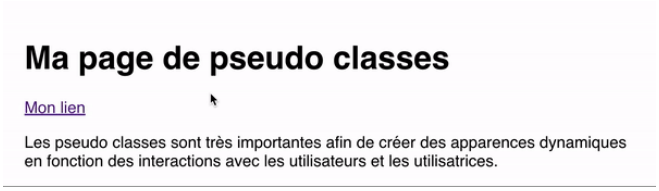
Là, c'est un peu différent. La pseudo-classe `:focus` applique un style lorsque l'élément est sélectionné.

C'est-à-dire ?

On peut notamment utiliser la touche "tab" ⇥ du clavier pour déclencher le focus sur chacun des éléments.

```
a:focus {
    background-color: #FCFFA6;
}
```

Ce qui nous permet d'obtenir, avec la sélection depuis la touche "tab" du clavier : ⇥



Stylisez un lien hypertexte déjà consulté avec :visited

Possibilité d'appliquer un style sur un lien hypertexte déjà été cliqué

Si vous ne souhaitez pas que les liens déjà visités soient colorés d'une façon différente qu'un lien hypertexte non cliqués, il vous faudra leur appliquer la même couleur qu'aux liens normaux. De nombreux sites web font cela (OpenClassrooms y compris !).

Exemple CSS :

```
a:visited {
    color: black;
}
```

> Les navigateurs ne connaissent pas toutes les propriétés CSS qui existent. En fait plus le navigateur est vieux, moins il connaît de fonctionnalités CSS.

Allez plus loin avec les sélecteurs avancés

En CSS, le plus dur est bien souvent de réussir à cibler l'élément dont on veut changer le style.

Le sélecteur universel *

Le sélecteur universel * sélectionne toutes les balises sans exception.

* {

/* Insérez ici votre style */

}

Le sélecteur d'une balise contenue dans une autre : A B

Prenons un exemple avec ce code écrit dans le fichier HTML :

```html
<h3>Titre avec <em>texte important</em></h3>
```

Dans CSS, on écrirait alors :

```css
h3 em {
/* Insérez ici votre style */
}
```

Ce morceau de code signifie en français :

> "Applique ce style à toutes les balises `<em>` situées à l'intérieur d'une balise `<h3>` ".

Notez qu'il n'y a pas de virgule entre les deux noms de balises.

Le sélecteur d'une balise qui en suit une autre : A + B

h3 + p {

/* Insérez ici votre style */

}

Ce qui aura pour résultat de sélectionner la première balise située après un titre <h3>.

Exemple :

<h3>Titre</h3>

Paragraphe

Le sélecteur d'une balise qui possède un attribut : a[attribut]

```
a[title] {
/* Insérez ici votre style */
}
```

En français, ce morceau de code signifie :

> "Sélectionne tous les liens hypertexte `<a>` qui possèdent un attribut `title` ".

Exemple de code HTML associé :

```html
<a href="http://site.com" title="Infobulle">
```

Il existe des variantes de cette forme de sélecteur :

- `a[attribut="Valeur"]` : une balise qui possède un attribut et une valeur exacte, comme :

```css
a[title="Cliquez ici"] {
/* Insérez ici votre style */
}
```

C'est la même chose, mais l'attribut doit en plus avoir exactement pour valeur “Cliquez ici”.

Exemple de code HTML associé :

```html
<a href="http://site.com" title="Cliquez ici">
```

- `a[attribut*="Valeur"]` : une balise, un attribut et une valeur, comme :

```css
a[title*="ici"] {
/* Insérez ici votre style */
}
```

Idem, l'attribut doit cette fois contenir dans sa valeur le mot “ici” (peu importe sa position).

Exemple de code HTML associé :

```html
<a href="http://site.com" title="Quelque part par ici">
```

Il en existe beaucoup d'autres sélecteurs. Si vous voulez une liste complète, vous pouvez vous renseigner directement à la source : sur le **site du W3C** !

EN RÉSUMÉ

- En CSS, on peut modifier l'apparence de certaines sections dynamiquement, après le chargement de la page, lorsque certaines interactions se produisent. On utilise pour cela les pseudo-classes.
- La pseudo-classe :hover modifie l'apparence d'un élément au survol (par exemple : a:hover modifie l'apparence des liens hypertextes lorsque la souris pointe dessus).
- La pseudo-classe :active modifie l'apparence des liens hypertextes au moment du clic.
- La pseudo-classe :visited modifie l'apparence des liens hypertextes lorsqu'un lien a déjà été visité.
- La pseudo-classe :focus modifie l'apparence d'un élément sélectionné via la touche "tab".
- Encore aujourd'hui, certaines propriétés ne sont pas totalement reconnues par tous les navigateurs.

QUIZZ partie 2

Structurez votre page

Utilisez la balise <header> pour l'en-tête

La plupart des sites web possèdent un **<header>**(en-tête, en français). On y trouve le plus souvent, à minima, un logo. On le place, pour des raisons de cohérence, en tête du code, donc au début de la balise <body> :

<header>

<!-- Placez ici le contenu de l'en-tête de votre page -->

</header>

Exemple d'entête :

Attention, la balise **<header>** n'a rien à voir avec la balise <head> que vous aviez vue dans la première partie :

- la balise **<header>** permet de créer une partie de la page HTML à créer ;
- la balise `<head>` permet d'inclure des informations générales sur la page, notamment son titre, des liens vers des polices, vers une feuille de style, etc.

Il peut y avoir plusieurs en-têtes dans votre page. Si celle-ci est découpée en plusieurs sections, chaque section peut en effet avoir son propre <header> .

Utilisez la balise <footer> pour le pied de page

À l'inverse de l'en-tête, le pied de page se trouve tout en bas de la page. On y trouve généralement des informations comme des liens de contact, les mentions légales, la politique de confidentialité, etc.

<footer>

<!-- Placez ici le contenu du pied de page -->

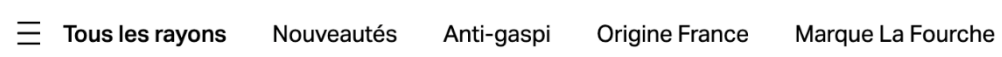
</footer>

Exemple de pied de page :

Utilisez la balise <nav> pour le menu de navigation

La balise <nav> doit regrouper tous les principaux liens de navigation du site. Vous y placerez par exemple le menu principal de votre site. Généralement, le menu est réalisé sous forme de liste à l'intérieur de la balise<nav> .

Exemple :




```
<nav>
  <ul>
    <li>
      <a href="#">Nouveautés</a>
    </li>
    <li>
      <a href="#">Anti-gaspi</a>
    </li>
    <li>
      <a href="#">Origine France</a>
    </li>
    <li>
      <a href="#">Marque La Fourche</a>
    </li>
  </ul>
</nav>
```

Utilisez la balise <main> pour le contenu principal de la page

La balise <main> permet de déclarer le contenu principal de votre page. Elle englobe la majeure partie du <body> d'une page, en excluant les autres éléments tels que le <header>, le <footer> et de potentiels <aside>. La balise <main> doit être unique : il n'y en a qu'une seule par page :

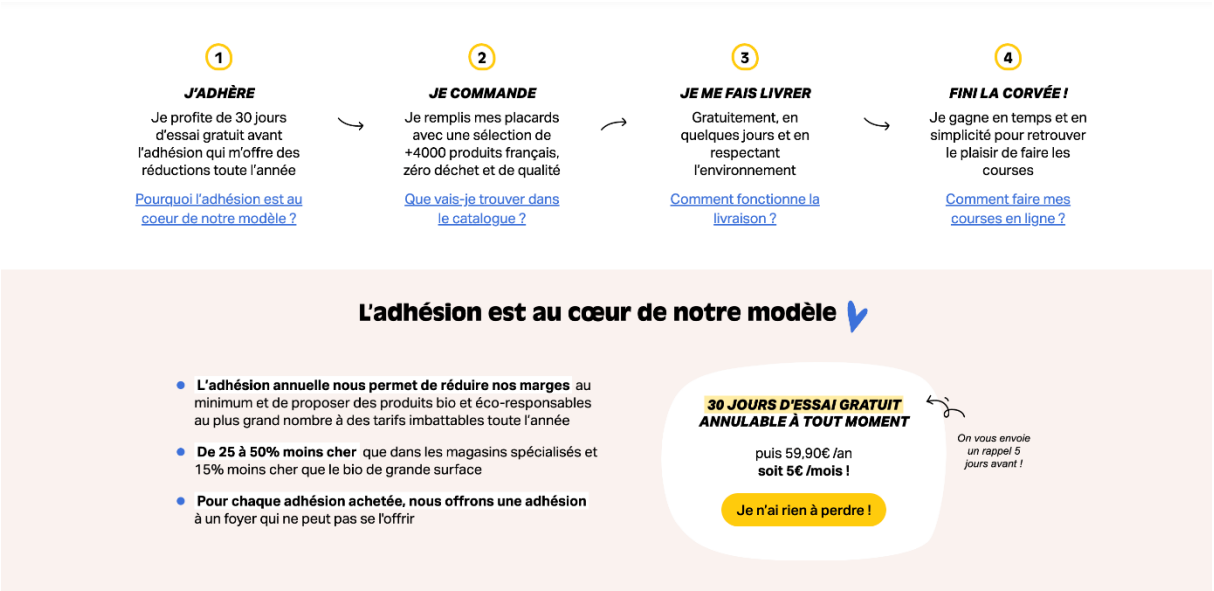
<main></main>

Utilisez des balises <section> pour structurer le contenu du <main>

La balise <section> sert à regrouper des contenus en fonction de leur thématique. Elle englobe généralement une portion du contenu au centre de la page.

<section></section>

Exemple :



Utilisez une balise <aside> pour des contenus additionnels dans le main

Sert à structurer la page

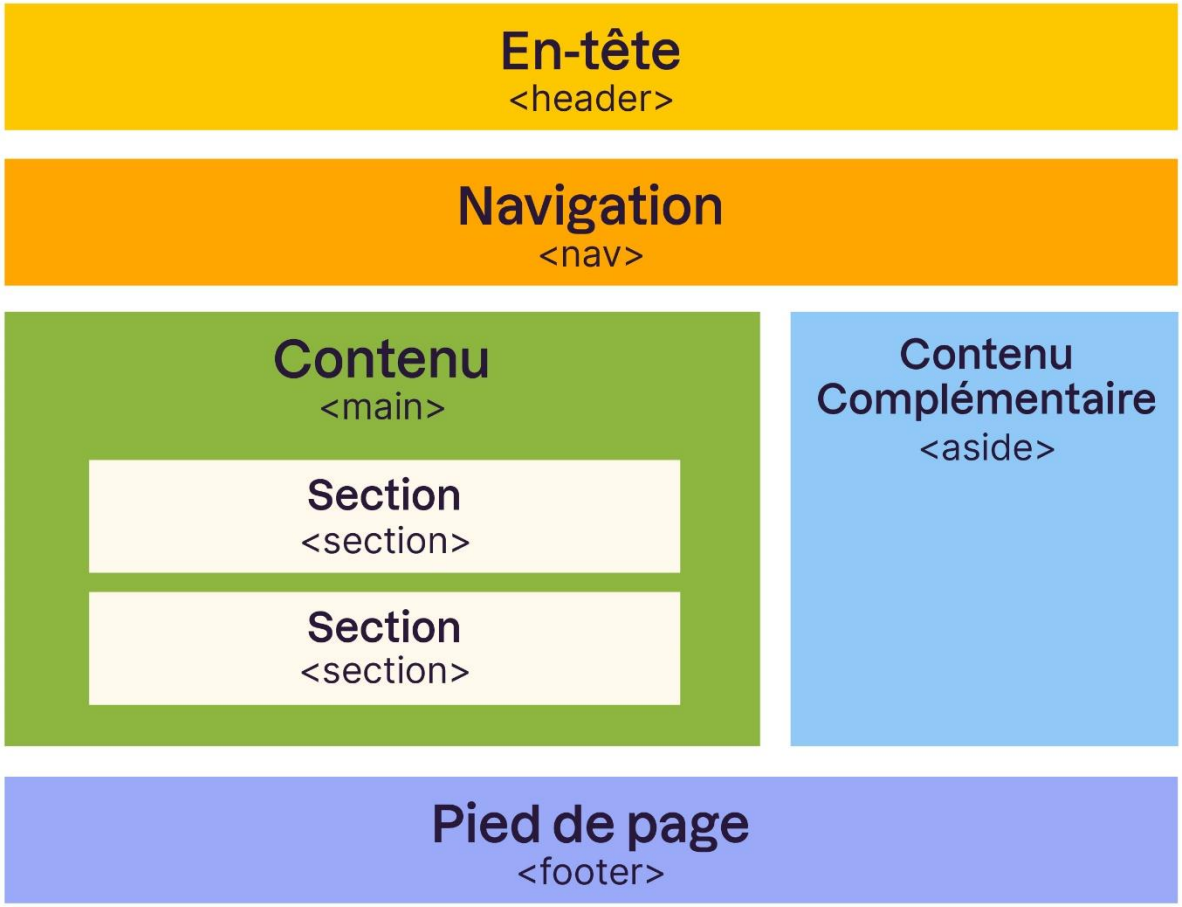
Exemple :permet de contenir des infos complémentaires au document qu'on visualise.

On peut avoir plusieurs blocs <aside>

Exemple : Sur Wikipédia, par exemple, il est courant de voir à droite un bloc d'informations complémentaires à l'article que l'on visualise.

À noter : la balise `<article>` sert à englober une portion généralement autonome de la page. C'est une partie de la page qui pourrait ainsi être reprise sur un autre site. C'est le cas par exemple des actualités (articles de journaux ou de blogs).

Pour résumer :



C’est un exemple d’organisation, il est tout à fait possible que le menu de navigation soit à droite, en haut, etc

Notre site peut fonctionner sans ces balises, mais c’est pour aider l’ordinateur :

Ceci est l’entête

Ceci est le pied de page, etc

EN RÉSUMÉ :

Plusieurs balises permettent de délimiter les différentes zones qui constituent la page web :

- <header> : en-tête ;
- <footer> : pied de page ;
- <nav> : liens principaux de navigation ;
- <section> : section de page ;
- <aside> : informations complémentaires ;
- <article> : article indépendant.

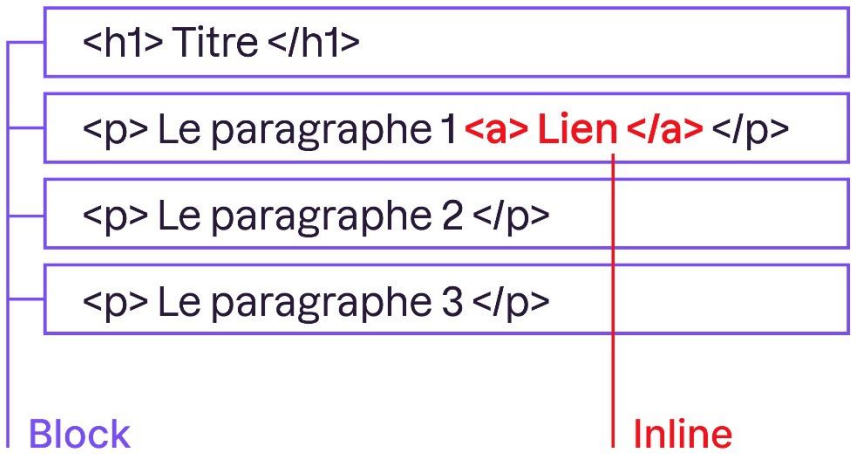
Ces balises peuvent être imbriquées les unes dans les autres. Ainsi, une section peut avoir son propre en-tête.

Ces balises ne s'occupent pas de la mise en page. Elles servent seulement à indiquer à l'ordinateur la fonction du texte qu'elles encadrent. On pourrait très bien placer l'en-tête en bas de la page, si on le souhaitait.

Découvrez le modèle des boîtes

- Block : ce type de balise crée automatiquement un retour à la ligne avant et après
- Inline : ce type se trouve obligatoirement à l'intérieur d'une balise block

Schéma :



Les blocs sont les uns en dessous des autres.

On peut aussi les imbriquer les uns à l'intérieur des autres. Le bloc main contient des blocs sections qui eux-mêmes peuvent contenir des div.

Quant à la balise a qui est de type inline se trouve à l'intérieur d'une balise block et le texte vient s'insérer sur la même ligne.

Faites bon usage des balises universelles et

Il existe deux balises génériques et, comme par hasard, la seule différence entre les deux est que l'une d'elles est inline et l'autre est block :

1. La balise****(qui est de type **inline**).
2. La balise** **(qui est de type **block**).

N'abusez pas des balises universelles, même si elles sont utiles dans certains cas : beaucoup de développeurs mettent des `<div>` et des `<span>` trop souvent, et oublient que d'autres balises plus adaptées existent.

Par exemple :

- `<span class="important">` : ici, il ne s'agit pas d'une bonne pratique, alors qu'il existe les balises `<strong>` ou `<em>` qui servent à indiquer l'importance !
- `<div class="titre">` : idem, cela ne convient pas, puisqu'il existe des balises faites spécialement pour les titres (`<h1>` , `<h2>` ...).

Dimensionnez les éléments avec width et height

Contrairement à un inline , un block peut avoir une largeur et une hauteur précises grâce à ces deux propriétés CSS :

1. width (**largeur** du bloc).
2. height (**hauteur** du bloc).

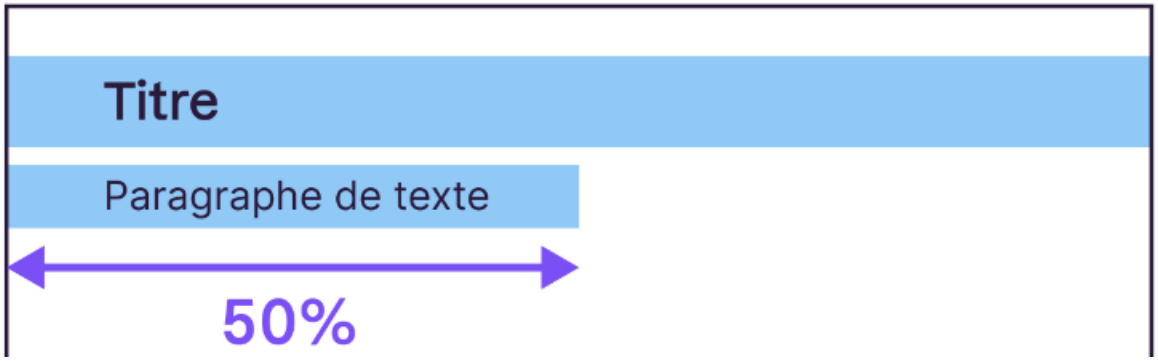
On les exprime en pixels px ou en pourcentage %.

Par défaut un bloc prend 100& de la largeur disponible

Maintenant, rajoutons un peu de CSS afin de modifier la largeur des paragraphes. Si on veut que tous les paragraphes aient une largeur de 50 %, on écrira en CSS :

```css
p {
    width: 50%;
}
```

Le résultat est visible juste en dessous :

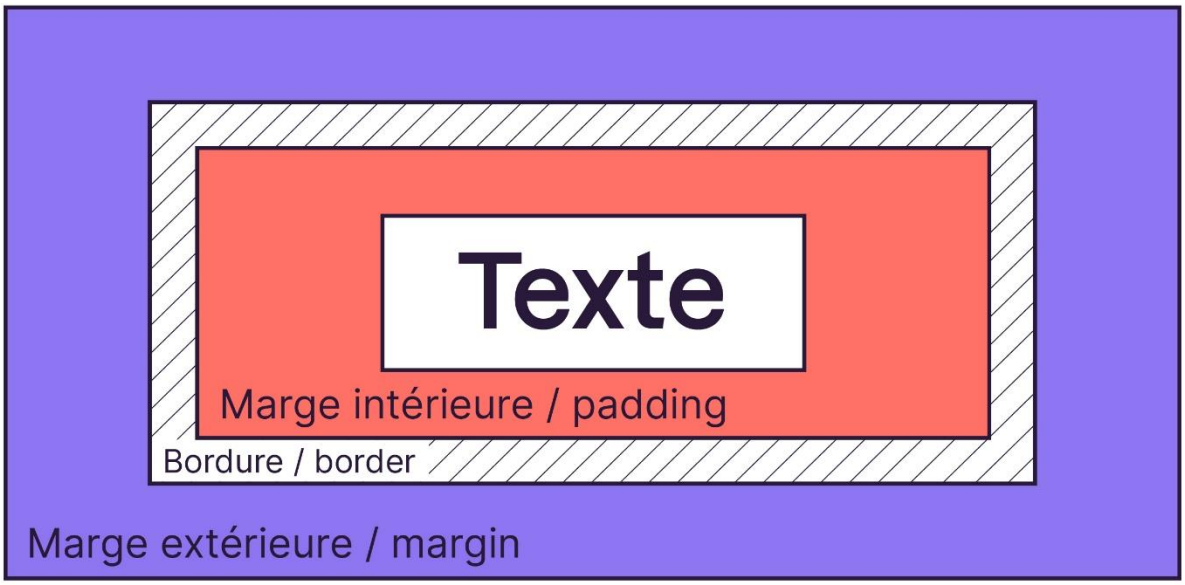


Les pourcentages sont utiles pour créer un design qui s'adapte automatiquement à la résolution d'écran du visiteur.

Les images sont un cas un peu particulier : elles se mettent les unes à la suite des autres, sans prendre toute la largeur, mais on peut malgré tout leur donner une dimension exacte.

Définissez des marges avec margin et padding

Tous les blocs possèdent des marges (intérieures et extérieures) :



En CSS, on peut modifier la taille des marges avec deux propriétés :

1. **margin** (taille de la **marge extérieure)**
2. **padding** (taille de la **marge intérieure**).

Par défaut il n'y pas de padding mais il y a un margin

Ajoutez une marge intérieure avec la propriété CSS padding

Pour ajouteraux paragraphes une marge intérieure (**padding**) de 16px par exemple, on écrit :

```css
p {
    width: 350px;
    background-color: #F1C864;
    text-align: justify;
    padding: 16px;
}
```

Avec padding

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

sans padding

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Ajoutez une marge extérieure avec la propriété CSS margin

On veut que nos paragraphes soient plus espacés entre eux

```css
p {
    width: 350px;
    background-color: #F1C864;
    text-align: justify;
    padding: 16px;
    margin: 50px;
}
```

Résultat :

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

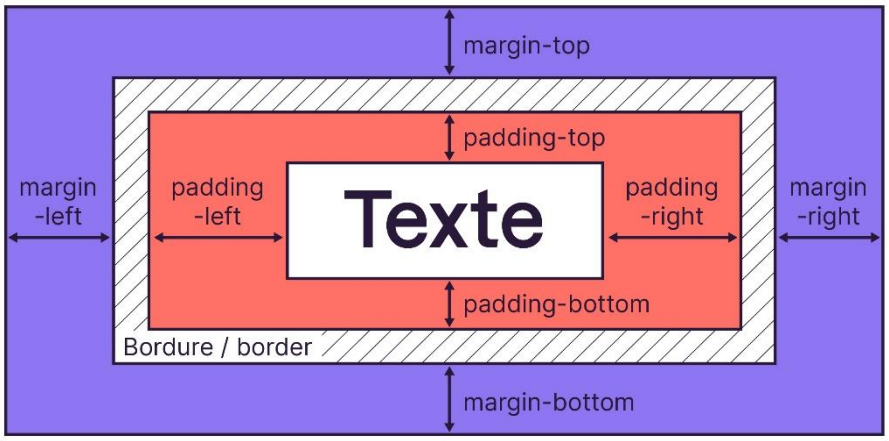
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorsqu'on fait un margin et un padding (ça s'applique

Aux quatre côtés du bloc)

Spécifiez les propriétés margin et padding

- **top**: haut ;
- **bottom**: bas ;
- **left**: gauche ;
- **right**: droite.



On peut utiliser margin ou padding en précisant les 4 valeurs (haut, droite, bas, gauche)

Exemple : margin: 2px 0 3px 1px;

Ce qui signifie :

" 2 pixels de marge en haut, 0 pixel de marge à droite (le px est facultatif dans ce cas), 3 pixels de marge en bas et 1 pixel de marge à gauche".

Centrez vos blocs avec width et margin: auto

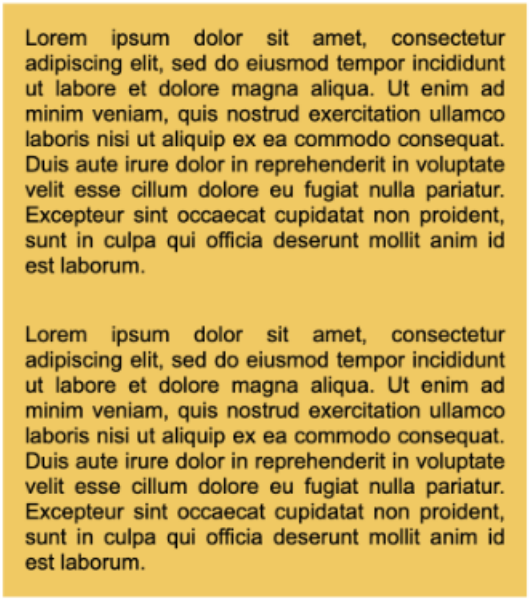
Pour centrer des blocs, il faut respecter les règles suivantes :

1. donner une largeur au bloc avec la propriété **width** ;
2. indiquer **margin: auto;** (les marges extérieures seront alors automatiques, et permettront de centrer le contenu.

Essayons cette technique sur nos paragraphes, en ajoutant simplement un

```css
p {
    width: 350px;
    background-color: #F1C864;
    text-align: justify;
    padding: 16px;
    margin: auto;
}
```

Et on obtient :



Les paragraphes sont centrés sur la page

Il n'est cependant pas possible de centrer verticalement un bloc avec cette technique. Seul le centrage horizontal est permis.

EN RÉSUMÉ :

- On distingue deux principaux types de balises en HTML :
    - les balises de type block comme <p> ou <h1> créent un retour à la ligne et occupent par défaut toute la largeur disponible. Elles se suivent de haut en bas ;
    - les balises de type inline comme <a> ou délimitent du texte au milieu d'une ligne. Elles se suivent de gauche à droite.
- On peut modifier la taille d'une balise de type block avec les propriétés CSS width (largeur) et height (hauteur).
- Les éléments de la page disposent chacun de padding (marges intérieures) et de margin (marges extérieures).
- On peut centrer le contenu d'un bloc dont la largeur est définie par width avec margin: auto;

Faites votre mise en page avec Flexbox

Comprenez la logique : un conteneur, des éléments

Pour faire de la mise en page avec flexbox :

1. Définir un conteneur.
2. Et placer à l'intérieur plusieurs éléments.

   Sur la même page on peut avoir plusieurs contenurs

Un conteneur (*container* en anglais) est une balise qui peut renfermer d'autres balises, comme du texte ou encore des images. Les conteneurs les plus célèbres sont les balises et .

```
<div class="container">
<div class="element element1">Élément 1</div>
<div class="element element2">Élément 2</div>
<div class="element element3">Élément 3</div>
</div>
```

➔ En faisant ça mes éléments seront l'un sur l'autre (à la verticale)

Alignez les éléments d'un conteneur avec display: flex

Css :

```
.container{
display: flex;

}
```

⇨ Mes blocs vont ressembler à ça (côte à côte)

⇨ 

Donnez leur une direction avec la propriété flex-direction

- row : organisés sur une ligne (par défaut) ;
- column : organisés sur une colonne ;
- row-reverse : organisés sur une ligne, mais en ordre inversé ;
- column-reverse : organisés sur une colonne, mais en ordre inversé.

```
.container {
   display: flex;
   flex-direction: column;
}
```

C’est la même chose qu’au début, mais maintenant ils ont un tas de propriétés

```
.container {
   display: flex;
   flex-direction: column-reverse;
}
```

Les blocs sont dans l’ordre inverse sans toucher le HTML

Retournez à la ligne avec la propriété flex-wrap

Les blocs par défaut essaient d’être sur la même ligne, quitte à s’écraser s’ils n’ont plus de place et provoquer des anomalies.

Voilà les différentes valeurs de flex-wrap :

1. nowrap : pas de retour à la ligne (par défaut) ;
2. wrap : les éléments vont à la ligne lorsqu'il n'y a plus la place ;
3. wrap-reverse : les éléments vont à la ligne, lorsqu'il n'y a plus la place, en sens inverse.

```
.container {
    display: flex;
    flex-wrap: nowrap;
    /* OU wrap;
    OU wrap-reverse; */
}
```

Dans les illustrations suivantes, vous pouvez voir les différents comportements de nos blocs en fonction de la valeur que l'on attribue à `flex-wrap` :

Avec `flex-wrap: nowrap;` :

Les éléments se serrent pour tenir sur la même ligne

Avec `flex-wrap: wrap;` :

Les éléments passent à la ligne s'ils ne rentrent pas tous sur la

Avec `flex-wrap: wrap-reverse;` :

Les éléments passent à la ligne mais à l'envers

Alignez les éléments sur un axe principal et secondaire

Les éléments sont organisés par défaut de manière horizontale mais on peut les organiser de manière verticale → l'axe principal

- si vos éléments sont organisés horizontalement, l'axe secondaire est l'axe vertical ;
- si vos éléments sont organisés verticalement, l'axe secondaire est l'axe horizontal.

Alignez sur l'axe principal avec la propriété justify-content

Pour changer leur alignement, on va utiliser justify-content, qui peut prendre ces valeurs :

- flex-start: alignés au début (par défaut) ;
- flex-end: alignés à la fin ;
- center: alignés au centre ;
- space-between: les éléments sont étirés sur tout l'axe (il y a de l'espace entre eux) ;
- space-around: idem, les éléments sont étirés sur tout l'axe, mais ils laissent aussi de l'espace sur les extrémités.

```css
.container {
    display: flex;
    justify-content: flex-start /* OU flex-end
    OU center
    OU space-between
    OU space-around; */
}
```

Verticale :

```css
.container {
    display: flex;
    flex-direction: column;
    justify-content: center;
    height: 350px; /* Un peu de hauteur pour que les éléments aient la place de
  bouger */
}
```

Répartissez les blocs sur plusieurs lignes avec align-content

Si on plusieurs lignes dans notre flexbox on choisit align-content

Aucun n'effet si une seule ligne

Prenons donc un cas de figure où nous avons plusieurs lignes. J'autorise les éléments à aller à la ligne avec **flex-wrap**.

- stretch (par défaut) : les éléments s'étirent pour occuper tout l'espace ;
- flex-start : les éléments sont placés au début ;
- flex-end : les éléments sont placés à la fin ;
- center : les éléments sont placés au centre ;
- space-between : les éléments sont séparés avec de l'espace entre eux ;
- space-around : idem, mais il y a aussi de l'espace au début et à la fin.

EN RÉSUMÉ

- Le principe de Flexbox est d'avoir un conteneur avec plusieurs éléments à l'intérieur. Avec `display: flex;` sur le conteneur, les éléments à l'intérieur sont agencés en mode Flexbox (horizontalement, par défaut).
- Flexbox peut gérer toutes les directions. Avec `flex-direction` , on peut indiquer si les éléments sont agencés horizontalement (par défaut) ou verticalement. Cela définit ce qu'on appelle l'*axe principal*.
- L'alignement des éléments se fait sur l'axe principal avec `justify-content` et sur l'axe secondaire avec `align-items` .
- Avec `flex-wrap`, on peut autoriser les éléments à revenir à la ligne s'ils n'ont plus d'espace.
- S'il y a plusieurs lignes, on peut indiquer comment les lignes doivent se répartir entre elles avec `align-content`.

Découvrez les bases de CSS Grids

Définissez une grid avec la propriété CSS display: grid

.conteneur {

display: grid;

}

On va définir les colonnes et les rangées avec deux propriétés :

1. **grid-template-columns** pour le nombre de colonnes, et la largeur de chacune d'entre elles.
2. **grid-template-rows** pour le nombre de rangées, et la hauteur de chacune d'entre elles.

Définissez vos colonnes avec la propriété CSS grid-template-columns

```
<div class="conteneur">
    <div class="box">🐸 Élément 1</div>
    <div class="box">🦊 Élément 2</div>
    <div class="box">🦄 Élément 3</div>
    <div class="box">🐵 Élément 4</div>
    <div class="box">🐻 Élément 5</div>
    <div class="box">🐿️ Élément 6</div>
    <div class="box">🦆 Élément 7</div>
    <div class="box">🦞 Élément 8</div>
    <div class="box">🐳 Élément 9</div>
</div>
```

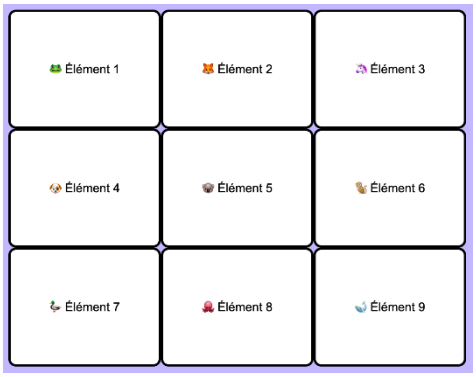
.box {

height: 150px;

}

Plus bas, dans le fichier CSS, créons 3 colonnes, chacune de 200px de large, à l'aide de la propriété **grid-template-columns**:

```
.conteneur {
    display: grid;
    grid-template-columns: 200px 200px 200px;
}
```



le nombre de colonne a été déduit par le navigateur

Définissez vos rangées avec la propriété CSS grid-template-rows

Passons maintenant à la propriété **grid-template-rows**. Et on fait bien attention à supprimer la hauteur fixe qu'on avait associée à la classe .box : height: 150px; .

on va pouvoir spécifier avec **grid-template-rows** , en indiquant la hauteur de chacune, comme on l'a fait pour grid-template-columns :

```css
.conteneur {
    display: grid;
    grid-template-columns: 200px 200px 200px;
    grid-template-rows: 100px 150px 200px;
}
```

Si vous laissez la hauteur fixe et que vous indiquez en même temps différentes hauteurs pour les rangées :

css

```css
.box {
  height: 150px;
}

.conteneur {
    display: grid;
    grid-template-columns: 200px 200px 200px;
    grid-template-rows: 100px 150px 200px;
}
```

Aérez votre contenu avec la propriété CSS gap

La propriété CSS **gap** permet de créer des espacements entre vos éléments. Si vous voulez garder les mêmes distances entre les rangées et les colonnes sans avoir à vous compliquer la vie, vous précisez simplement une valeur.

```css
.conteneur {
    display: grid;
    grid-template-columns: 200px 200px 200px;
    grid-template-rows: 100px 150px 200px;
    gap: 10px;
}
```

Et on obtient :

Choisissez vos unités

On peut utiliser : px, em, rem, pourcentages

Les « fractions units » => fr

Il est plus simple d'utiliser fr que les pourcentages

Définissez la taille des éléments de votre grid

Pour cela, vous allez apprendre à maîtriser les propriétés liées à **grid-column** et **grid-row**.

À chaque fois que vous écrivez **display: grid**, le navigateur se représente votre conteneur comme un ensemble d'éléments délimités par des lignes horizontales et verticales. Ces lignes sont invisibles, mais peuvent être inspectées avec les outils de développement.

On aura besoin des propriétés suivantes pour déclarer nos éléments :

- **grid-column-start** indique la ligne verticale de départ de l'élément ;
- **grid-column-end** indique la ligne verticale d'arrivée de l'élément ;
- **grid-row-start** indique la ligne horizontale de départ de l'élément ;
- **grid-row-end** indique la ligne horizontale d'arrivée de l'élément.

Mesurez vos colonnes

Attendez, mais pourquoi on dit d'aller jusqu'à 4 pour prendre toute la largeur, alors qu'il n'y a que 3 colonnes ? On ne devrait pas plutôt dire grid-column-end: 3; pour dire qu'elle s'arrête à la troisième colonne ?

Et non car ça reviendrait à dire qu'elle s'arrête juste avant le dernier tiers, donc juste avant la troisième colonne. C'est déroutant je vous l'accorde mais en fait on compte +1 avec cette technique. Le navigateur considère que pour couvrir toute la largeur, on "atteint" un élément suivant.

EN RÉSUMÉ

- Les CSS Grids sont complémentaires à Flexbox et permettent de créer facilement des mises en page plus élaborées que Flexbox, sans forcément avoir des éléments de la même taille.
- Pour déclarer une grid, on déclare simplement `display: grid;` sur le conteneur : notre navigateur comprend tout de suite que nos éléments sont dans la grid.
- On définit les colonnes avec `grid-template-columns` et les rangées avec `grid-template-rows` : en fonction du nombre de valeurs passées, de nouvelles colonnes et rangées sont créées.
- En plus des unités classiques `px`, `em`, `rem` et `%`, les `fr` sont encore plus simples, et permettent d'indiquer une fraction de la grille.
- `gap` permet d'espacer les éléments entre eux.
- Les grids créent implicitement des lignes horizontales et verticales délimitant les différentes rangées et colonnes.
- Chaque élément peut avoir :
    - son propre point de départ horizontal avec `grid-row-start` ;
    - son point d'arrivée horizontal avec `grid-row-end` ;
    - son point de départ vertical avec `grid-column-start` ;
    - et son point d'arrivée vertical avec `grid-column-end` .

Abordez d’autres techniques de mise en page

Transformez vos éléments avec display

Display : permet de transformer n’importe quel élément de la page

On peut imposer à des liens qui sont inline d’apparaitre sous forme de type block

Ex :

```
a {
   display: block;
}
```

Cachez vos éléments avec display: none

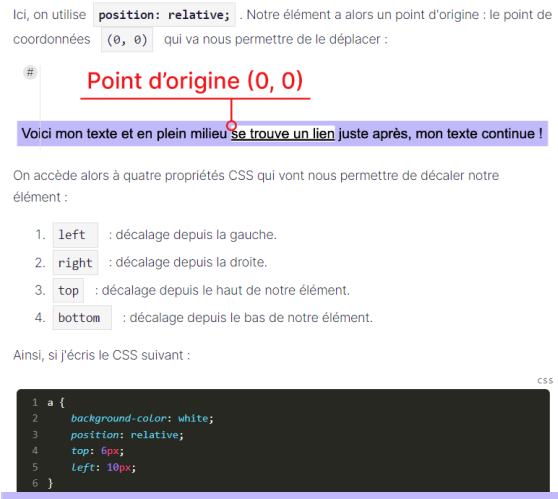
On peut masquer complétement un élément de la page

```
.secret {
   display: none;
}
```

Positionnez vos éléments avec la propriété CSS position

Position : Elle permet de positionner avec précision des éléments sur la page (et même parfois de les superposer).

- Le **positionnement relatif** permet de décaler l'élément par rapport à sa position normale.
- Le **positionnement absolu** permet de placer un élément n'importe où sur la page (en haut à gauche, en bas à droite, tout au centre, etc.).
- D'autres **types de positionnements** comme **fixed** ou encore **sticky**, qui ont des comportements assez semblables, peuvent être utiles si l'on veut qu'un élément, tel que le menu de navigation par exemple, reste à l'écran.



Voici mon texte et en plein milieu se trouve un lien juste après, mon texte continue !

Définissez un positionnement absolu avec `position: absolute;`

Le positionnement absolu permet de placer un élément (réellement) n'importe où sur la page. Pour l'utiliser, on écrit tout simplement `position: absolute;` :

```css
a {
    background-color: white;
    position: absolute;
    top: 6px;
    left: 10px;
}
```

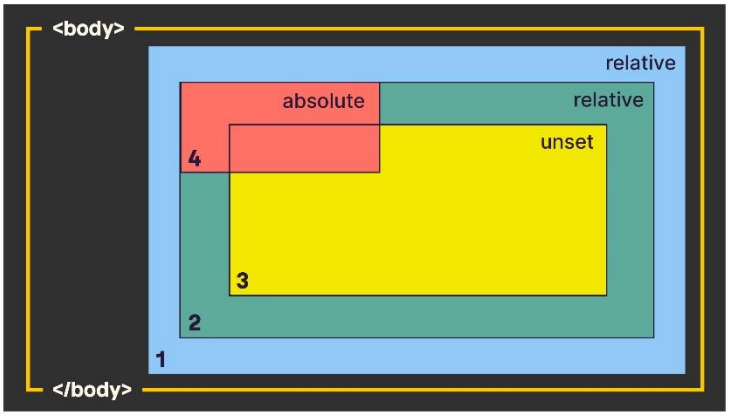
Regardez ce que ça donne :

se trouve un lien
Voici mon texte et en plein milieu juste après, mon texte continue !

Le lien est décalé par rapport à la page

On peut utiliser : left, right, top, bottom

On peut leur donner une valeur en pixel, en pourcentage



Gérez le chevauchement avec la propriété z-index

la propriété **z-index** pour indiquer quel élément doit apparaître au-dessus des autres : vous empilez vos éléments dans l'ordre souhaité ! L'élément ayant la valeur de z-index la plus élevée sera placé par-dessus les autres.

Bloquez un élément avec fixed ou sticky

le bloc se fige, même si on descend plus bas dans la page. Pour faire cela, il y a deux options possibles : attribuer la valeur fixed ou sticky à la propriété CSS position

position : fixed ; => élément fixe

position : sticky ; => élément adhérent

EN RÉSUMÉ

- La propriété display permet de changer le comportement de base d'un élément :
    - transformer un élément inline en block ;
    - inversement, un élément block en inline ;
    - mais aussi de faire un mélange des deux avec inline-block.
- Le positionnement relatif permet de décaler un bloc par rapport à sa position normale.
- Le positionnement absolu permet de placer un élément où l'on souhaite sur la page, au pixel près.
- Un élément A positionné en absolu à l'intérieur d'un autre élément B (lui-même positionné en absolu, fixe ou relatif) se positionne par rapport à l'élément B, et non par rapport au coin en haut à gauche de la page.
- Lorsque plusieurs éléments s'empilent, il est possible de les ordonner avec z-index.

Ajoutez des tableaux

Créez un tableau en HTML et CSS

Balise en pair (de type block) : <table></table> indique le début et la fin du tableau

- **<tr> </tr>** indique le début et la fin d'**une ligne du tableau** ;
- **<td> </td>** indique le début et la fin du contenu d'**une cellule**.

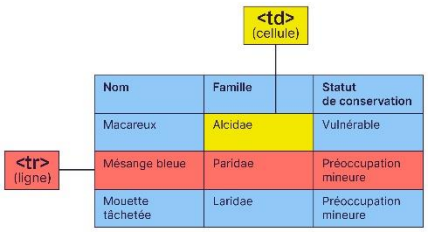


Tableau 2 lignes, 3 colonnes :

```html
<table>
    <tr>
        <td>Carmen</td>
        <td>33 ans</td>
        <td>Espagne</td>
    </tr>
    <tr>
        <td>Michelle</td>
        <td>26 ans</td>
        <td>États-Unis</td>
    </tr>
</table>
```

→ résultat : carmen 33 ans Espagne

- Michelle 26 ans États-Unis

Ajoutez des bordures au tableau HTML grâce à la propriété CSS border

On utilise `td` pour selectionner toutes les cellules des tableaux de la page. Donc pour créer des bordures à nos cellules, on fait :

```css
td /* Toutes les cellules des tableaux... */
{
    border: 1px solid black; /* auront une bordure de 1px */
}
```

le résultat :

Carmen	33 ans	Espagne
Michelle	26 ans	États-Unis

Collez les bordures du tableau HTML avec la propriété CSS border-collapse

border-collapse est une propriété CSS qui permet de “coller les bordures entre elles”. Elle peut prendre deux valeurs :

1. collapse : les bordures seront collées entre elles, c'est l'effet qu'on recherche ici ;
2. separate : les bordures seront dissociées (valeur par défaut).

```css
table {
    border-collapse: collapse;
}

td {
    border: 1px solid black;
}
```

Ce qui nous permet d'obtenir :

Carmen	33 ans	Espagne
Michelle	26 ans	États-Unis

Ajoutez une ligne d'en-tête au tableau avec la balise HTML <th>

La ligne d'en-tête se crée avec un <tr> mais les cellules qu'elle contient sont, cette fois, encadrées par des balises **<th>** (pour *table header* ou "en-tête de tableau" en français) et non pas <td> !

structure :

```
<table>
    <tr>
        <th>Nom</th>
        <th>Âge</th>
        <th>Pays</th>
    </tr>
    <tr>
        <td>Carmen</td>
        <td>33 ans</td>
        <td>Espagne</td>
    </tr>
    <tr>
        <td>Michelle</td>
        <td>26 ans</td>
        <td>États-Unis</td>
    </tr>
</table>
```

La ligne d'en-tête est très facile à reconnaître pour deux raisons :

- les cellules sont des**<th>** au lieu des <td> habituels ;
- c'est la première ligne du tableau (c'est idiot, mais encore faut-il le préciser).

Par défaut les entêtes sont en gras

Donnez un titre au tableau avec la balise HTML <caption>

Caption : nous permet d'ajouter un titre à notre tableau

```
<table>
    <caption>Passagers du vol 377</caption>
    <tr>
        <th>Nom</th>
        <th>Âge</th>
        <th>Pays</th>
    </tr>
    <tr>
        <td>Carmen</td>
        <td>33 ans</td>
        <td>Espagne</td>
    </tr>
    <tr>
        <td>Michelle</td>
        <td>26 ans</td>
        <td>États-Unis</td>
    </tr>
</table>
```

Et voilà le résultat :

Passagers du vol 377

Nom	Âge	Pays
Carmen	33 ans	Espagne
Michelle	26 ans	États-Unis

On peut changer la position du titre avec la propriété CSS : caption-side par défaut elle est sur top, on peut la mettre en bas (botttom)

On peut ajouter des marges à notre tableau avec du padding

```css
td, th {
    border: 1px solid black;
    padding: 15px;
}
```

:t voilà le résultat avec une marge de 15 p

Passagers du vol 377

Nom	Âge	Pays
Carmen	33 ans	Espagne
Michelle	26 ans	États-Unis

Centrez un tableau HTML avec la propriété CSS margin

C'est également sous le sélecteur `table` qu'on va indiquer au navigateur de centrer le tableau sur la page grâce à la propriété CSS `margin`, pour laquelle on va préciser la valeur `auto` :

css

```css
table {
    border-collapse: collapse;
    margin: auto;
}
```

Voici ce que ça donne :

Passagers du vol 377

Nom	Âge	Pays
Carmen	33 ans	Espagne
Michelle	26 ans	États-Unis

Le tableau est centré sur la page

Structurez un grand tableau avec des balises HTML

Divisez un tableau avec les balises HTML thead, tbody et tfoot (généralement dans les gros tableaux)

1. **l'en-tête du tableau (en haut)** se définit avec les balises **<thead></thead>**
2. **le corps du tableau (au centre)** se définit avec les balises **<tbody></tbody>**
3. **le pied du tableau (en bas)** se définit avec les balises **<tfoot></tfoot>**

il est conseillé d'écrire les balises dans l'ordre suivant :

1. <thead>
2. <tfoot>
3. <tbody>

Fusionnez des cellules du tableau avec les attributs colspan et rowspan

Pour fusionner des cellules entre elles, il suffit d'ajouter un attribut dans la balise HTML **<td>**

1. **l'attribut colspan permet de fusionner des colonnes** : la fusion s'effectue horizontalement, aussi bien sur les lignes d'en-tête que sur le contenu lui-même.
2. **l'attribut rowspan permet de fusionner des lignes** : là, deux lignes seront groupées entre elles. La fusion s'effectuera verticalement.

<table>
<tr><th>Titre du film</th><th>Pour enfants ?</th><th>Pour adolescents ?</th></tr>
<tr><td>Massacre à la tronçonneuse</td><td>Non, trop violent</td><td rowspan="2">Oui</td></tr>
<tr><td>The Ring</td><td>Effrayant</td></tr>
<tr><td>Les bisounours font du ski</td><td>Oui, adapté</td><td>Pas assez violent...</td></tr>
<tr><td>Lucky Luke, seul contre tous</td><td colspan="2">Pour toute la famille !</td></tr>
</table>

EN RÉSUMÉ

- Un tableau s'insère avec la balise HTML <table> et se définit ligne par ligne avec <tr>.
- Chaque ligne comporte des cellules <td> (cellules normales) ou <th> (cellules d'en-tête).
- Le titre du tableau se définit avec <caption>.
- On peut ajouter une bordure aux cellules du tableau HTML avec la propriété CSS border . Pour coller les bordures entre elles, on utilise la propriété CSS border-collapse.
- Un tableau peut être divisé en trois sections grâce aux balises HTML <thead> (en-tête), <tbody> (corps) et <tfoot> (bas du tableau). L'utilisation de ces balises n'est pas obligatoire.
- En HTML, on peut fusionner des cellules horizontalement avec l'attribut colspan, ou verticalement avec rowspan. Il faut indiquer combien de cellules doivent être fusionnées.

Créez des formulaires

Pour créer un formulaire : <form></form>

Voici les deux attributs indispensables pour construire un formulaire fonctionnel :

1. L'attribut **method** indique par quel moyen les données vont être envoyées. Ne vous en préoccupez pas pour le moment. Sachez juste que les méthodes les plus utilisées sont **get** et **post**.
2. L'attribut **action** indique l'adresse de la page ou du programme qui va traiter les informations.

Structure :

```
<p>Texte avant le formulaire</p>
    <form method="get" action="">
        <p>Texte à l'intérieur du formulaire</p>
    </form>
<p>Texte après le formulaire</p>
```

On laisse le champ vide pour action car il n y a pas de backend

Insérez des champs de texte avec la balise HTML <input>

Balise orpheline: <input>

La valeur de son attribut : ***type***, changera le type de champ : texte, email, date, password, etc

Ajoutez une zone de texte monoligne avec <input type="text">

<input type="text">

Il faut donner un nom, même s'il n'apparait pas sur la page, il est indispensable parce qu'il permettra plus tard d'identifier le champ

<input type="text" name="prenom">

Pour notre formulaire uniquement le champ texte:

<form method="get" action="">

<input type="text" name="prenom">

</form>

Ajoutez un champ de libellé avec la balise HTML <label> </label>

Un champ prérempli avec une indication pour guider le visteur

<form method="get" action="">

<label>Votre prénom</label> : <input type="text" name="prenom">

</form>

Mais cela ne suffit pas : il faut lier le label à la zone de texte.

Pour ce faire, on doit donner un nom à la zone de texte, non pas avec l'attribut name mais avec l'attribut **id** (que l'on peut utiliser sur toutes les balises).

- L'attribut **id** est utilisé pour **identifier** l'élément HTML afin d'y accéder et de le manipuler. Il est donc unique pour cet élément.
- L'attribut **name**, lui, réfère à la variable du formulaire que l'élément concerne. Ici, il n'y a qu'un seul élément qui pourra référer à la variable prenom ; name et id auront donc la même valeur. Mais lorsque nous utiliserons des checkbox ou des radio, plusieurs éléments correspondront à la même variable. Mais nous verrons cela un peu plus tard, et vous comprendrez mieux.

<form method="get" action="">

<label for="prenom">Votre prénom</label> : <input type="text" name="prenom" id="prenom">

</form>

On peut ajouter un certain nombre d'autres attributs à la balise `<input>` pour personnaliser son fonctionnement :

- agrandir le champ avec `size` ;
- limiter le nombre de caractères que l'on peut saisir avec `maxlength` ;
- pré-remplir le champ avec une valeur par défaut à l'aide de `value` ;
- donner une indication sur le contenu du champ avec `placeholder` . Cette indication disparaîtra dès que le visiteur aura cliqué à l'intérieur du champ.

```html
<form method="get" action="">
    <p>
        <label for="prenom">Votre prénom :</label>
        <input type="text" name="prenom" id="prenom" placeholder="Ex. : Mathieu" size="30" maxlength="20">
    </p>
</form>
```

Testez vous-même le résultat pour observer le comportement du champ. En attendant, voici le rendu du champ dans son état initial (avant que le visiteur ne renseigne quoi que ce soit) :

Votre prénom : Ex. : Mathieu

Ajoutez une zone de texte multiligne avec <textarea> </textarea>

Lorsque un utilisateur a besoin d'écrire un paragraphe

Comme pour chaque élément : il faut donner un *name* et utiliser un *label* qui explique de quoi il s'agit :

```html
<form method="get" action="">
    <p>
        <label for="ameliorer">Des conseils pour améliorer mon site ?</label>
<br>
        <textarea name="ameliorer" id="ameliorer"></textarea>
    </p>
</form>
```

Et voici le résultat en image :

Des conseils pour améliorer mon site ?

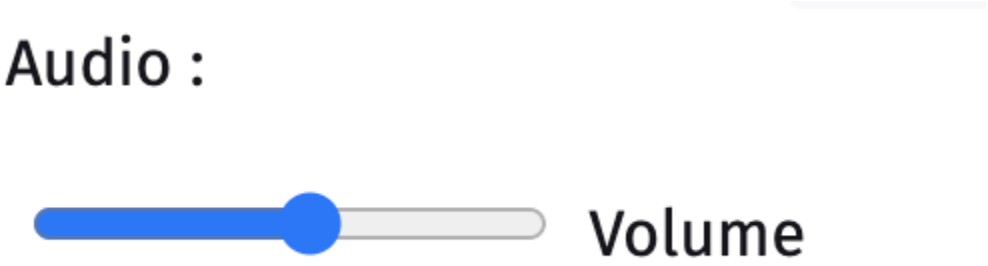
Mais on peut modifier la taille par défaut du <textarea>, et ce, de deux façons différentes :

1. En appliquant au <textarea> les **propriétés CSS** width et height .
2. En ajoutant à la balise <textarea> **les attributs** :
    1. rows (indique le **nombre de lignes** qui peuvent être affichées) ;
    2. et cols (indique le **nombre de colonnes** qui peuvent être affichées).

Pas besoin d'utiliser l'attribut : ***value*** pour pré-remplir, on écrit tout simplement le texte entre la balise ouvrante et fermante

Utilisez `<input>` pour des formats de saisie particuliers

Type de champ	Code

un e-mail	**<input type="email">**
une URL	**<input type="url">** On peut demander à saisir une adresse absolue, commençant généralement par http://
un numéro de téléphone	**<input type="tel">**
un nombre entier	**<input type="number">** Le champ s'affiche en général avec des petites flèches pour changer la valeur.
un curseur (aussi appelé “slider”)	**<input type="range">** On utilise range pour demander au visiteur une valeur comprise entre deux bornes :  Un champ de curseur
une date	Différents types de champs de sélection de date existent, comme **<input type="date">** pour sélectionner une date :

HTML CSS

```
<label for="start">Date :</label>

<input type="date" id="start" name="trip-start"
       value="2023-01-01"
       min="2023-01-01" max="2025-12-31">
```

Un champ de date et le code associé

Mais il existe des variantes :

- **<input type="time">** pour l'heure ;
- **<input type="week">** pour la semaine ;
- **<input type="month"** pour le mois ;
- **<input type="datetime">** pour la date et l'heure (avec gestion du décalage horaire) ;
- **<input type="datetime-local">** pour la date et l'heure (sans gestion du décalage horaire).

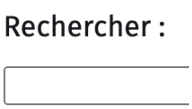
Vérifiez bien quels navigateurs gèrent ce type de champ, il n'est pas encore complètement reconnu.

une recherche

<input type="search">

HTML CSS

```
<label for="site-search">Rechercher :</label>
<input type="search" id="site-search" name="q">

<button>GO !</button>
```



Un champ de recherche et le code associé

Pour les types de champs nombre, date et curseur, vous pouvez personnaliser le fonctionnement du champ avec les attributs suivants :

- min : valeur minimale autorisée ;
- max : valeur maximale autorisée ;
- step : c'est un “pas” de déplacement. Si vous indiquez un pas de 2, le champ n'acceptera que des valeurs de 2 en 2 (par exemple 0, 2, 4, 6…).

Laissez le visiteur choisir une option

Offrir plusieurs manières aux users de choisir :

1. Les **cases à cocher** (plusieurs choix possibles).
2. Les **boutons radio** (un seul choix possible).
3. Les **listes déroulantes** (un seul choix possible).

Insérez des cases à cocher avec <input type="checkbox">

```
<input type="checkbox" name="choix">
```

:ajoutez un `<label>` et le tour est joué :

```
<form method="get" action="">
    <p>
        Cochez les aliments que vous aimez manger :<br>
        <input type="checkbox" name="frites" id="frites"> <label for="frites">Frites</label><br>
        <input type="checkbox" name="steak" id="steak"> <label for="steak">Steak</label><br>
        <input type="checkbox" name="epinards" id="epinards"> <label for="epinards">Épinards</label><br>
        <input type="checkbox" name="huitres" id="huitres"> <label for="huitres">Huitres</label>
    </p>
</form>
```

:e qui donne, une liste d'aliments (le visiteur a la possibilité d'en cocher plusi

Cochez les aliments que vous aimez manger :
- ☐ Frites
- ☐ Steak
- ☐ Épinards
- ☐ Huîtres

N'oubliez pas de donner un nom différent à chaque case à cocher, cela vous permettra d'identifier plus tard lesquelles ont été cochées par le visiteur.

Vous pouvez faire en sorte qu'une case soit cochée par défaut, avec l'attribut `checked` : `<input type="checkbox" name="choix" checked>`.

Insérez des boutons radio avec <input type="radio">

Faire un choix parmi une liste :

```html
<form method="get" action="">
    <p>
        Veuillez indiquer la tranche d'âge dans laquelle vous vous situez :
<br>
        <input type="radio" name="age" value="moins15" id="moins15"> <label
for="moins15">Moins de 15 ans</label><br>
        <input type="radio" name="age" value="medium15-25" id="medium15-25">
<label for="medium15-25">15-25 ans</label><br>
        <input type="radio" name="age" value="medium25-40" id="medium25-40">
<label for="medium25-40">25-40 ans</label><br>
        <input type="radio" name="age" value="plus40" id="plus40"> <label
for="plus40">Plus de 40 ans</label>
    </p>
</form>
```

Ce qui donne :

Veuillez indiquer la tranche d'âge dans laquelle vous vous situez :
- ○ Moins de 15 ans
- ○ 15-25 ans
- ○ 25-40 ans
- ○ Plus de 40 ans

Pourquoi avoir mis le même nom pour chaque option?

Pour que le navigateur sache de quel groupe la sélection fait partie et si on l'enlève on peut cocher tous les boutons.

Si deux zones de groupes différents, faut donner un nom différent à chaque groupe exemple :

Si vous avez deux zones d'options différentes, il faut donner un `name` unique à chaque groupe :

html

```html
<form method="get" action="">
    <p>
        Veuillez indiquer la tranche d'âge dans laquelle vous vous situez :
<br>
        <input type="radio" name="age" value="moins15" id="moins15"> <label
for="moins15">Moins de 15 ans</label><br>
        <input type="radio" name="age" value="medium15-25" id="medium15-25">
<label for="medium15-25">15-25 ans</label><br>
        <input type="radio" name="age" value="medium25-40" id="medium25-40">
<label for="medium25-40">25-40 ans</label><br>
        <input type="radio" name="age" value="plus40" id="plus40"> <label
for="plus40">Plus de 40 ans</label>
    </p>
    <p>
        Sur quel continent habitez-vous ?<br>
        <input type="radio" name="continent" value="europe" id="europe">
<label for="europe">Europe</label><br>
        <input type="radio" name="continent" value="afrique" id="afrique">
<label for="afrique">Afrique</label><br>
        <input type="radio" name="continent" value="asie" id="asie"> <label
for="asie">Asie</label><br>
        <input type="radio" name="continent" value="amerique" id="amerique">
<label for="amerique">Amérique</label><br>
        <input type="radio" name="continent" value="oceanie" id="oceanie">
<label for="oceanie">Océanie</label>
    </p>
</form>
```

Ce qui donne :

Veuillez indiquer la tranche d'âge dans laquelle vous vous situez :
- ○ Moins de 15 ans
- ○ 15-25 ans
- ○ 25-40 ans
- ○ Plus de 40 ans

Sur quel continent habitez-vous ?
- ○ Europe
- ○ Afrique
- ○ Asie
- ○ Amérique
- ○ Océanie

Insérez une liste déroulante avec les balises <select> et <option>

Le fonctionnement est un peu différent :

1. On utilise la balise **<select> </select>** pour **indiquer le début et la fin de la liste déroulante**.
2. On ajoute l'attribut name à la balise pour **donner un nom à la liste**.
3. Puis, à l'intérieur du <select> </select>, on place plusieurs balises **<option> </option>** (une par choix possible), pour donner à chacune d'elles un attribut value afin d'identifier ce que le visiteur a choisi :

```html
<form method="get" action="">
    <p>
        <label for="pays">Dans quel pays habitez-vous ?</label><br>
        <select name="pays" id="pays">
            <option value="espagne">Espagne</option>
            <option value="royaume-uni">Royaume-Uni</option>
            <option value="canada">Canada</option>
            <option value="japon">Japon</option>
        </select>
    </p>
</form>
```

Si vous voulez qu'une option soit sélectionnée par défaut, utilisez l'attribut `selected` :

```html
<option value="canada" selected>Canada</option>
```

À vous d'essayer pour voir ce que ça donne :

Dans quel pays habitez-vous ?
Canada

EN RÉSUMÉ

- Un formulaire est englobé par la balise HTML <form></form> à laquelle on ajoute les attributs method et action.
- On utilise ensuite d'autres balises HTML pour permettre au visiteur du site de saisir des informations :
    - la balise orpheline <input> pour un champ de saisie monoligne ;
    - la balise en paire <textarea> </textarea> pour un champ de saisie multiligne ;
    - la balise en paire <select> </select> suivi d'options avec la balise en paire <option> </option> pour une liste déroulante.
- Le champ <input> peut également être configuré pour saisir d'autres types de données : e-mail, URL, numéro de téléphone, date, etc.
- Un label peut être relié à n'importe quel input avec l'attribut for correspondant à l'attribut id pour le champ utilisé.

Finalisez un formulaire et ajoutez un bouton d'envoi

Regroupez des champs avec la balise <fieldset>

Si on a un grand formulaire, il peut être utile de les regrouper avec plusieurs balises <fieldset>

Chaque fieldset peut contenir une <legend>

Exemple :

```html
<form method="get" action="">
  <fieldset>
     <legend>Vos coordonnées</legend> <!-- Titre du fieldset -->
     <label for="nom">Quel est votre nom ?</label>
     <input type="text" name="nom" id="nom">
     <label for="prenom">Quel est votre prénom ?</label>
     <input type="text" name="prenom" id="prenom">
     <label for="email">Quel est votre e-mail ?</label>
     <input type="email" name="email" id="email">
  </fieldset>
  <fieldset>
     <legend>Votre souhait</legend> <!-- Titre du fieldset -->
     <p>
        Faites un souhait que vous voudriez voir exaucé :
        <input type="radio" name="souhait" value="riche" id="riche"> <label for="riche">Être riche</label>
        <input type="radio" name="souhait" value="celebre" id="celebre"> <label for="celebre">Être célèbre</label>
        <input type="radio" name="souhait" value="intelligent" id="intelligent"> <label for="intelligent">Être <strong>encore</strong> plus intelligent</label>
        <input type="radio" name="souhait" value="autre" id="autre"> <label for="autre">Autre...</label>
     </p>
     <p>
        <label for="precisions">Si "Autre", veuillez préciser :</label>
        <textarea name="precisions" id="precisions" cols="40" rows="4"></textarea>
     </p>
  </fieldset>
</form>
```

On peut ajouter du CSS pour avoir un style sur notre fieldset

Sélectionnez automatiquement un champ avec autofocus

Vous pouvez également placer automatiquement le curseur dans l'un des champs de votre formulaire, avec l'attribut `autofocus` . Dès que le visiteur chargera la page, le curseur se placera dans ce champ. C'est très pratique d'un point de vue de l'expérience utilisateur.

Par exemple, pour que le curseur soit par défaut dans le champ `prenom` , on écrit :

html

```html
<input type="text" name="prenom" id="prenom" autofocus>
```

Rendez un champ obligatoire avec required

Vous pouvez également faire en sorte qu'un champ soit obligatoire, en lui donnant l'attribut `required` .

html

```html
<input type="text" name="prenom" id="prenom" required>
```

Le navigateur indiquera alors au visiteur, si le champ est vide au moment de l'envoi, qu'il doit impérativement être rempli.

On dispose de pseudo-classes en CSS pour changer le style :

des éléments requis avec `:required` ;

et des éléments invalides avec `:invalid` .

N'oubliez pas non plus que vous disposez de la pseudo-classe `:focus` pour changer l'apparence d'un champ lorsque le curseur se trouve à l'intérieur.

Exemple pour colorer le fond des champs requis :

css

```css
:required {
    background-color:#F2A3BB;
}
```

Créez le bouton d'envoi du formulaire avec <input>

Elle existe en quatre versions :

1. **type="submit" : c'est celui que vous utiliserez le plus souvent.** Le visiteur sera conduit à la page indiquée dans l'attribut action du formulaire.
2. type="reset" : remise à zéro du formulaire (testez pour voir).
3. type="image" : équivalent du bouton submit, présenté cette fois sous forme d'image. Rajoutez l'attribut src pour indiquer l'URL de l'image.
4. type="button" : bouton générique, qui n'aura (par défaut) aucun effet. En général, ce bouton est géré en JavaScript pour exécuter des actions sur la page.

On peut changer le texte affiché à l'intérieur des boutons avec l'attribut `value` :

```html
<input type="submit" value="Envoyer">
```

Ce qui nous donne un input qui a l'apparence d'un bouton :

Le formulaire n'a pas de page backend spécifiée dans l'attribut "action", donc lorsque vous cliquez sur le bouton "Envoyer", le navigateur reste sur la même page. Cependant, le contenu du formulaire apparaîtra dans la barre URL du navigateur.

```
index.html?
nom=nebra&prenom=mathieu&email=coucou%40mathieu.com&souhait=autre&prec
isions=Apprendre+le+machine+learning
```

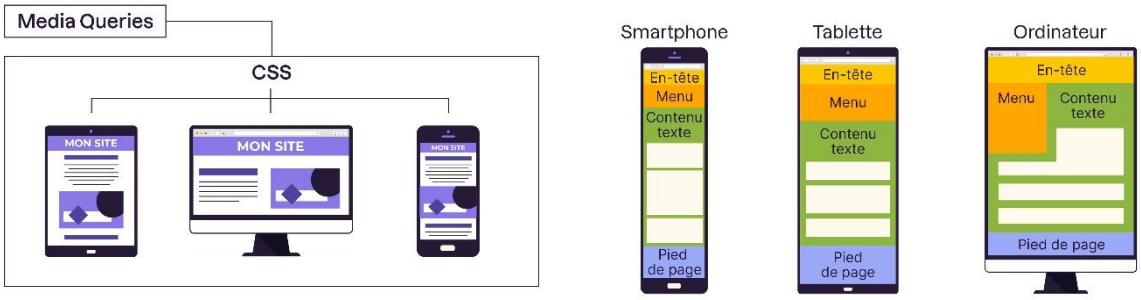
EN RÉSUMÉ

- Il est possible de regrouper les champs au sein d'une balise <fieldset> qui comporte une légende, avec la balise <legend>.
- On peut sélectionner automatiquement un champ en précisant l'attribut autofocus.
- Il est également possible de rendre un champ obligatoire avec l'attribut required.
- Le bouton qui permet de valider le formulaire est créé à partir d'une balise input en faisant : <input type="submit" value="Envoyer">

Utilisez le responsive design avec les Media Queries

Appliquez une media query avec @media

Les media queries sont des règles qui indiquent quand on doit appliquer des propriétés CSS, en fonction de la taille de l'écran sur lequel s'affiche le site web.



Syntaxe :

@media screen and (max-width: 1200px) {

/* Insérez vos propriétés CSS ici, avec vos sélecteurs*/

}

Les tailles min-width ou max-witdh sont souvent appelées breakpoint

> Attention, afin que les Media Queries soient prises en compte correctement sur tous les dispositifs, il est également essentiel de rajouter dans l'en-tête (partie `<head>` de notre site) la ligne suivante :
>
> ```
> <meta name="viewport" content="width=device-width, initial-scale=1.0">
> ```

Utilisez les règles disponibles

Il existe de nombreuses règles permettant de construire des media queries. Mais pour ne pas vous perdre, je vous ai uniquement listé les plus utilisées :

- height : hauteur de la zone d'affichage (fenêtre) ;
- width : largeur de la zone d'affichage (fenêtre) ;
- orientation : orientation du périphérique (portrait ou paysage) ;
- media : type d'écran de sortie. Avec la valeur la plus utilisée :
    - screen : écran “classique”,
    - all : tout type de média,
    - print : imprimante (pratique pour formater un contenu pour l’imprimer).

On peut rajouter le préfixe min- ou max- devant la plupart de ces règles. Ainsi, min-width signifie “largeur minimale”, max-height signifie “hauteur maximale”, etc.

Les règles peuvent être combinées à l'aide des mots suivants:

- only : “uniquement” ;
- and : “et” ;
- not : “non”.

Codez des interfaces responsives

Exploitez les minimum et maximum

On peut demander à ce qu'un bloc ait des dimensions minimales et maximales. C'est très pratique, car cela nous permet de définir des dimensions “limites” pour que notre site s'adapte aux différentes résolutions d'écran de nos visiteurs :

- min-width : largeur minimale ;
- min-height : hauteur minimale ;
- max-width : largeur maximale ;
- max-height : hauteur maximale.

Coupez ce qui dépasse avec overflow

Parfois, quand on définit des tailles pour nos blocs, ils peuvent devenir trop petits pour le texte qu'ils contiennent. Les propriétés CSS que nous allons voir permettent de contrôler les dépassements de texte et de décider quoi faire. Par exemple, vous pouvez définir une taille de 250px de large et 110px de haut pour un paragraphe avec une bordure et du texte plein à l'intérieur.

Si vous voulez que le texte ne dépasse pas les limites du paragraphe, il va falloir utiliser la propriété CSS overflow. Voici les valeurs qu'elle peut accepter :

- visible (par défaut) : si le texte dépasse les limites de taille, il reste visible et sort volontairement du bloc ;
- hidden : si le texte dépasse les limites, il sera tout simplement caché. On ne pourra pas voir tout le texte ;
- scroll : là encore, le texte sera caché s'il dépasse les limites. Sauf que cette fois, le navigateur mettra en place des barres de défilement pour qu'on puisse lire l'ensemble du texte. C'est un peu comme un cadre à l'intérieur de la page ;
- auto : c'est le mode “pilote automatique”. En gros, c'est le navigateur qui décide de mettre ou non des barres de défilement (il n'en mettra que si c'est nécessaire). C'est la valeur que je conseille d'utiliser le plus souvent.

EN RÉSUMÉ

Pour adapter un site internet aux différentes tailles d'écran, plusieurs techniques sont nécessaires :

- Il faut tout d'abord réussir à sélectionner les différentes tailles d'écran disponibles, afin d'appliquer le style souhaité.
- Pour cela, on a recours aux media queries qui sont la manière de "conditionner" le style appliqué : "Si l'écran de l'utilisateur est plus petit que la taille XXXpx , alors appliquer ce style".
- Il est ensuite possible d'utiliser les différentes astuces :
    - display: none pour cacher un élément qui ne rentre pas sur les tailles d'écran plus petites ;
    - overflow pour permettre de scroller avec la souris dans un élément container ;
    - min-width et max-width pour définir des tailles minimum et maximum.

1. La **liste des balises HTML**.
2. La **liste des propriétés CSS**.