

Module B1/B3

Programmation Web

Plan du cours

- Qu'est-ce que le Web ? Qu'est-ce qu'une page Web ?
- Apprendre à structurer une page web élémentaire (*titres, paragraphes, listes, liens, images*).
- Savoir insérer des médias, créer des tableaux, concevoir des formulaires.
- Découvrir comment séparer contenu et présentation, utiliser des sélecteurs et appliquer les styles de base (*couleurs, polices, espacements...*).
- Utiliser Flexbox et les media queries pour concevoir des pages adaptables à tous les écrans.
- Éditer l'apparence avec l'utilisation de polices, d'effets visuels, créer des transitions fluides.

Plan et objectifs

Fondamentaux du Web

Fondamentaux du CSS

Médias, Tableaux et Formulaires

Responsive Design

Styles avancés et animations

Bonnes pratiques et accessibilité

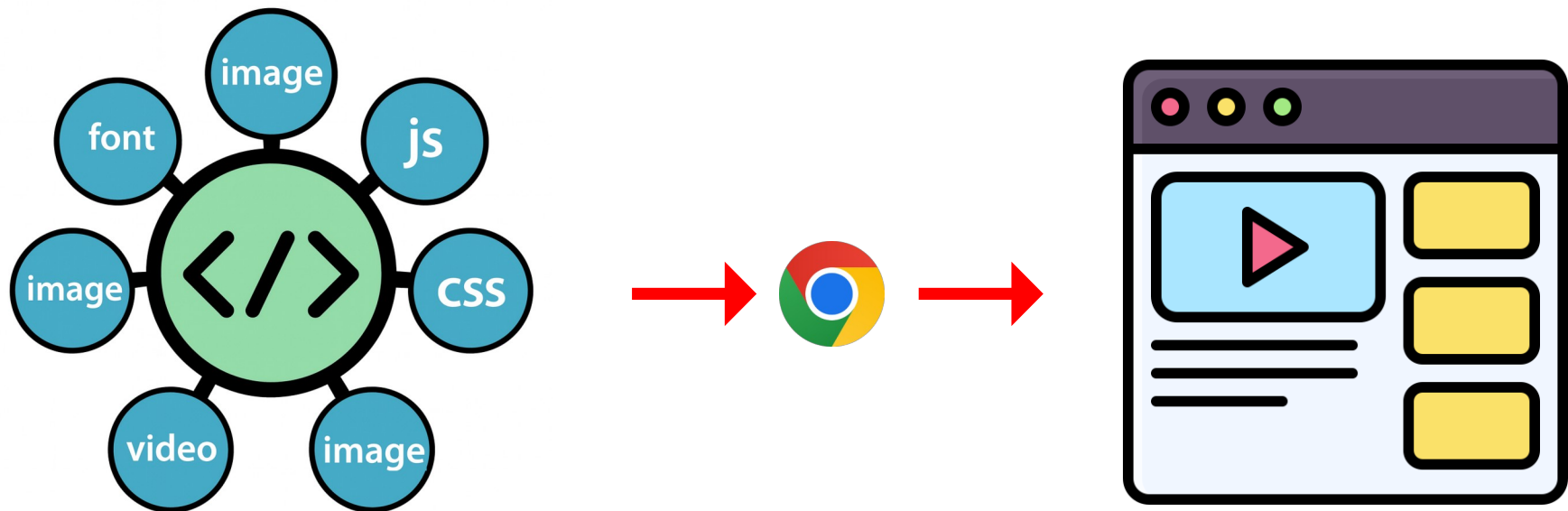
- Comprendre ce qu'est le Web (*url, hyperdocument, etc*).
- Comprendre la structure minimale d'une page Web (*DOM, doctype, html, head, body*).
- Identifier et utiliser les balises de base (*titres, paragraphes, listes, liens, images*).
- Découvrir les balises sémantiques (*header, nav, article, section, footer*).

Définition et fonctionnement Web

Un hyper-document (*page Web*) est un ensemble de fichiers textes que le navigateur interprète pour générer un affichage.

- Fichier principal : HTML
- Fichiers liés : CSS, JavaScript, images, sons, vidéos...

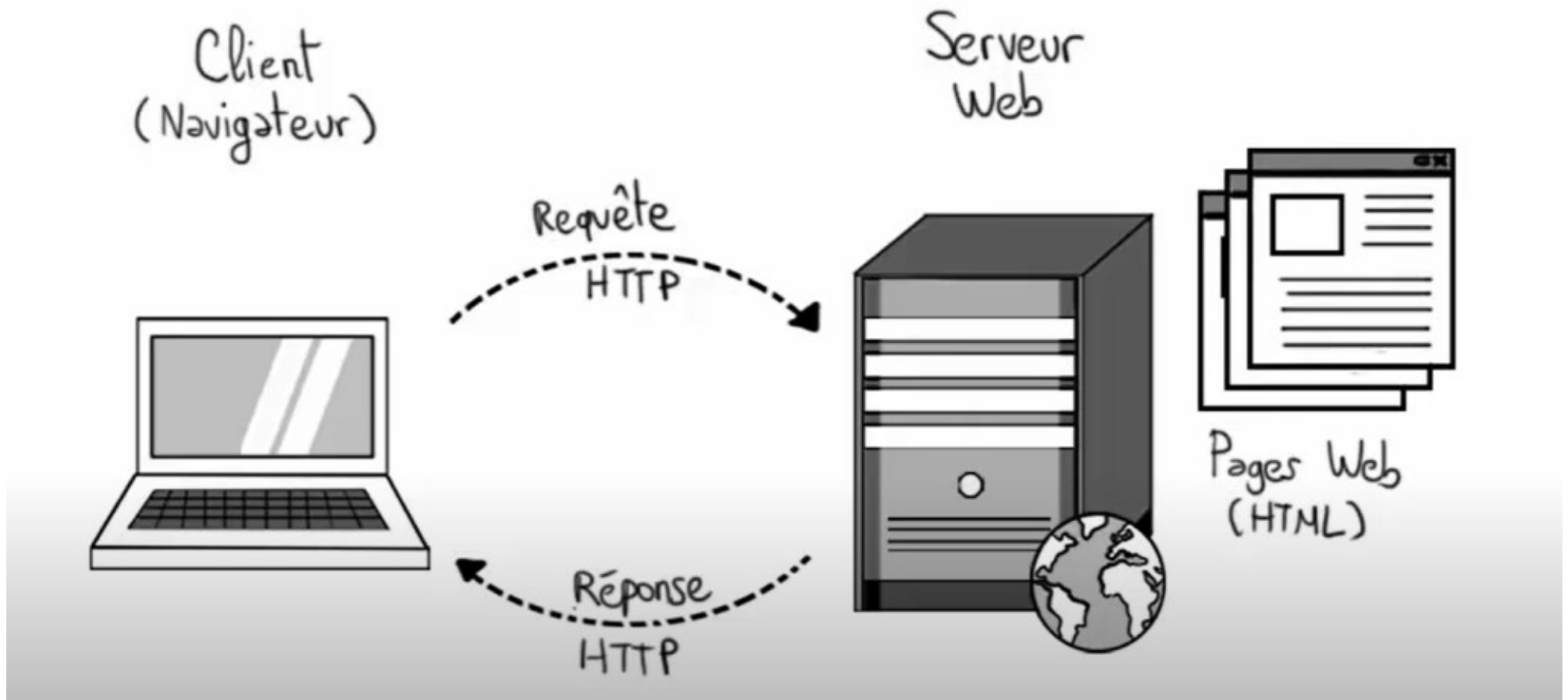
Le navigateur contient des moteurs capables d'interpréter ces différents fichiers et différents langages. Ainsi, une page Web n'est pas un seul fichier, mais une combinaison de ressources téléchargées et assemblées côté client.



Définition et fonctionnement Web

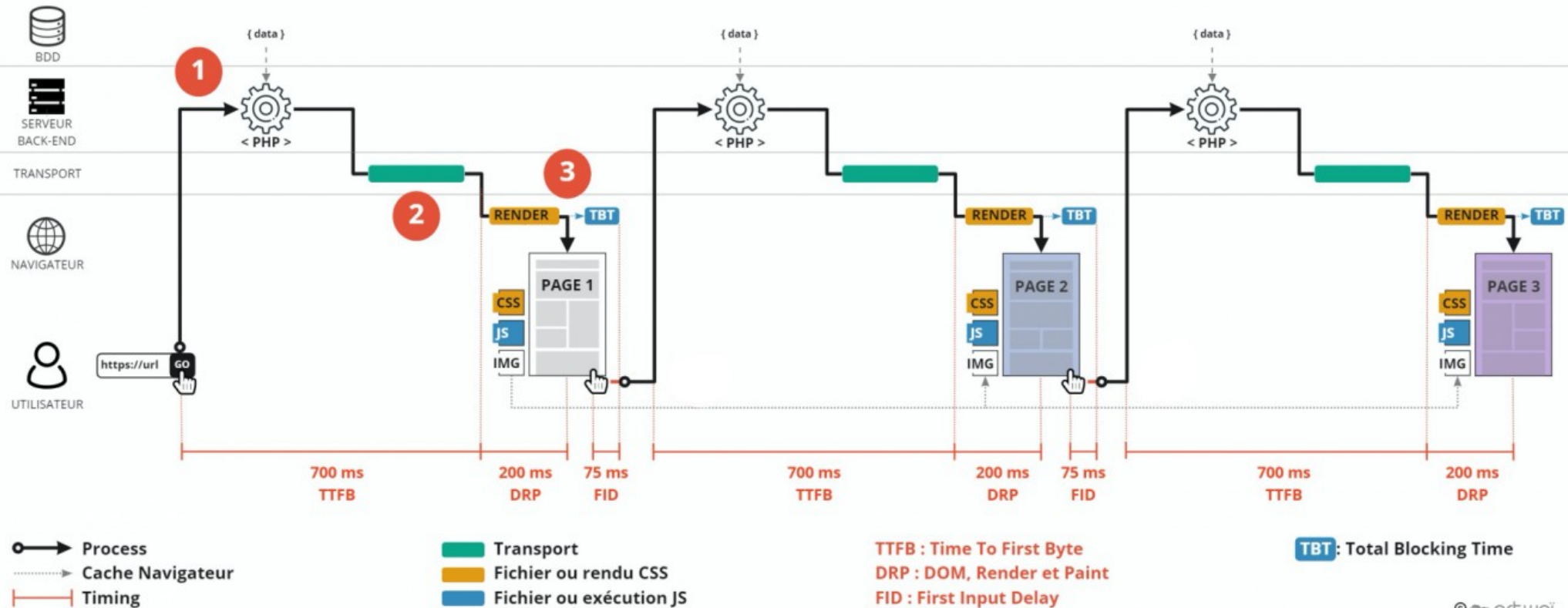
- Le mot Web désigne l'ensemble des données mondiales reliées entre elles par des liens hypertextes. Ces données sont hébergées sur des machines éparses appelées serveurs.
- Le protocole associé HTTP (*HyperText Transfer Protocol*) permet la communication entre un **client** (*navigateur*) et un **serveur** (*machine qui héberge les pages*).
- Inventé entre 1989 et 1991 au CERN par Tim Berners-Lee et Robert Cailliau, le Web repose sur trois piliers :
 - HTTP : protocole d'échange (*définie les rôles client et serveur*),
 - URL : adresse permettant de localiser une ressource,
 - HTML : langage qui décrit la structure des documents.

Définition et fonctionnement Web



<https://www.cpesb.leboucher-maths.fr/1-le-web/les-serveurs-web-3/>

Définition et fonctionnement Web



<https://www.artwai.com/conception-site-web-differentes-approches/>

Plan et objectifs

Fondamentaux du Web

Fondamentaux du CSS

Médias, Tableaux et Formulaires

Responsive Design

Styles avancés et animations

Bonnes pratiques et accessibilité

- Comprendre ce qu'est le Web (*url, hyperdocument, etc*).
- Comprendre la structure minimale d'une page Web (*DOM, doctype, html, head, body*).
- Identifier et utiliser les balises de base (*titres, paragraphes, listes, liens, images*).
- Découvrir les balises sémantiques (*header, nav, article, section, footer*).

Structure minimale d'une page Web

Une page HTML est un document texte structuré. Elle doit respecter une hiérarchie de balises.

Une balise décrit un élément de la page : balise ouvrante, contenu, balise fermante (*Certaines balises sont auto-fermantes*).

Ci-dessous : une balise **parente** (*div*) possédants trois **enfants** :


- `<h1>` : un titre
- `<p>` : un paragraphe
- `` : une image (*auto-fermante*)

```
<div>
  <h1>Bonjour</h1>
  <p>ceci est un paragraphe</p>
  
</div>
```



Bonjour

ceci est un paragraphe

 Logo du site

Structure minimale d'une page Web

4 parties sont essentielles à une page :


- `<!doctype html>` : type de document (HTML5).
- `<html>` : racine du document.
- `<head>` : informations techniques, non visibles (*métadonnées*).
- `<body>` : contenu affiché à l'écran.

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Exemple minimal</title>
  </head>
  <body>
    <div>
      <h1>Bonjour</h1>
      <p>ceci est un paragraphe</p>
      
    </div>
  </body>
</html>
```



Bonjour

ceci est un paragraphe

 Logo du site

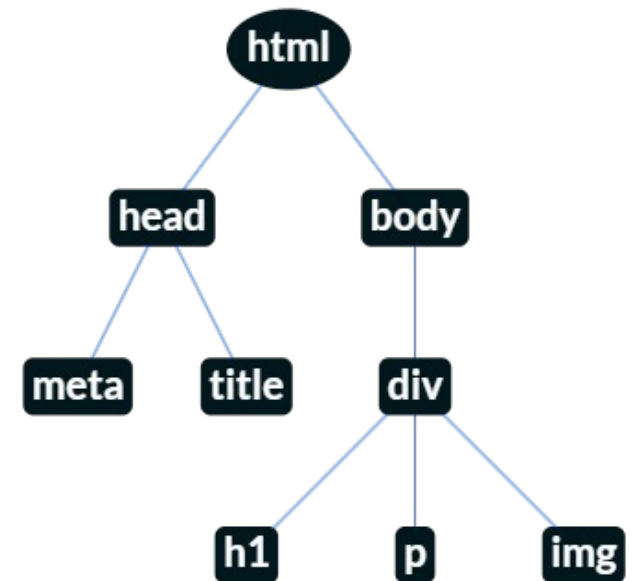
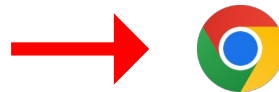
Structure minimale d'une page Web

Le **DOM** (Document Object Model) est la représentation interne d'une page web par le navigateur.

Quand un navigateur reçoit le fichier HTML, il le transforme en un arbre d'objets. Chaque balise, chaque texte, chaque attribut devient un nœud de cet arbre.

Le CSS et le JavaScript interagissent avec le DOM.

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Exemple minimal</title>
  </head>
  <body>
    <div>
      <h1>Bonjour</h1>
      <p>ceci est un paragraphe</p>
      
    </div>
  </body>
</html>
```



Structure minimale d'une page Web

Les attributs en HTML :

- Un attribut fournit une information supplémentaire à une balise.
- Est écrit sous la forme **nom="valeur"** dans la balise ouvrante.
- Certains attributs sont communs à presque toutes les balises (*id*, *class*, *title*), d'autres sont spécifiques.
- Ils servent à configurer les balises, ils sont fondamentaux en HTML

```
<a href="https://exemple.com" title="Aller sur le site">Un lien</a>  
  
<input type="checkbox" checked>
```

Plan et objectifs

Fondamentaux du Web

Fondamentaux du CSS

Médias, Tableaux et Formulaires

Responsive Design

Styles avancés et animations

Bonnes pratiques et accessibilité

- Comprendre ce qu'est le Web (*url, hyperdocument, etc*).
- Comprendre la structure minimale d'une page Web (*DOM, doctype, html, head, body*).
- Identifier et utiliser les balises de base (*titres, paragraphes, listes, liens, images*).
- Découvrir les balises sémantiques (*header, nav, article, section, footer*).

Balises élémentaires : titres

- Les **titres** structurent la page de `<h1>` à `<h6>` :
 - `<h1>` = titre principal (*unique par page, bonne pratique*).
 - `<h2>` à `<h6>` = sous-titres hiérarchiques.
- Cette structure aide :
 - à la navigation (*lecteurs d'écran, outils d'accessibilité, etc*),
 - au référencement SEO (*Search Engine Optimization*).



```
<h1>Titre de niveau 1</h1>
<h2>Titre de niveau 2</h2>
<h3>Titre de niveau 3</h3>
<h4>Titre de niveau 4</h4>
<h5>Titre de niveau 5</h5>
<h6>Titre de niveau 6</h6>
```



Titre de niveau 1

Titre de niveau 2

Titre de niveau 3

Titre de niveau 4

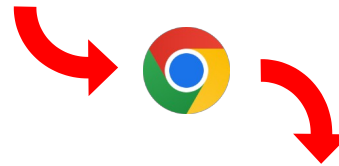
Titre de niveau 5

Titre de niveau 6

Balises élémentaires : paragraphes

- Le paragraphe est créé avec la balise <p>.
- Ajout automatique d'un saut de ligne avant et après. Ne pas utiliser
 par bonnes pratiques.
- Un document peut contenir plusieurs paragraphes.

```
<p>Lorem Ipsum ... specimen book.</p>  
<p>It has survived ... Lorem Ipsum</p>
```



Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Balises élémentaires : listes

- Liste non ordonnée : `` (éléments avec puces).
- Liste ordonnée : `` (éléments numérotés).
- Chaque élément de la liste est un ``.
- `` et `` définissent le type de liste, mais ne sont pas eux-même des éléments affichables.
- Les listes peuvent être imbriquées pour représenter une hiérarchie.

```
<ul>  
  <li>Langages  
    <ol>  
      <li>HTML</li>  
      <li>CSS</li>  
      <li>JavaScript</li>  
    </ol>  
  </li>  
  <li>Outils</li>  
</ul>
```



- Langages
 1. HTML
 2. CSS
 3. JavaScript
- Outils

Balises élémentaires : les liens

- Créés avec la balise `<a>`. L'attribut *href* indique la destination :
 - absolue → vers un site externe.
 - relative → vers une page du même site.
- Possibilité d'ouvrir dans un nouvel onglet avec `target="_blank"`.
- Un lien peut contenir n'importe quelle balise en son sein.

```
<!-- Lien absolu -->
<a href="https://www.example.com">Visiter Example</a>

<!-- Lien relatif -->
<a href="contact.html">Page Contact</a>

<!-- Lien ouvrant un nouvel onglet -->
<a href="guide.pdf" target="_blank">Télécharger le guide</a>
```



[Visiter Example](https://www.example.com) [Page Contact](contact.html) [Télécharger le guide](guide.pdf)

Balises élémentaires : images

- Utiliser la balise auto-fermante ``. Les attributs principaux sont :
 - `src` → chemin du fichier,
 - `alt` → description (*accessibilité, SEO*).

```
<p>Du texte</p>
<!-- Une image valide -->

<!-- Une image invalide -->

```



Plan et objectifs

Fondamentaux du Web

Fondamentaux du CSS

Médias, Tableaux et Formulaires

Responsive Design

Styles avancés et animations

Bonnes pratiques et accessibilité

- Comprendre ce qu'est le Web (*url, hyperdocument, etc*).
- Comprendre la structure minimale d'une page Web (*DOM, doctype, html, head, body*).
- Identifier et utiliser les balises de base (*titres, paragraphes, listes, liens, images*).
- Découvrir les balises sémantiques (*header, nav, article, section, footer*).

Les balises sémantiques

Pourquoi des balises sémantiques ? Les anciennes pages utilisaient surtout `<div>` pour tout organiser. Difficile à comprendre pour les navigateurs, les moteurs de recherche, les technologies d'assistance.

Les balises sémantiques n'ont pas d'impact visuel. Elles apportent du sens, décrivent le rôle de chaque partie de la page.

```
<div>
  <h1>Mon site</h1>
  <div>Menu</div>
</div>

<div>
  <div>
    <h2>Article</h2>
    <p>Texte...</p>
  </div>
</div>

<div>
  © 2025
</div>
```



```
<header>
  <h1>Mon site</h1>
  <nav>Menu</nav>
</header>

<main>
  <article>
    <h2>Article</h2>
    <p>Texte...</p>
  </article>
</main>

<footer>
  © 2025
</footer>
```

Les balises sémantiques

Les balises principales :

- **<header>** : en-tête d'une page / section.
- **<nav>** : zone de navigation (*menus, liens*).
- **<article>** : contenu autonome (*article, post, fiche produit...*).
- **<section>** : encadre un ensemble d'éléments cohérent, thématique.
- **<footer>** : pied d'une page ou d'une section.

```
<header>
  <h1>Actualités</h1>
  <nav>
    <a href="index.html">Accueil</a>
    <a href="contact.html">Contact</a>
  </nav>
</header>

<main>
  <article>
    <h2>Nouvelle version du site</h2>
    <p>Découvrez les nouveautés !</p>
  </article>

  <section>
    <h2>Autres infos</h2>
    <p>Contenu complémentaire...</p>
  </section>
</main>

<footer>
  © 2025 – Mon site
</footer>
```

Exercices 1 et 2

Plan et objectifs

Fondamentaux du Web

Fondamentaux du CSS

Médias, Tableaux et Formulaires

Responsive Design

Styles avancés et animations

Bonnes pratiques et accessibilité

- Comprendre le rôle du CSS
- Les sélecteurs CSS
- Mise en forme simple

Comprendre le rôle du CSS

Le HTML est là pour organiser et structurer le contenu, mais il ne dit rien sur l'apparence.

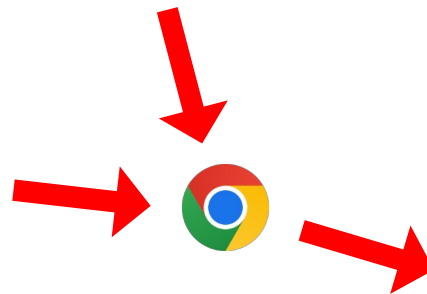
Le CSS a été inventé pour séparer la structure et le style. Grâce au CSS, nous pouvons choisir les couleurs, les polices, les marges, les animations, etc.

Par bonne pratique, le CSS est écrit dans un ou plusieurs fichiers séparés du HTML.

```
<h1 id="titre">Bienvenue</h1>
<p class="intro">Ceci est mon premier site web.</p>
<p class="intro">Le HTML structure le contenu, mais sans style particulier.</p>
```

```
#titre {
  color: darkblue;
}

.intro {
  color: gray;
  font-style: italic;
}
```



Bienvenue

Ceci est mon premier site web.

Le HTML structure le contenu, mais sans style particulier.

Comprendre le rôle du CSS

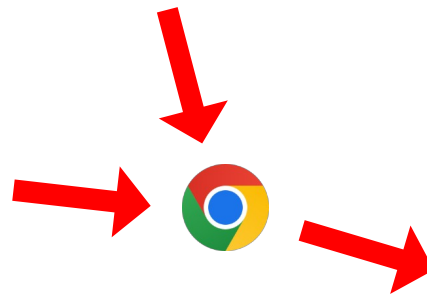
Le CSS répond à plusieurs rôles :

- Mise en forme : couleurs, polices, tailles, espacements.
- Organisation : homogénéité du style sur toutes les pages.
- Responsive design : adapter la présentation selon l'écran (*mobile, tablette, PC*)

```
<h1 id="titre">Bienvenue</h1>
<p class="intro">Ceci est mon premier site web.</p>
<p class="intro">Le HTML structure le contenu, mais sans style particulier.</p>
```

```
#titre {
  color: darkblue;
}

.intro {
  color: gray;
  font-style: italic;
}
```



Bienvenue

Ceci est mon premier site web.

Le HTML structure le contenu, mais sans style particulier.

Comprendre le rôle du CSS

Une règle CSS associe un sélecteur à un ensemble de propriétés.

- **Sélecteur** → indique à quels éléments HTML la règle s'applique.
- **Propriétés** → définissent quoi modifier (*couleur, police, taille, etc*).
- **Valeurs** → précisent comment appliquer la propriété.

```
#titre {  
  color: darkblue;      /* propriété : color → valeur : darkblue */  
}  
  
.intro {  
  color: gray;          /* propriété : color → valeur : gray */  
  font-style: italic;   /* propriété : font-style → valeur : italic */  
}
```

Le CSS contient des centaines de propriétés. Il est donc essentiel d'avoir une bonne base de connaissances, et de savoir se documenter rapidement (*MDN Web Docs, W3C...*). Le bon développeur web n'est pas celui qui connaît tout par cœur, mais celui qui sait chercher et appliquer la bonne propriété au bon moment.

Plan et objectifs

Fondamentaux du Web

Fondamentaux du CSS

Médias, Tableaux et Formulaires

Responsive Design

Styles avancés et animations

Bonnes pratiques et accessibilité

- Comprendre le rôle du CSS
- Les sélecteurs CSS
- Mise en forme simple

Sélecteurs

Élément ciblé	syntaxe	Description
Type	balise	Cible toutes les balises d'un type donné (ex. tous les <p>).
id	#id	Cible un élément unique, identifié par son id.
class	.class	Cible tous les éléments regroupés par une même class.
Descendant	parent enfant	Cible tous les éléments contenus dans un autre.
Enfant direct	parent > enfant	Cible uniquement les enfants immédiats d'un élément parent.
Frère adjacent	frère1 + frère2	Cible le frère2 placé juste après un frère1.
Frères généraux	frère1 ~ frère2	Cible tous les frère2 qui suivent un frère1.
Tout élément	*	Cible tous les éléments de la page (rarement utilisé seul).
Premier enfant	parent:first-child	Cible un élément s'il est le premier enfant de son parent.
Dernier enfant	parent:last-child	Cible un élément s'il est le dernier enfant de son parent

Sélecteurs

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>Exemple Sélecteurs</title>
</head>
<body>

  <h1>Ma liste de courses</h1>
  <p>Ce paragraphe suit directement le titre.</p>
  <p>Celui-ci ne sera pas affecté.</p>

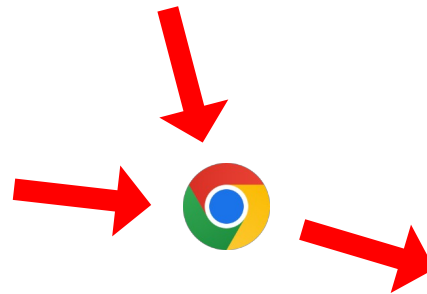
  <ul>
    <li>Pâtes</li>
    <li>Riz</li>
    <li>Pommes</li>
  </ul>

</body>
</html>
```

```
li {
  list-style: square;
}

h1 + p {
  font-weight: bold;
  color: darkblue;
}

li:last-child {
  color: green;
}
```



Ma liste de courses

Ce paragraphe suit directement le titre.

Celui-ci ne sera pas affecté.

- Pâtes
- Riz
- Pommes

Plan et objectifs

Fondamentaux du Web

Fondamentaux du CSS

Médias, Tableaux et Formulaires

Responsive Design

Styles avancés et animations

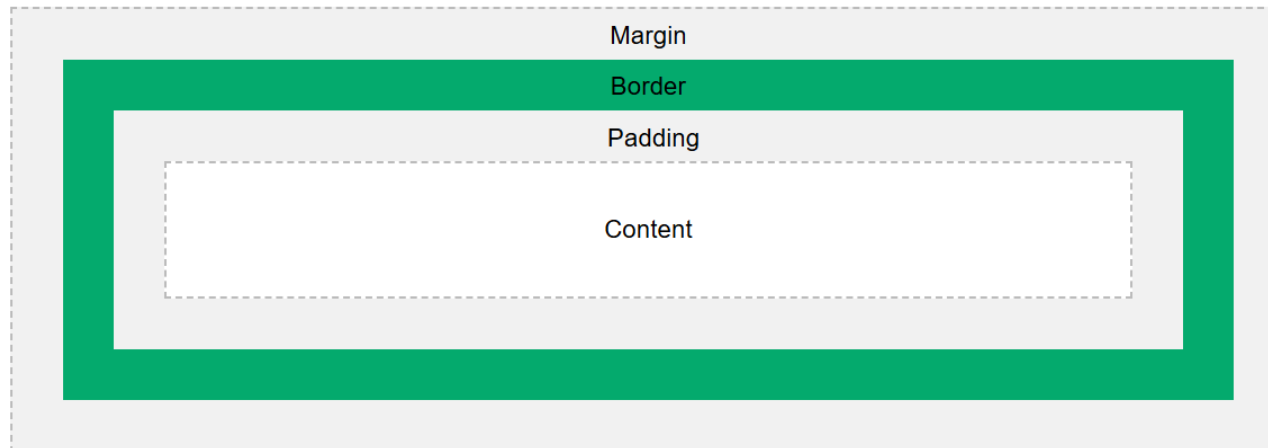
Bonnes pratiques et accessibilité

- Comprendre le rôle du CSS
- Les sélecteurs CSS
- Mise en forme simple

Mise en forme

Chaque élément HTML est une boîte rectangulaire composée de 4 zones :

- **content** → élément lui même.
- **padding** → espace entre le contenu et la bordure.
- **border** → cadre autour de l'élément.
- **margin** → espace autour de la boîte, séparant des autres éléments.



- $\text{width} + \text{padding} + \text{border} = \text{vraie largeur d'un élément}$
- $\text{height} + \text{padding} + \text{border} = \text{vraie hauteur d'un élément}$

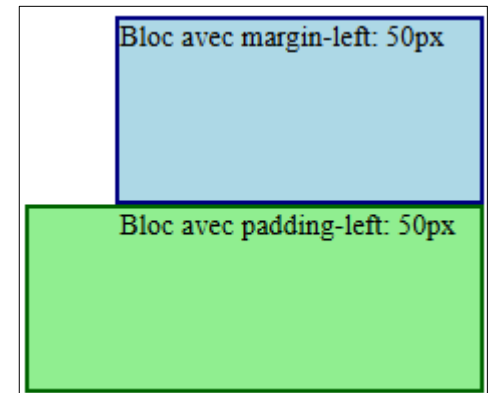
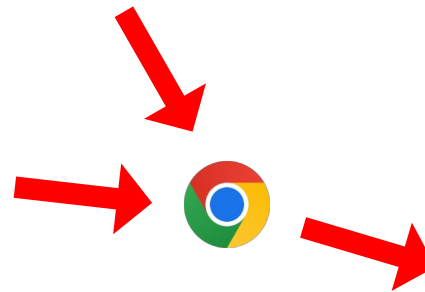
Mise en forme



```
<div class="marge">Bloc avec margin-left: 50px</div>  
<div class="padding">Bloc avec padding-left: 50px</div>
```



```
.marge {  
  background-color: lightblue;  
  border: 3px solid navy;  
  margin-left: 50px;  
  width: 200px;  
  height: 100px;  
}  
  
.padding {  
  background-color: lightgreen;  
  border: 3px solid darkgreen;  
  padding-left: 50px;  
  width: 200px;  
  height: 100px;  
}
```

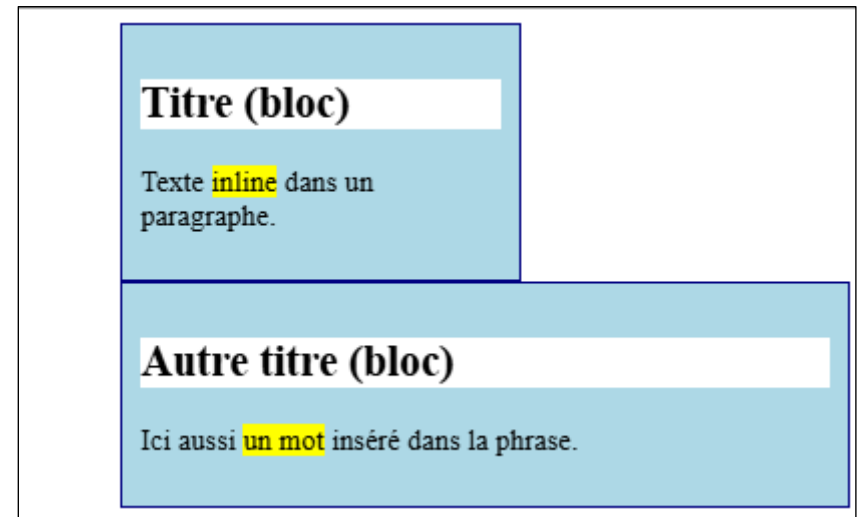


Mise en forme

Un élément bloc commence sur une nouvelle ligne et occupe toute la largeur disponible que lui offre son parent (ex : `<div>`, `<p>`, `<h1>`). Un élément en ligne s'insère dans le flux du texte sans retour à la ligne (ex : ``, `<a>`, ``).

Constatez le padding interne du bloc bleu. Les `<p>` ne débordent pas par dessus.

```
.bloc {  
  border: 2px solid navy;  
  padding: 10px;  
  background-color: lightblue;  
  margin-left : 50px;  
}  
  
#bloc1 {  
  width: 200px;  
}  
  
h2 {  
  background-color: white;  
}  
  
span {  
  background-color: yellow;  
}
```



Mise en forme

- Modifier la couleur, la taille, la police et le style du texte :

```
h1 {  
  color: darkred;  
  font-family: Arial, sans-serif;  
  font-size: 24px;  
  font-weight: bold;  
  font-style: italic;  
}
```



Bienvenue sur mon site

- Arrière-plans et bordures :

```
div {  
  background-color: lightblue;  
  border: 3px solid navy;  
  border-radius: 10px;  
  width : 200px;  
}
```



Ceci est un bloc avec un fond
coloré et une bordure arrondie.

Mise en forme

Espaces et alignement :

```
p {  
  margin: 20px;  
  padding: 10px;  
  text-align: justify;  
  line-height: 2;  
  border : 1px solid red;  
  width : 300px;  
}
```



Voici un paragraphe plus long pour voir l'effet des marges, du padding, de l'alignement et de l'interlignage. Le texte est justifié pour que les bords gauche et droit soient alignés.

Listes et décorations (*souligner, surligner, etc*) :

```
ul {  
  list-style-type: square;  
}  
  
a {  
  text-decoration: none;  
  color: darkgreen;  
}
```



- Premier élément
- Deuxième élément
- Troisième élément

Voici un [lien vers une ressource](#).

Exercices 3

Plan et objectifs

Fondamentaux du Web

Fondamentaux du CSS

Médias, Tableaux et Formulaires

Responsive Design

Styles avancés et animations

Bonnes pratiques et accessibilité

- Introduction aux médias
- Les tableaux
- Formulaires