



RÉSUMÉ THÉORIQUE – FILIÈRE DÉVELOPPEMENT DIGITAL

M104 - DÉVELOPPER DES SITES WEB STATIQUES



100 heures



SOMMAIRE



1. Créer une page web en HTML

Introduction au langage HTML

Utiliser l'environnement de développement

Définir des éléments basiques d'une page HTML

2. Implémenter une page web statique

Réaliser une page web statique

Intégrer des formulaires dans une page web

3. Mettre en forme une page web avec les feuilles de style CSS

Introduire le CSS

Utiliser les propriétés CSS

Manipuler le positionnement

Adapter une page web au dispositif d'affichage

Créer des animations

Adapter des templates HTML/CSS avec un site Web

4. Maîtriser Bootstrap

Intégrer Bootstrap

Maitriser les classes CSS de base

Utiliser les composants Bootstrap

5. Héberger un site Web

Déployer en intranet

Déployer en ligne

MODALITÉS PÉDAGOGIQUES



1

LE GUIDE DE SOUTIEN

Il contient le résumé théorique et le manuel des travaux pratiques



2

LA VERSION PDF

Une version PDF est mise en ligne sur l'espace apprenant et formateur de la plateforme WebForce Life



3

DES CONTENUS TÉLÉCHARGEABLES

Les fiches de résumés ou des exercices sont téléchargeables sur WebForce Life



4

DU CONTENU INTERACTIF

Vous disposez de contenus interactifs sous forme d'exercices et de cours à utiliser sur WebForce Life



5

DES RESSOURCES EN LIGNES

Les ressources sont consultables en synchrone et en asynchrone pour s'adapter au rythme de l'apprentissage



PARTIE 1

Créer une page web en HTML

Dans ce module, vous allez :

- Découvrir l'historique du web et du langage HTML
- Manipuler les outils de création d'une page web
- Connaître et manipuler les éléments de base d'une page web



 20 heures



CHAPITRE 1

Introduction au langage HTML

Ce que vous allez apprendre dans ce chapitre :

- L'historique de l'HTML
- La mission du W3C
- L'utilisation des navigateurs



02 heures



CHAPITRE 1

Introduction au langage HTML

1. Historique de l'HTML
2. Rôle du W3C
3. Propriétés des navigateurs

Définitions

- Le **World Wide Web (WWW)**, abrégé le **Web**, est le principal service de partage et de recherche d'informations sur **Internet** (le réseau informatique mondial) [1].
- Le **Web** donne aux utilisateurs l'accès à une vaste gamme de pages et de documents (appelés **ressources**) qui sont connectés au moyen de **liens hypertexte** (hyperliens) et accessibles via un **navigateur**.
- Les **ressources** du web correspondent aux entités informatiques comme le texte, les images, les vidéos, d'autres ressources.
- Pour chaque ressource sur le web, une adresse en ligne appelée **URL** (Uniform Resource Locator) est attribuée.
- Un **document hypertexte** (composé du texte et des liens hypertextes) est écrit en langage de balisage hypertexte (**HTML**).
- Le Web fonctionne dans le mode client-serveur :
 - Les **serveurs** sont des machines (ordinateurs) qui stockent et transmettent des documents à d'autres ordinateurs sur le réseau
 - Les **clients** sont aussi des machines, sur lesquelles sont installés des programmes qui demandent des documents à un serveur
 - Le **navigateur** est le logiciel qui permet aux utilisateurs de visualiser les documents récupérés. Parmi les navigateurs les plus utilisés on cite : Google Chrome, Internet Explorer, Mozilla Firefox, Opera...



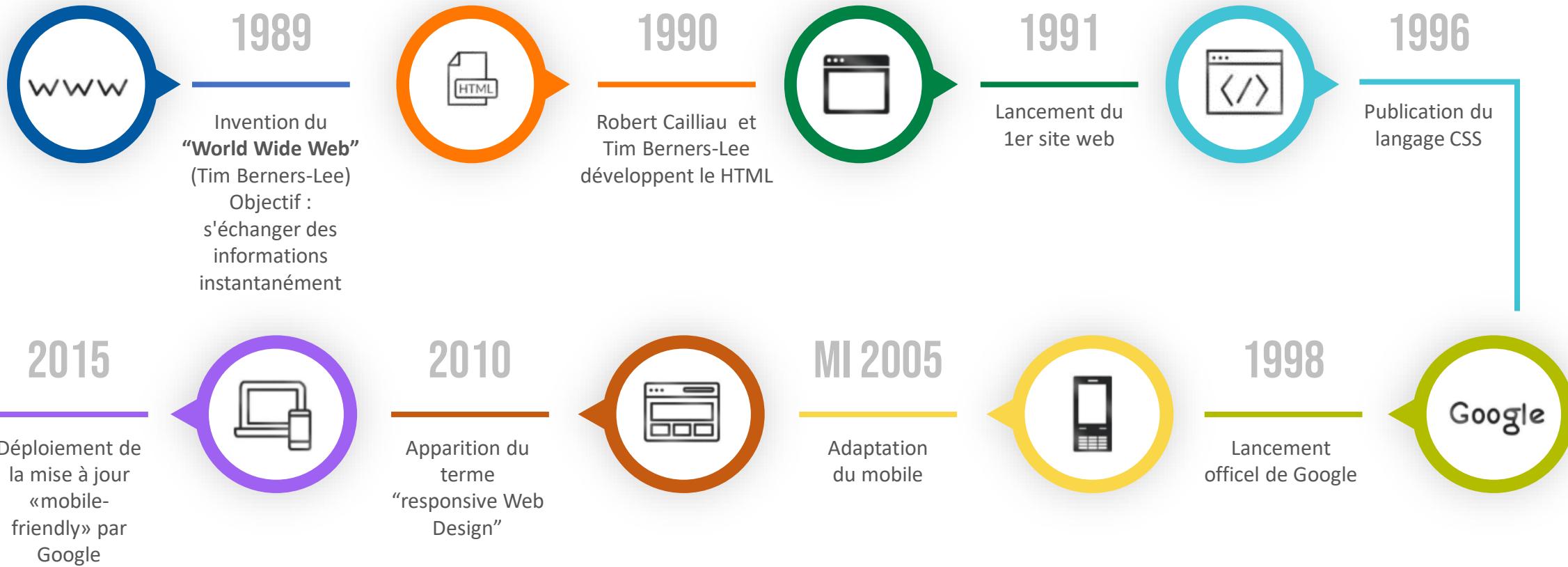
Figure 1 : les navigateurs les plus utilisés

[1] <http://home.cern/fr/science/computing/birth-web>

01 - Introduction au langage HTML

Historique de l'HTML

Le web a connu une grande évolution depuis son apparition :



Source : <https://www.wddonline.com/2016/09/27/history-world-wide-web/>

01 - Introduction au langage HTML

Historique de l'HTML

Évolution Langage HTML

- **HTML1 (1990 - 1992)** : Première publication du langage HTML
- **1994 : Apports de Netscape Navigator** : support de nombreux éléments de présentation est ajouté : attributs de texte, clignotement, centrage, etc
- **HTML 2.0 (1995 - 1996)**
- **HTML 3.2 et 4.0 (1997)** : Le 14 janvier 1997, le W3C publie la spécification HTML 3.2. Ses plus importantes nouveautés sont la standardisation des tables et de nombreux éléments de présentation. Le 18 décembre 1997, le W3C publie la spécification HTML 4.0 qui standardise de nombreuses extensions supportant les styles et les scripts, les cadres (frames) et les objets (inclusion généralisée de contenu).
- **XHTML (2000- 2006)** : Le développement de HTML en tant qu'application du Standard Generalized Markup Language(SGML) est officiellement abandonné au profit de XHTML, application de Extensible Markup Language(XML).
- **HTML 5 (2007 à nos jours)** : la dernière version du HTML.



Figure 2 : Logo du langage HTML5



CHAPITRE 1

Introduction au langage HTML

1. Historique de l'HTML
2. **Rôle du W3C**
3. Propriétés des navigateurs

01 - Introduction au langage HTML

Rôle du W3C

Le **W3C** (World Wide Web Consortium) est un organisme international à but non lucratif.

Sa mission est de définir les standards techniques liés au web pour assurer l'accessibilité, l'interopérabilité et la pérennité et des documents Web [2]

La valeur sociale du Web vient du fait qu'il permet la communication et le partage des connaissances entre les humains. L'un des principaux objectifs du W3C est de mettre ces avantages à la disposition de tous, quels que soient leur infrastructure matérielle et logicielle, et leur langue. Donc, tous les sites web doivent respecter une norme afin de garantir une meilleure cohérence de données, de formats et de fonctionnalités [3]

En l'absence des standards et des normes, chaque navigateur web aura ses propres règles d'interprétation des données Web. Or, le rôle du W3C est de standardiser les technologies du web et mettre en accord tous les navigateurs ainsi que les développeurs sur un ensemble de règles à respecter par tout le monde.



Figure 3 : logo du W3C

[2] <https://www.w3.org/>

[3] <https://www.w3.org/Consortium/mission#principles>



CHAPITRE 1

Introduction au langage HTML

1. Historique de l'HTML
2. Rôle du W3C
- 3. Propriétés des navigateurs**

Le rôle d'un navigateur

Récupérer le code (généralement écrit en HTML, PHP, JavaScript ou d'autres langages de programmation Web), à partir du **serveur**, puis **l'interpréter** et le **restituer** sous forme de page Web graphique pour l'affichage.

Interpréter l'adresse d'un site web (URL) et gérer les requêtes client / Serveur.

Faciliter le développement et le test des pages Web.

Le fonctionnement d'un navigateur

- En utilisant le **navigateur Web**, l'utilisateur envoie une requête dite "**Requête HTTP**" au serveur qui héberge le site web.
- Le **serveur Web** reçoit la requête de l'utilisateur qui contient l'adresse IP du serveur, le port sur lequel tourne le serveur Web, la ressource demandée et la méthode avec laquelle on exploite cette ressource.
- Le navigateur reçoit la page web demandée sous forme de code appelé : "**code HTML**" et l'interprète pour afficher des objets visuels.

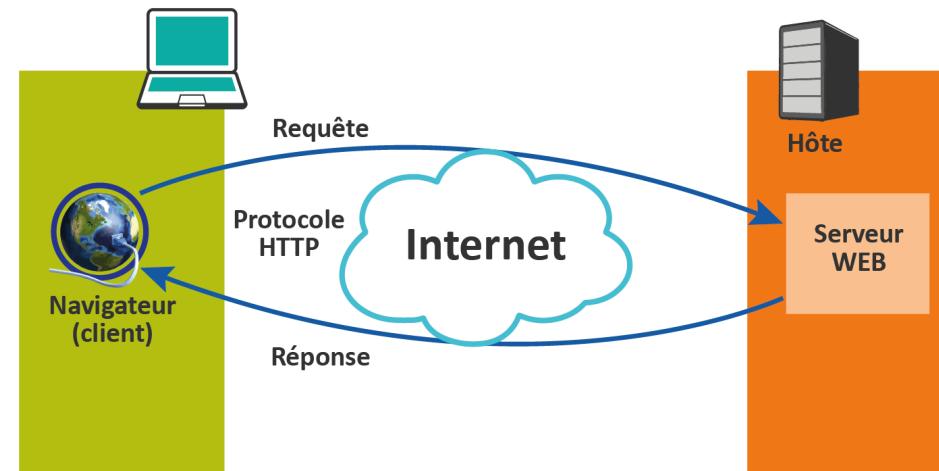
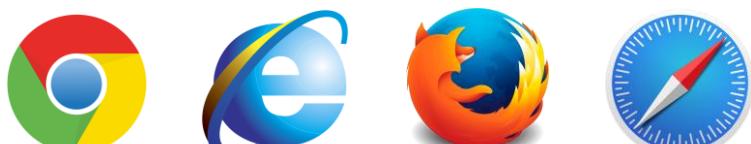


Figure 4 : Communication client-serveur web en HTTP [4]

[4] <https://delmas-rigoutsos.nom.fr/documents/YDelmas-ArchiWeb/YDelmas-ArchiWeb.html>



CHAPITRE 2

Utiliser l'environnement de développement pour produire du HTML

Ce que vous allez apprendre dans ce chapitre :

- Utiliser Visual Studio Code comme éditeur de code
- Installer des extensions de VS Code



01 heure



CHAPITRE 2

Utiliser l'environnement de développement pour produire du HTML

1. Utilisation de Visual Studio Code
2. Installation des extensions de VS Code

02 - Utiliser l'environnement de développement pour produire du HTML

Utilisation de Visual Studio Code

Le logiciel Visual studio Code

Visual Studio Code est un logiciel gratuit développé par Microsoft. Il fonctionne sous différents systèmes : Windows, Linus, Mac OS.

VS Code permet l'édition, la correction, le débogage et l'exécution du code source dans plusieurs langages informatiques : Visual Basic, JavaScript, XML, Python, HTML, CSS,

VS code offre :

- Une présentation sophistiquée du code.
- Une auto-complétion du code source.
- Un ensemble d'extensions permettant de simplifier la programmation.
- La détection du langage de programmation par l'extension du fichier.



Figure 5 : Logo du logiciel VS Code [5]

[5] <https://code.visualstudio.com/>

02 - Utiliser l'environnement de développement pour produire du HTML

Utilisation de Visual Studio Code

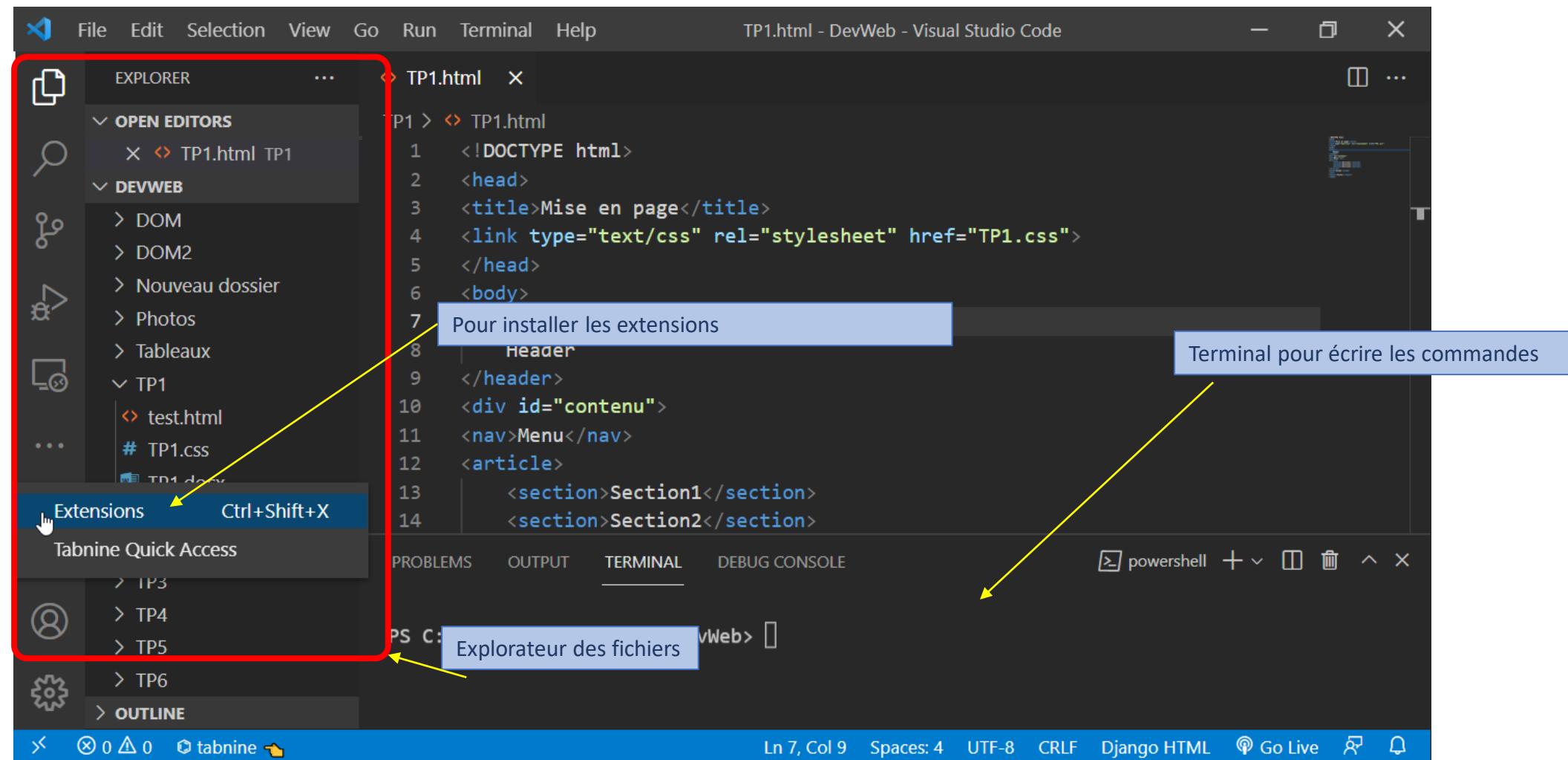


Figure 6 : Environnement de travail de VS Code



CHAPITRE 2

Utiliser l'environnement de développement pour produire du HTML

1. Utilisation de Visual Studio Code
2. Installation des extensions de VS Code

02 - Utiliser l'environnement de développement pour produire du HTML

Utilisation des extensions VS Code



Les extensions de VS Code

Les extensions de Visual Studio Code permettent d'accélérer le développement et favorisent le partage entre les membres de l'équipe de développement. Le modèle d'extensibilité riche de VS Code permet aux «auteurs d'extensions» ainsi qu'aux utilisateurs d'intégrer facilement à l'environnement de développement de VS code.

- Parmi les extensions connues de VS code [6] :
 - **Bracket Pair Colorizer** : Cette extension permet d'identifier la correspondance entre des caractères et des couleurs.
 - **Better Comments** : donne la possibilité de personnaliser la police et l'opacité des commentaires
 - **Snippets** : Un snippet est un modèle qui peut être inséré dans un document. Il est inséré par une commande ou par un texte déclencheur.
 - **CSS Peak** : Permet de visualiser les ID et les classes CSS dans les fichiers html à partir des fichiers CSS respectifs
 - **GitLens** : permet de découvrir rapidement qui, pourquoi et quand une ligne ou un bloc de code a été modifié. Autrement dit, permet de remonter dans l'histoire pour mieux comprendre comment et pourquoi le code a évolué.

[6] <https://marketplace.visualstudio.com/VSCodium>

02 - Utiliser l'environnement de développement pour produire du HTML

Utilisation des extensions VS Code

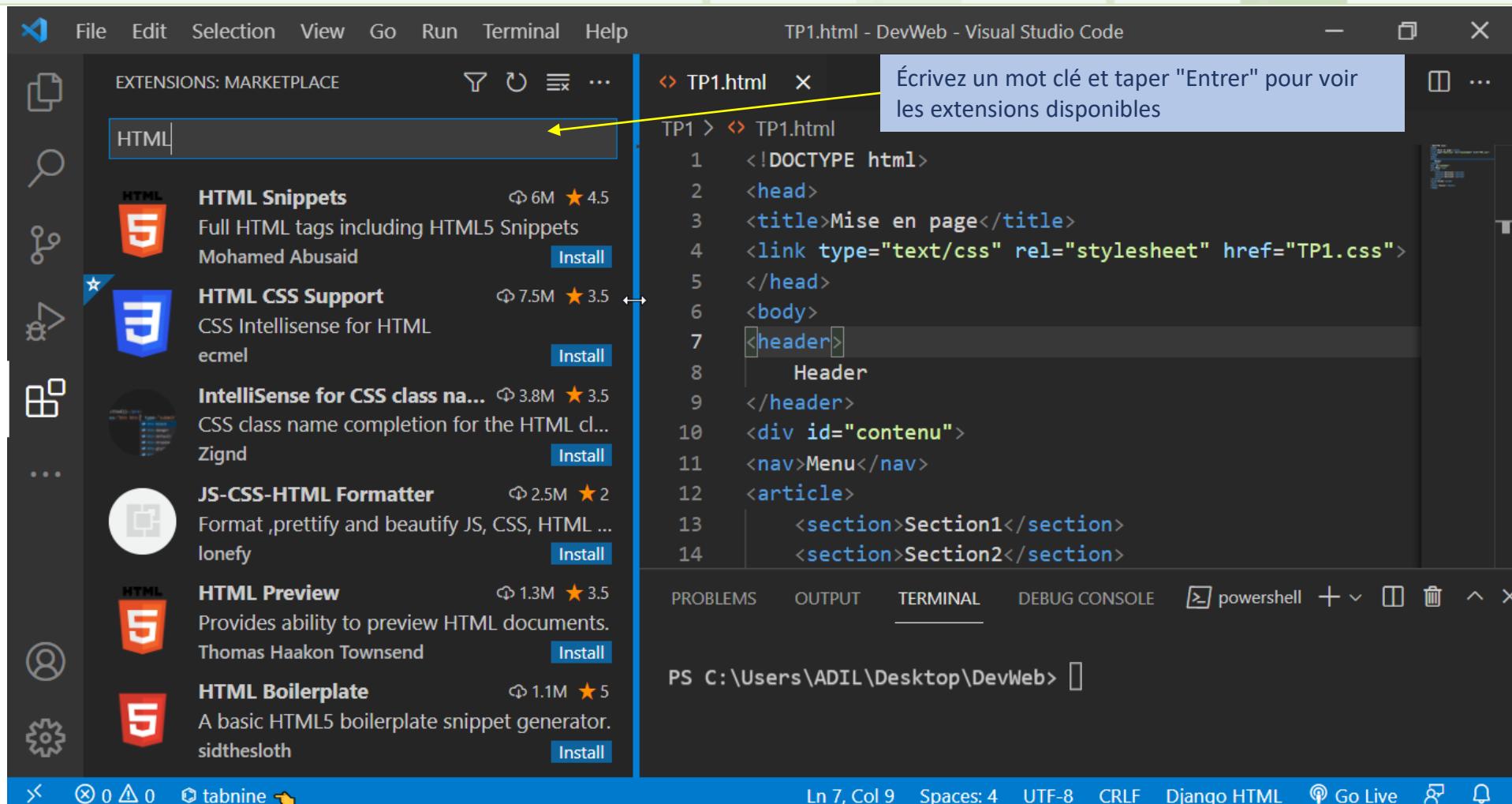


Figure 7 : Installation des extensions dans VS Code

CHAPITRE 3

Définir les éléments basiques d'une page HTML

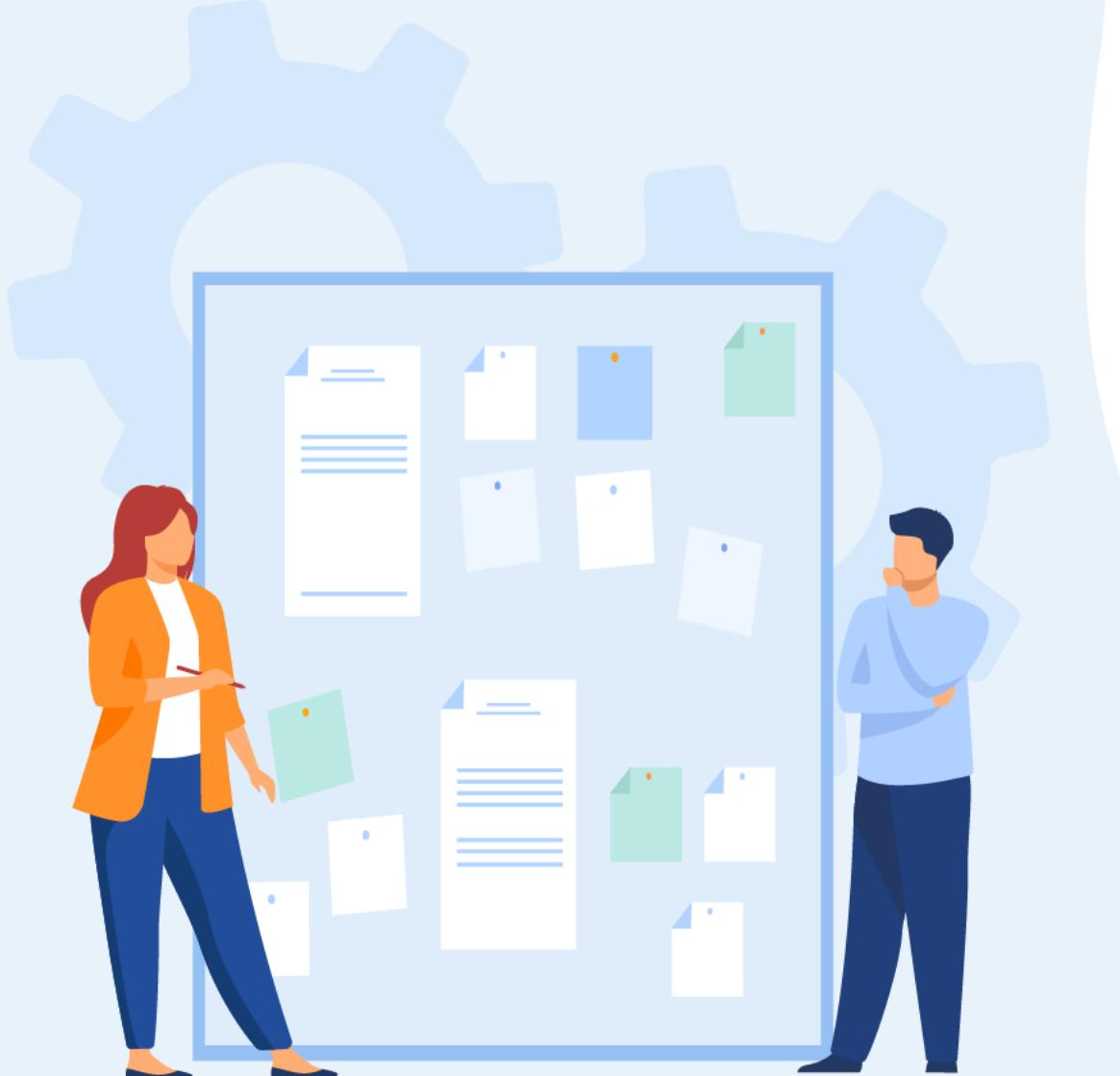


Ce que vous allez apprendre dans ce chapitre :

- Les balises HTML permettant de concevoir une page Web statique
- Les balises de structuration des pages web



17 heures



CHAPITRE 3

Définir les éléments basiques d'une page HTML

1. Normes W3C
2. Squelette d'une page HTML
3. Balises de bases (contenu)
4. Couleurs
5. Listes en HTML
6. Types de chemin d'accès
7. Liens en HTML
8. Balises multimédia (images, audio, vidéo)
9. Tableaux en HTML
10. Balises de structuration

03 - Définir les éléments basiques d'une page HTML

Normes W3C



WEBFORCE
BE THE CHANGE

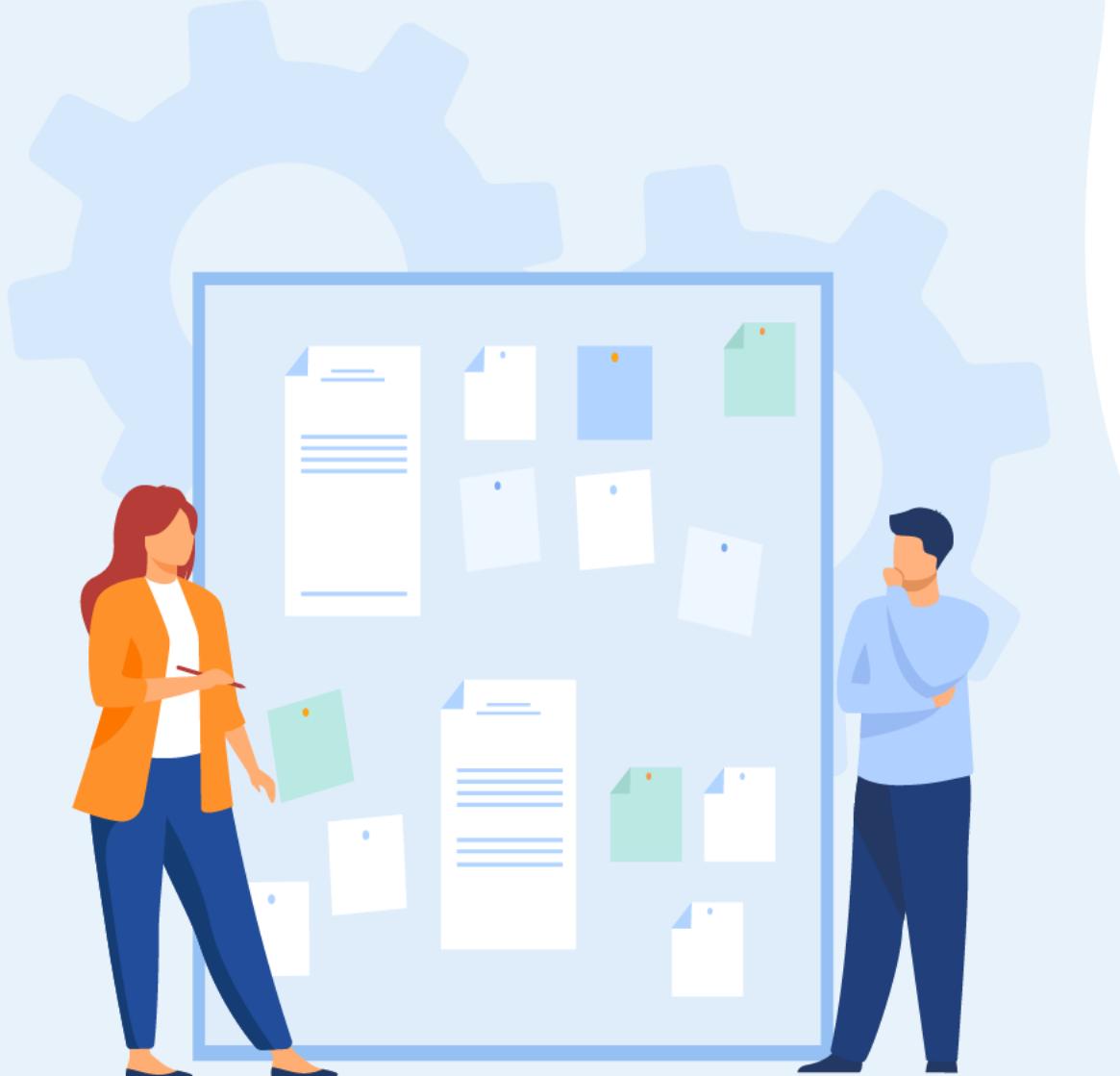
Les normes du W3C

W3C est chargé de promouvoir la compatibilité des technologies du World Wide Web telles que HTML, XML, RDF, CSS, PNG, SVG, ...

Parmi les normes W3C concernant HTML on cite :

- Les balises <html>, <head>, <title>, et <body> sont obligatoires
- Les éléments doivent être correctement imbriqués
- Les éléments doivent toujours être fermés
- Les éléments HTML doivent être en minuscules
- Les documents HTML doivent avoir un élément racine
- Les noms d'attributs doivent être en minuscules
- Les valeurs des attributs des balises doivent être indiquées

Validation du code HTML avec **W3C Validator** : <http://validator.w3.org/>



CHAPITRE 3

Définir les éléments basiques d'une page HTML

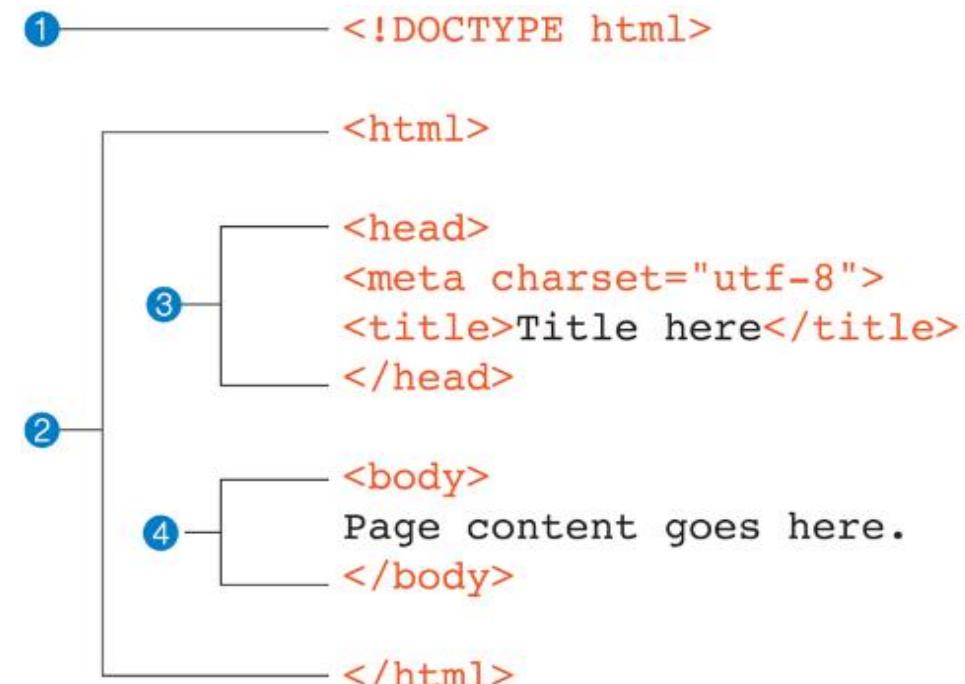
1. Normes W3C
2. **Squelette d'une page HTML**
3. Balises de bases (contenu)
4. Couleurs
5. Listes en HTML
6. Types de chemin d'accès
7. Liens en HTML
8. Balises multimédia (images, audio, vidéo)
9. Tableaux en HTML
10. Balises de structuration

03 - Définir les éléments basiques d'une page HTML

Squelette d'une page HTML

Squelette d'une page Web

① Déclaration du type de document (HTML 5)



② Élément racine qui contient tous les éléments du document HTML

③ Contient des informations descriptives sur le document

④ Contient tout ce que nous voulons afficher dans la fenêtre du navigateur

Figure 8 : Structure minimale d'un document html

03 - Définir les éléments basiques d'une page HTML

Squelette d'une page HTML

Notion de balise

- Une balise est un élément de texte encadrée par les caractères inférieur (<) et supérieur (>)
- Une balise correspond à un élément du code HTML d'une page web (un objet, une mise en forme, ..)
- Une balise a généralement la forme suivante :

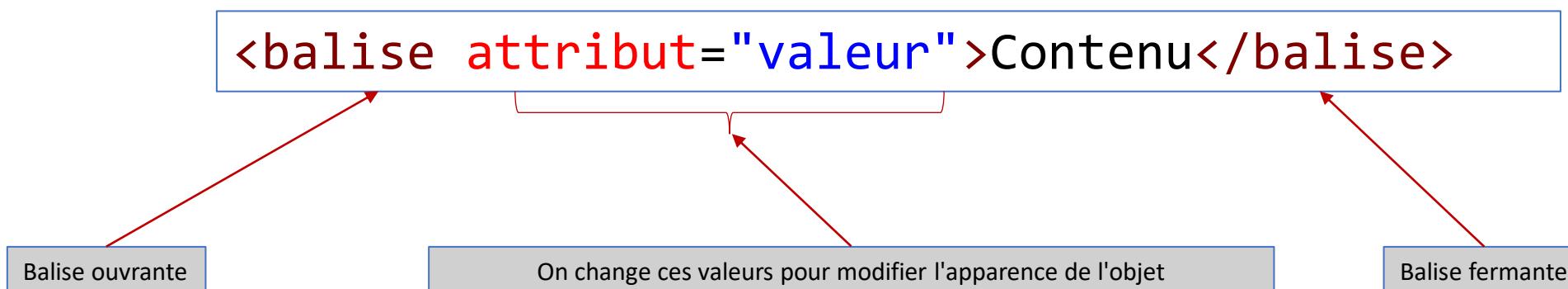


Figure 9 : Forme d'une balise HTML

03 - Définir les éléments basiques d'une page HTML

Squelette d'une page HTML

Notion d'encapsulation

Les balises doivent être correctement encapsulées :

```
<p>
  <b>
    Paragraphe
  </b>
</p>
```

Exemple 1 : Code correct

```
<p>
  <b>
    Paragraphe
  </p>
</b>
```

Exemple 2 : Code incorrect

```
<div>
  <h1>titre</h1>
  <h2>un sous titre</h2>
  <p>un paragraphe avec
    <em>des mots importants</em>
    pour terminer avec une image
    
  </p>
</div>
```

Exemple 3 : Code correct

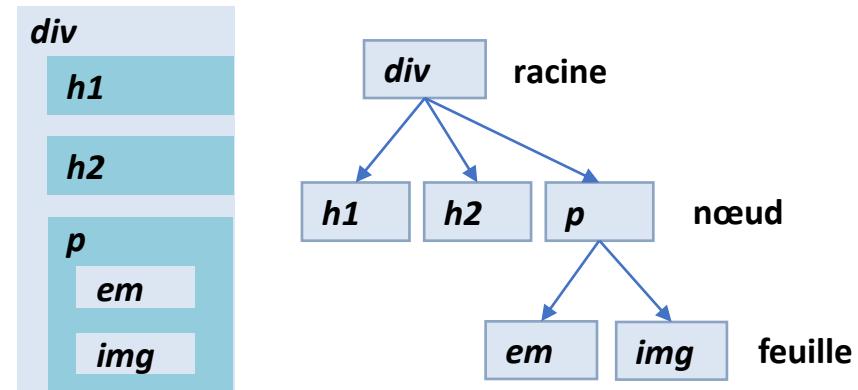


Figure 10 : Encapsulation des balises HTML

03 - Définir les éléments basiques d'une page HTML

Squelette d'une page HTML



Les types d'éléments HTML

Il existe deux types d'éléments : bloc (**bloc**) et en ligne (**inline**)

Les éléments « **bloc** » s'étendent en largeur pour remplir totalement l'espace offert par son conteneur. Par défaut, la boîte devient aussi large que son conteneur, occupant 100% de l'espace disponible. Le nouvel élément bloc commence dans une nouvelle ligne.

Les éléments « **inline** » se suivent en ligne

Bloc

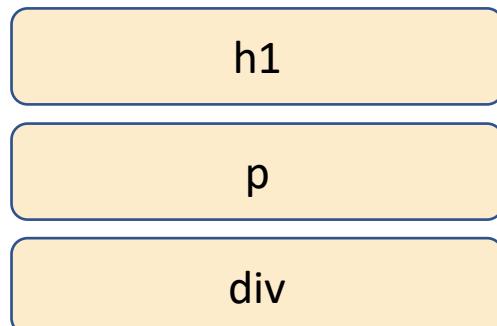


Figure 11 : Exemples d'éléments « bloc »

Inline

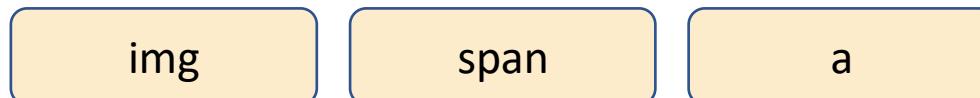


Figure 12 : Exemples d'éléments « inline »

03 - Définir les éléments basiques d'une page HTML

Squelette d'une page HTML



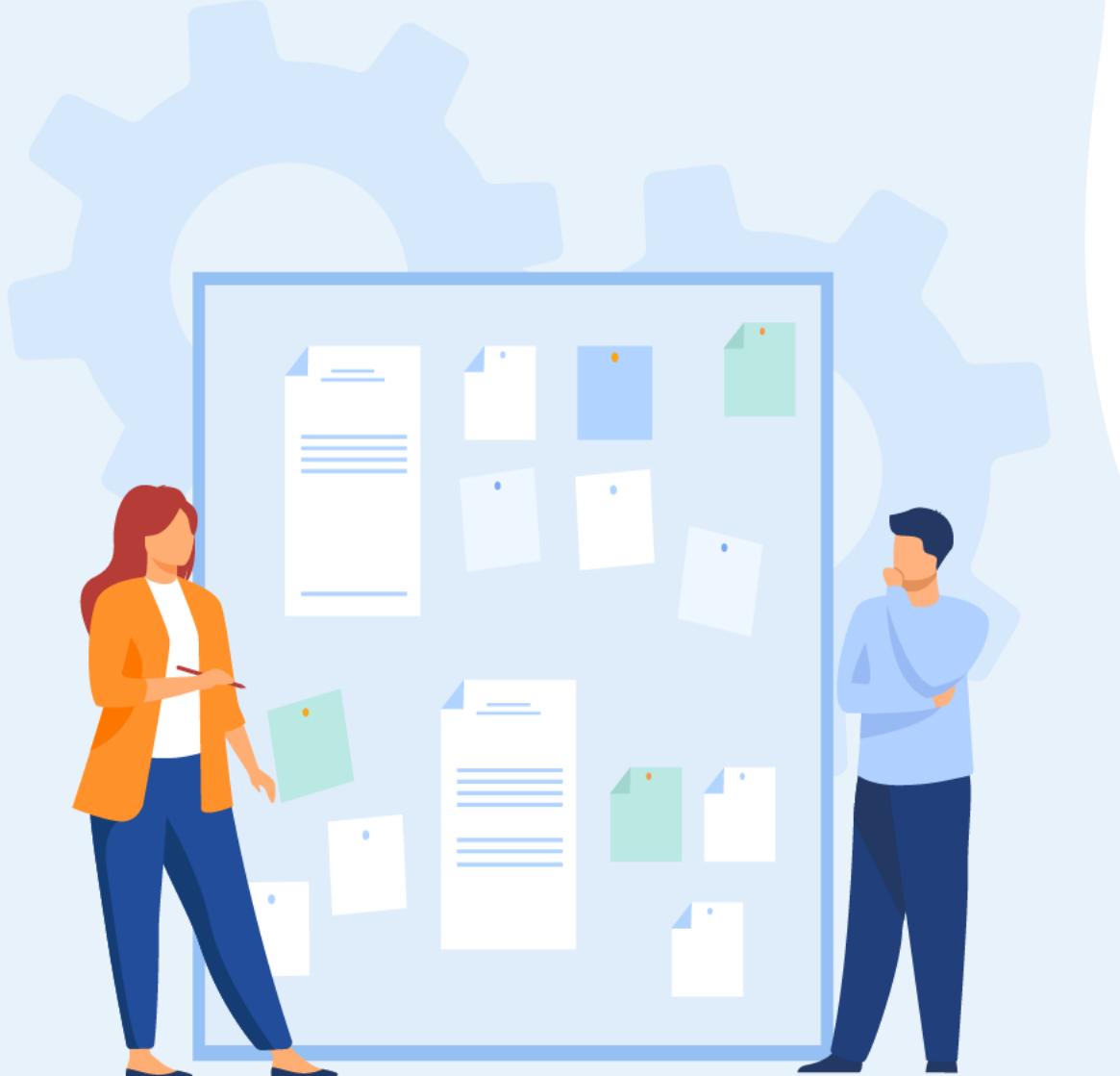
Les types d'éléments HTML

Éléments de niveau bloc en HTML :

<address>	<article>	<aside>	<blockquote>	<canvas>	<dd>	<div>
<dl>	<dt>	<fieldset>	<figcaption>	<figure>	<footer>	<form>
<h1> - <H6>	<header>	<hr>		<main>	<nav>	<noscript>
	<p>	<pre>	<section>	<table>	<tfoot>	
<video>						

Éléments de niveau inline en HTML :

<a>	<abbr>	<acronym>		<bdo>	<big>	
<button>	<cite>	<code>	<dfn>		<i>	
<input>	<kbd>	<label>	<map>	<object>	<output>	<q>
<samp>	<script>	<select>	<small>			<sub>
<sup>	<textarea>	<time>	<tt>	<var>		



CHAPITRE 3

Définir les éléments basiques d'une page HTML

1. Normes W3C
2. Squelette d'une page HTML
- 3. Balises de bases (contenu)**
4. Couleurs
5. Listes en HTML
6. Types de chemin d'accès
7. Liens en HTML
8. Balises multimédia (images, audio, vidéo)
9. Tableaux en HTML
10. Balises de structuration

03 - Définir les éléments basiques d'une page HTML

Balises de base



Rappel:

```
<!DOCTYPE html>
<html>
<head>
    <!-- en-tête du document -->
    <meta charset="UTF-8">
    <meta name="description" content="...">
    <meta name="keywords" content="...">
    <meta name="author" content="...">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="refresh" content="30">
    <title>....</title>
</head>
<body>
    <!-- corps du document-->
</body>
</html>
```

Figure 13 : Structure générale d'une page HTML

03 - Définir les éléments basiques d'une page HTML

Balises de base



WEBFORCE
BE THE CHANGE

L'entête du document HTML

```
<!DOCTYPE html>
```

Cette balise spécifie la version du langage HTML (HTML5 dans notre cas).

```
<html lang="fr" dir="rtl">
```

L'élément racine d'une page web. Peut spécifier un certain nombre de métadonnées comme la langue et l'orientation de la page.

```
<head>
```

C'est l'élément entête qui contient les métadonnées d'une page web.

Cet élément n'est pas visible sur la page web.

03 - Définir les éléments basiques d'une page HTML

Balises de base



La balise <meta>

- Les métadonnées **meta** sont des données (informations) sur les données. La balise <meta> fournit des métadonnées sur le document HTML.
- Les métadonnées ne seront pas affichées sur la page, mais seulement interprétables par les machines.

Exemples de balises meta :

```
<meta charset="UTF-8">
```

Spécifie le format d'encodage du contenu.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Cet élément est indispensable pour avoir une page web responsive (qui s'adapte à la taille de l'écran utilisé).

```
<meta name="author " content=" x y z ">
<meta name="description " content=" Cours web ">
<meta name="keywords " content=" HTML,CSS,JavaScript ">
```

Ces éléments fournissent des informations sur le document html (utilisées par les moteurs de recherche)

03 - Définir les éléments basiques d'une page HTML

Balises de base

La balise <title>

```
<title>Mon document</title>
```

Définit le texte à afficher sur la barre du titre.

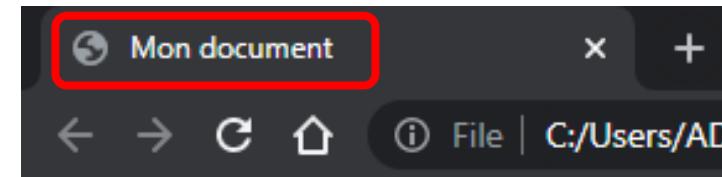


Figure 14 : Titre du document HTML

La balise <body>

```
<body>
```

C'est l'élément qui contient tous les composants visibles de la page web : les textes, les paragraphes, les images, les tableaux, les animations...

03 - Définir les éléments basiques d'une page HTML

Balises de base



Exemple d'une page HTML vide

```
<!DOCTYPE html>
<html lang="fr" dir="rtl">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Mon document</title>
</head>
<body>
    <!-- Ceci est un commentaire -->
</body>
</html>
```

03 - Définir les éléments basiques d'une page HTML

Balises de base



WEBFORCE
BE THE CHANGE

Les paragraphes en HTML

- La balise <p> désigne un nouveau paragraphe
- L'attribut "title" permet d'afficher un infobulle quand la souris survole le paragraphe

Exemple de déclaration d'un paragraphe :

```
<p title="infobulle sur le paragraphe">  
    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Enim tortor at auctor urna. Dignissim enim sit amet venenatis. Varius sit amet mattis vulputate enim. Urna et pharetra pharetra massa massa.  
</p>
```

03 - Définir les éléments basiques d'une page HTML

Balises de base



Les titres en HTML

Il existe 6 niveaux de titres (headers) en HTML. Les balises `<h1>`, `<h2>`, ...`<h6>` permettent l'affichage de ces titres.

Exemple :

```
<h1>Niveau1</h1>
<h2>Niveau2</h2>
<h3>Niveau3</h3>
<h4>Niveau4</h4>
<h5>Niveau5</h5>
<h6>Niveau6</h6>
```

Niveau1

Niveau2

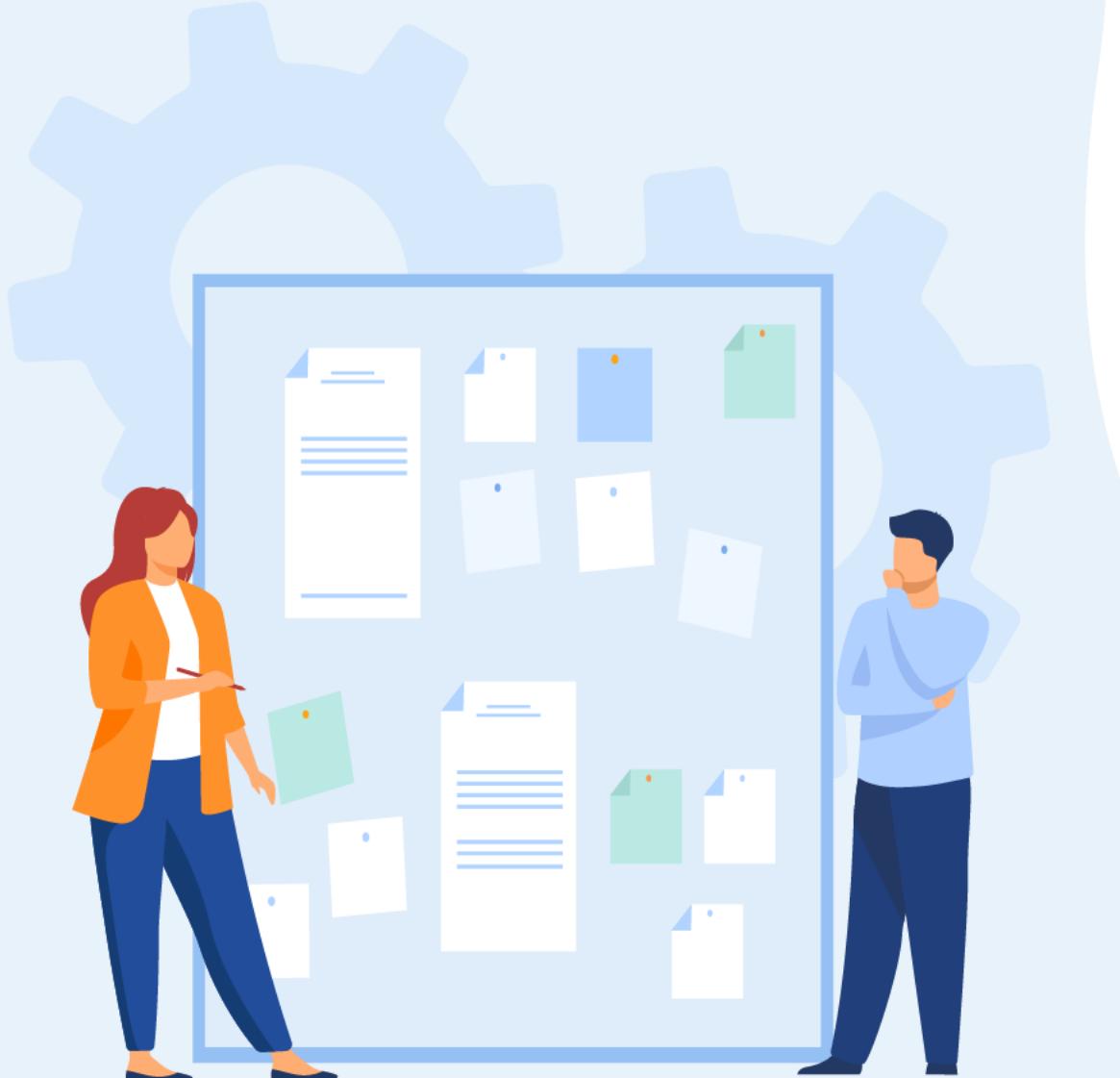
Niveau3

Niveau4

Niveau5

Niveau6

Figure 15: les titres en HTML



CHAPITRE 3

Définir les éléments basiques d'une page HTML

1. Normes W3C
2. Squelette d'une page HTML
3. Balises de bases (contenu)
- 4. Couleurs**
5. Listes en HTML
6. Types de chemin d'accès
7. Liens en HTML
8. Balises multimédia (images, audio, vidéo)
9. Tableaux en HTML
10. Balises de structuration

03 - Définir les éléments basiques d'une page HTML

Les couleurs



Les couleurs en HTML

En HTML, On peut spécifier les couleurs de 3 manières différentes :

- En utilisant la **valeur Hex** : HEX est un système de numérotation hexadécimale des couleurs (basé sur les chiffres décimaux de 0 à 9 et les lettres latines A à F). La conception Web utilise le code de couleur hexadécimal sous forme **#RRGGBB** : RR est le rouge, GG est le vert et BB est le bleu. Chaque fraction de couleur est comprise entre 00 et FF.
- En utilisant les **noms des couleurs** : Les noms des couleurs doivent être spécifiés en anglais (blue, red, etc.).
- En utilisant la **valeur RGB (Red, Green, Blue)** : Le code couleur **RGB** est représenté sous forme de 3 couples de codes en chiffres compris entre 0 et 255. Les trois codes représentent respectivement le dosage du rouge, du vert et du bleu.

Couleurs du web :

Exemples de sites pour avoir les codes des couleurs

- https://www.w3schools.com/colors/colors_picker.asp
- <https://htmlcolorcodes.com/>

03 - Définir les éléments basiques d'une page HTML

Les couleurs



WEBFORCE
BE THE CHANGE

Les couleurs en HTML (Exemples)

- L'attribut **bgcolor** : permet de préciser la couleur utilisée pour le fond du document :

```
<body bgcolor="#eab676">  
<body bgcolor="rgb(234,182,118)">
```

- La balise **** (obsolète)

```
<font color="FF0000">Le code #FF0000 est la même chose que la couleur rouge notée «red».</font>
```

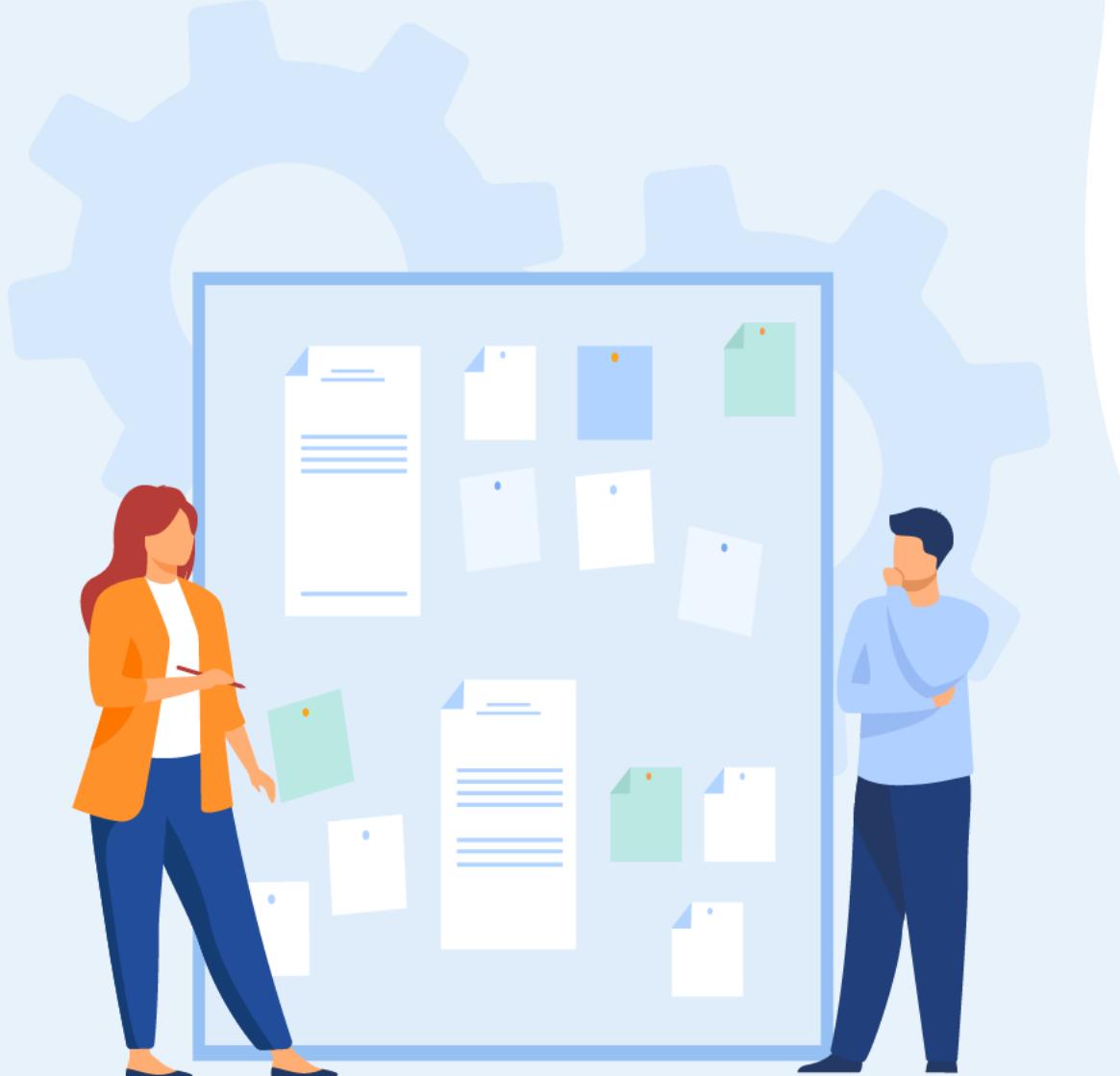
- La balise **<p>**

```
<p style="color:#8ebf42;"> C'est un texte vert</p>
```

- La balise **span** :

La balise **** est un conteneur en ligne utilisé pour marquer une partie d'un texte ou une partie d'un document. Elle est facilement stylisée par CSS ou manipulée avec JavaScript.

```
<span style="color:red">Le texte est maintenant en rouge.</span>  
<span style="color:#556B2F">Voici le code pour le vert olive foncé.</span>  
<span style="color:#745"> Le code 745 est l'abréviation de 774455.</span>
```



CHAPITRE 3

Définir les éléments basiques d'une page HTML

1. Normes W3C
2. Squelette d'une page HTML
3. Balises de bases (contenu)
4. Couleurs
5. **Listes en HTML**
6. Types de chemin d'accès
7. Liens en HTML
8. Balises multimédia (images, audio, vidéo)
9. Tableaux en HTML
10. Balises de structuration

03 - Définir les éléments basiques d'une page HTML

Les listes



WEBFORCE
BE THE CHANGE

Les listes en HTML

Les listes sont utiles pour apporter de la clarté et de l'ordre aux documents web.

En HTML, il est possible de définir trois types de listes : les listes ordonnées, les listes non ordonnées, et les listes de définition.

Les balises et les attributs utilisées sont :

- **Balise ul** : définir une liste non ordonnée
- **Balise ol** : définir une liste ordonnée
- **Balise dl** : définir une liste de définition
- **Balise li** : définir un élément de liste
- L'attribut "**type**" permet de préciser le type de numérotation utilisée pour présenter la liste
- L'attribut "**value**" permet d'initialiser le numéro courant par la valeur souhaitée

03 - Définir les éléments basiques d'une page HTML

Les listes



Les listes ordonnées et non ordonnées :

Exemple de liste non ordonnée

```
<ul>
    <li>Élément1</li>
    <li type="circle">Élément2</li>
    <li type="square">Élément3</li>
</ul>
```

Exemple de liste ordonnée

```
<ol>
    <li type="I">Élément1</li>
    <li type="i">Élément2</li>
    <li type="A">Élément3</li>
    <li type="a">Élément4</li>
</ol>
```

Exemple de liste ordonnée

```
<ol>
    <li>Élément1</li>
    <li>Élément2</li>
    <li value="1">Élément3</li>
    <li>Élément4</li>
</ol>
```

- Elément1
- Elément2
- Elément3

Figure 16 : Liste non ordonnée

- I. Elément1
- ii. Elément2
- C. Elément3
- d. Elément4

Figure 17 : Liste ordonnée 1

1. Elément1
2. Elément2
1. Elément3
2. Elément4

Figure 18 : Liste ordonnée 2

03 - Définir les éléments basiques d'une page HTML

Les listes



Les listes de définition

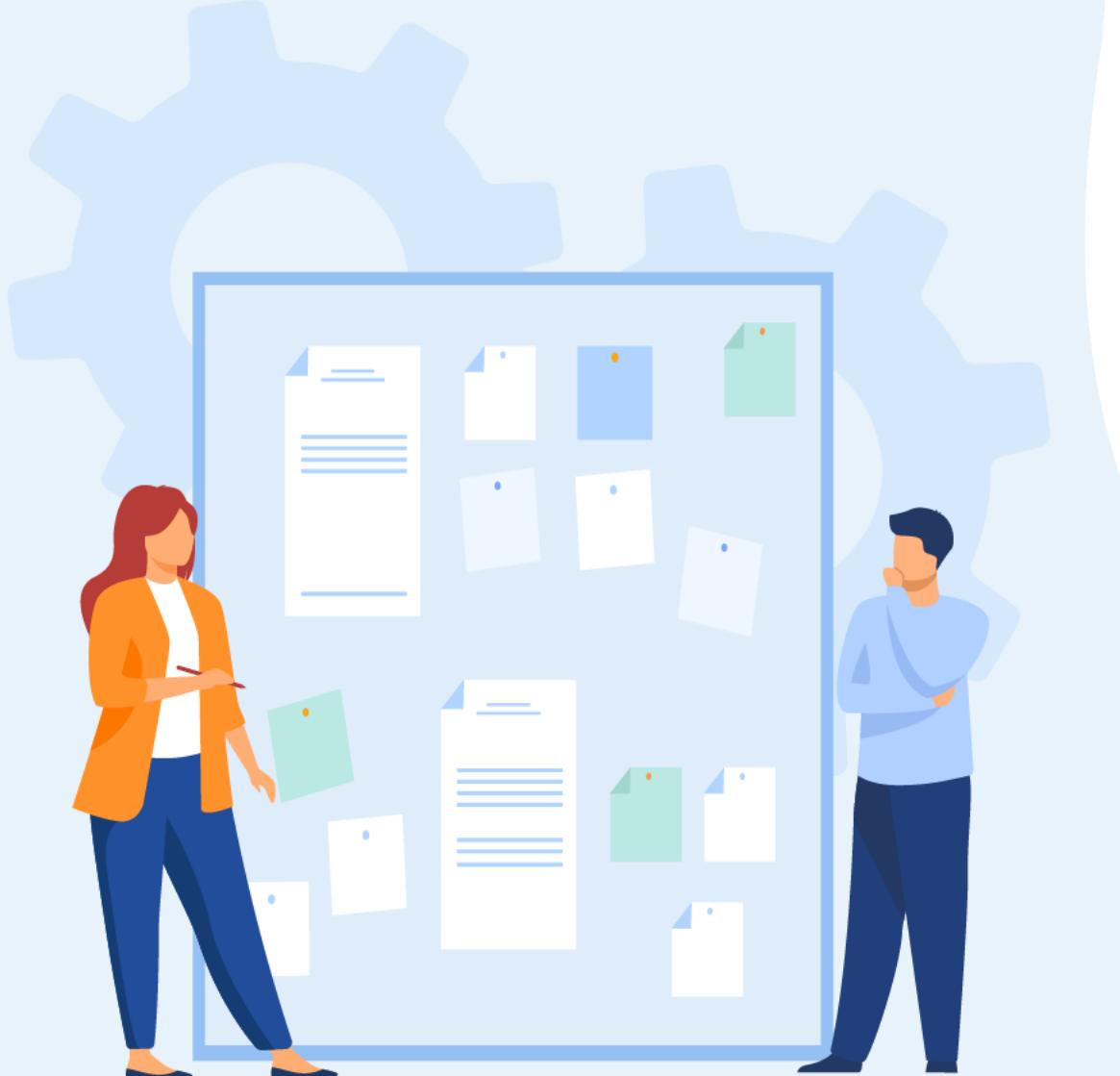
- La liste de définition est délimitée par les balises `<dl>` (Definition List),
- L'élément `DT` (Definition Title) est utilisé pour donner le titre de la définition et l'élément `DD` pour donner la définition.
- Pour les deux éléments `DT` et `DD`, la balise fermante est optionnelle.

Exemple :

```
<dl>
  <dt>Langages</dt>
    <dd>C++</dd>
    <dd>Java</dd>
    <dd>Python</dd>
  <dt>SGBD</dt>
    <dd>MySQL</dd>
    <dd>MS SQL Server</dd>
    <dd>Oracle</dd>
</dl>
```

Langages	C++
	Java
	Python
SGBD	MySQL
	MS SQL Server
	Oracle

Figure 19 : Liste de définition



CHAPITRE 3

Définir les éléments basiques d'une page HTML

1. Normes W3C
2. Squelette d'une page HTML
3. Balises de bases (contenu)
4. Les couleurs
5. Listes en HTML
6. **Types de chemin d'accès**
7. Liens en HTML
8. Balises multimédia (images, audio, vidéo)
9. Tableaux avec du HTML
10. Balises de structuration

03 - Définir les éléments basiques d'une page HTML

Types de chemins d'accès



Chemins d'accès

- Le chemin d'accès d'un fichier (ou d'un répertoire) est une chaîne de caractères représentant une liste de noms de répertoires séparés par des barres obliques. Cette chaîne décrit la position de ce fichier (ou répertoire) dans le système de fichiers.
- Généralement, un chemin standard peut être constitué de trois composants :
 - Une lettre de volume ou de lecteur suivie du séparateur de volumes (:).
 - Un nom de répertoire. Le caractère de séparation de répertoires sépare les sous-répertoires au sein de la hiérarchie des répertoires.
 - Un nom de fichier. Le caractère de séparation de répertoires sépare le chemin et le nom de fichier.
- Types de chemins d'accès :**
 - Le **chemin absolu** se rapporte toujours au dossier parent (racine).
 - Le **chemin relatif** basé sur la structure des dossiers.

03 - Définir les éléments basiques d'une page HTML

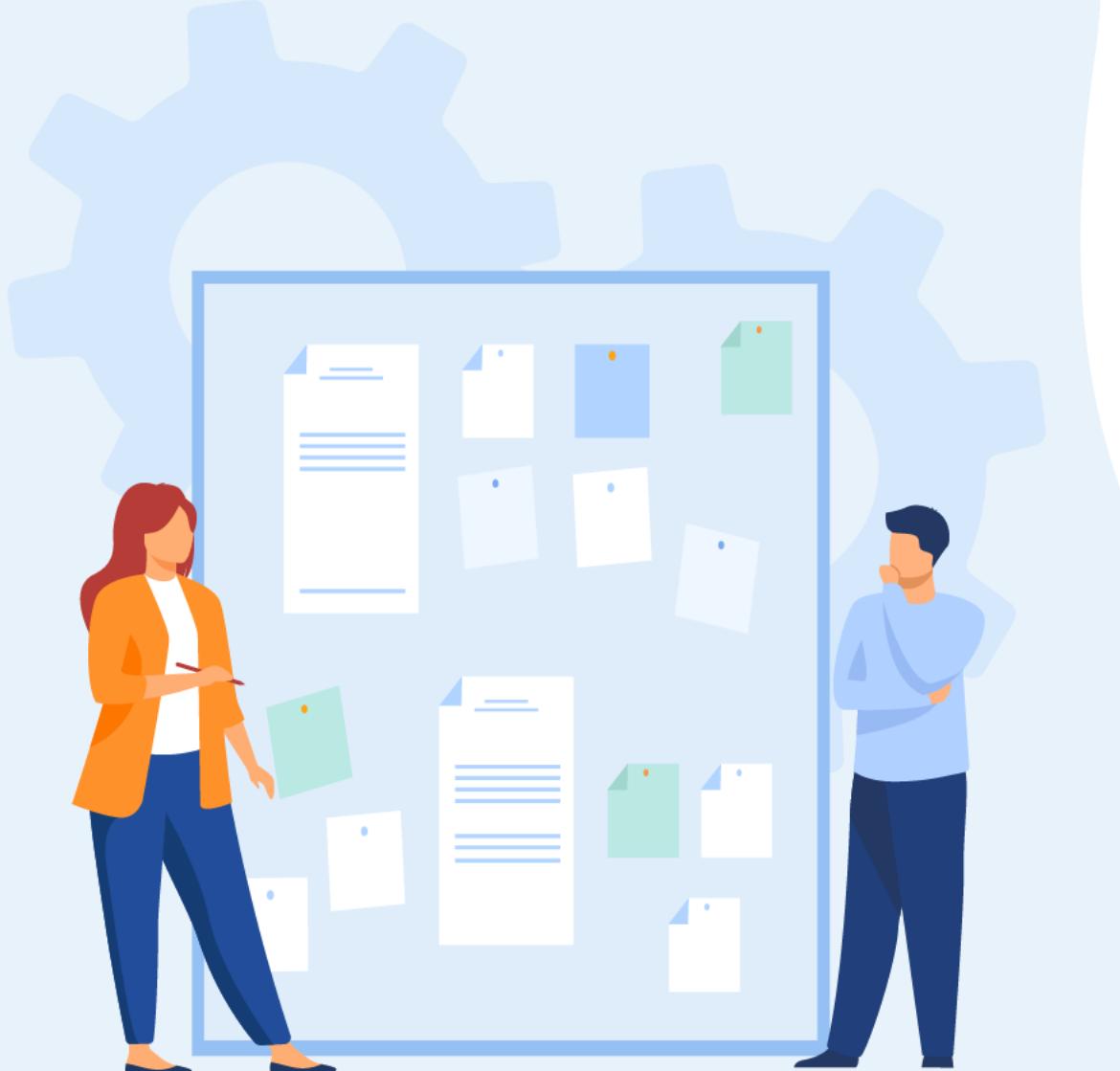
Types de chemins d'accès



Chemins d'accès

Exemples :

Chemin	Description
C:\Documents\images\logo.png	Chemin de fichier absolu à partir de la racine du lecteur C: .
\Program Files\utilities\projet.exe	Chemin absolu à partir de la racine du lecteur actif.
docs\charte.pdf	Chemin relatif à un fichier dans un sous-répertoire du répertoire actif.
..\images\logo.png	Chemin relatif à un fichier dans répertoire qui est un pair du répertoire actif.
C:\Projets\html\index.html	Chemin d'accès absolu à un fichier à partir de la racine du lecteur C:
C:Projets\html\index.html	Chemin d'accès relatif à partir du répertoire actif du C: lecteur.



CHAPITRE 3

Définir les éléments basiques d'une page HTML

1. Normes W3C
2. Squelette d'une page HTML
3. Balises de bases (contenu)
4. Les couleurs
5. Listes en HTML
6. Types de chemin d'accès
7. **Liens en HTML**
8. Balises multimédia (images, audio, vidéo)
9. Tableaux avec du HTML
10. Balises de structuration

03 - Définir les éléments basiques d'une page HTML

Les liens



La balise <a>

La balise <a> permet de créer un **lien hypertexte**, c'est-à-dire rendre une partie du texte active suite au clic de l'utilisateur.

Pour chaque lien on associe une adresse destination où l'utilisateur sera envoyé après son clic.

Cette balise permet également de créer des liens à l'intérieur d'un document ainsi que l'envoi d'un mail.

Les principaux attributs des liens hypertextes sont :

- **href** : la cible du lien
- **title** : affiche une info bulle sur le lien.
- **Target** :

_blank : ouverture de la page cible dans une nouvelle fenêtre.

_parent : ouverture de la page cible dans le cadre parent (de niveau immédiatement supérieur).

_self : (valeur par défaut), ouverture de la page cible dans le cadre d'appel.

_top : ouverture de la page cible dans la fenêtre hôte (par-dessus le FRAMESET).

03 - Définir les éléments basiques d'une page HTML

Les liens



Exemples de liens

Lien vers un autre document HTML (Lien relatif)

```
<a href="cible.html" title="Page cible">Cliquer ici pour aller à la page</a>
```

Lien vers une url (Lien absolu)

```
<a href="https://www.ofppt.ma/" title="OFPPT">Site officiel de l'OFPPT</a>
```

Un lien qui s'ouvre dans une nouvelle fenêtre

```
<a href="http://www.ofppt.ma/" title="OFPPT" target="_blank">Site officiel de l'OFPPT</a>
```

Lien vers un emplacement dans le même document (Lien interne)

```
<!-- Cible du lien -->
<h1 id="haut">titre de la page </h1>
. . .
<!-- Lien -->
<a href="#haut">haut de la page </a>
```

03 - Définir les éléments basiques d'une page HTML

Les liens



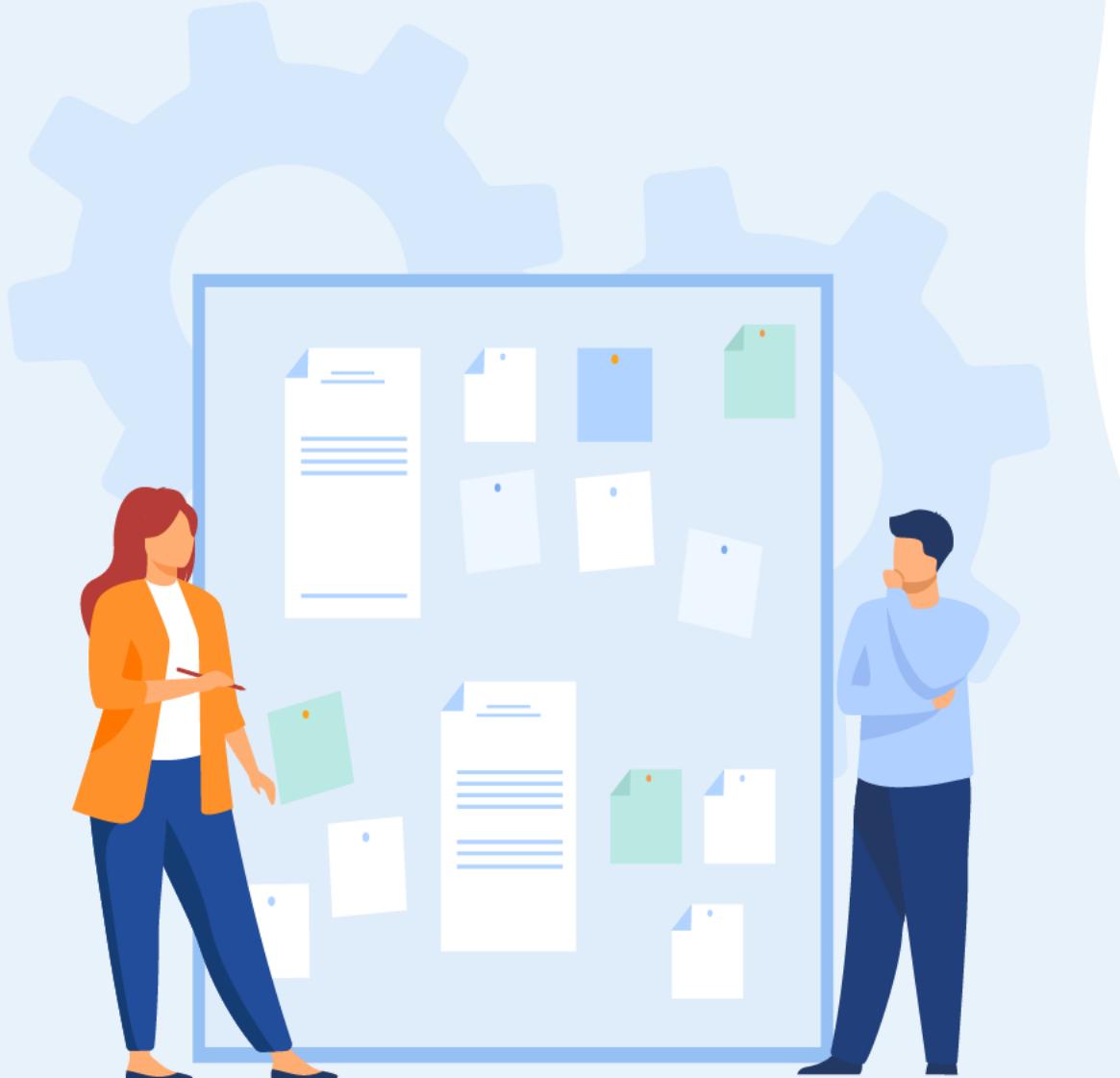
Exemples de liens

Lien pour envoyer un mail :

```
<a href="mailto:contact@ofppt.ma"> Contactez-moi par email en cliquant sur ce lien</a>
```

Lien hypertexte sur une image :

```
<a href="http://getbootstrap.com/" title="Bootstrap">
  
</a>
```



CHAPITRE 3

Définir les éléments basiques d'une page HTML

1. Normes W3C
2. Squelette d'une page HTML
3. Balises de bases (contenu)
4. Les couleurs
5. Listes en HTML
6. Types de chemin d'accès
7. Liens en HTML
- 8. Balises multimédia (images, audio, vidéo)**
9. Tableaux avec du HTML
10. Balises de structuration

03 - Définir les éléments basiques d'une page HTML

Les balises multimédia (images, audio, vidéo)



La balise ****

La balise **** permet d'insérer une image dans une page web.

- L'attribut **src** définit le chemin de l'image source
- l'attribut **alt** le texte alternatif si l'image n'est pas affichée.
- Les attributs **width** et **height** définissent la taille de l'image (en pixels)

Exemple :

```



```

03 - Définir les éléments basiques d'une page HTML

Les balises multimédia (images, audio, vidéo)



La balise <picture>

L'élément HTML <picture> permet d'afficher différentes images pour différents appareils ou tailles d'écran. Le navigateur choisira la source la plus pertinente selon :

- La disposition de la page
- L'appareil utilisé
- Les formats pris en charge

Si aucune correspondance n'est trouvée parmi les éléments <source>, c'est le fichier défini par l'attribut src de l'élément qui sera utilisé.

Exemple :

```
<picture>
    <source srcset=".../Photos/ofppt.jpg" media="(min-width : 600px)">
        
    </picture>
```

Si la largeur de l'écran dépasse 600 px, c'est la source « ofppt.jpg » qui prend l'objet <picture>.

03 - Définir les éléments basiques d'une page HTML

Les balises multimédia (images, audio, vidéo)



WEBFORCE
BE THE CHANGE

La balise <audio>

L'élément <audio> permet de lire un fichier audio dans la page HTML. Les attributs utilisés avec cette balise sont :

- **Controls** : pour afficher les boutons play, stop, preview...
- **Autoplay** : pour démarrer la lecture automatique de l'audio lors du chargement de la page web.

Le navigateur essaie de lire la première source. En cas d'erreur, il passe à la deuxième source. Si le navigateur ne prend pas en charge l'élément <audio>, il affiche le message en bas.

Exemple :

```
<audio controls autoplay>
    <source src="son.ogg" type="audio/ogg">
    <source src="son.mp3" type="audio/mpeg">Votre navigateur ne supporte pas la balise "audio".
</audio>
```

03 - Définir les éléments basiques d'une page HTML

Les balises multimédia (images, audio, vidéo)



La balise <video>

L'élément <video> permet d'incorporer une vidéo dans une page Web. Les attributs utilisés avec cette balise sont :

- **Controls** : pour afficher les boutons play, stop, ...
- **Autoplay** : pour démarrer la lecture automatique de la vidéo lors du chargement de la page web.
- **Poster** : spécifie une image à afficher pendant le téléchargement de la vidéo. Si cet attribut n'est pas renseigné, la première image de la vidéo sera utilisée à sa place.

Le navigateur essaie de lire la première source. En cas d'erreur, il passe à la deuxième source. Si le navigateur ne prend pas en charge l'élément <video>, il affiche le message en bas.

Exemple :

```
<video width="320" height="240" controls autoplay poster="/images/poster.gif">
    <source src="movie.mp4" type="video/mp4">
    <source src="movie.ogg" type="video/ogg"> Votre navigateur ne supporte pas la balise "video".
</video>
```

03 - Définir les éléments basiques d'une page HTML

Les balises multimédia (images, audio, vidéo)



La balise <iframe>

La balise <iframe> permet d'afficher une vidéo

Pour ce faire, on peut utiliser l'identifiant de la vidéo pour y faire référence dans le code HTML.

Exemple :

```
<iframe width="640" height="360" src="https://www.youtube.com/embed/Ma1vSev07ug">  
</iframe>
```

03 - Définir les éléments basiques d'une page HTML

Les balises multimédia (images, audio, vidéo)



La balise <object>

L'élément <object> est une solution polyvalente pour inclure des objets génériques.

Cet élément permet aux auteurs HTML de spécifier tout ce qui est requis par un objet pour sa présentation par un agent utilisateur.

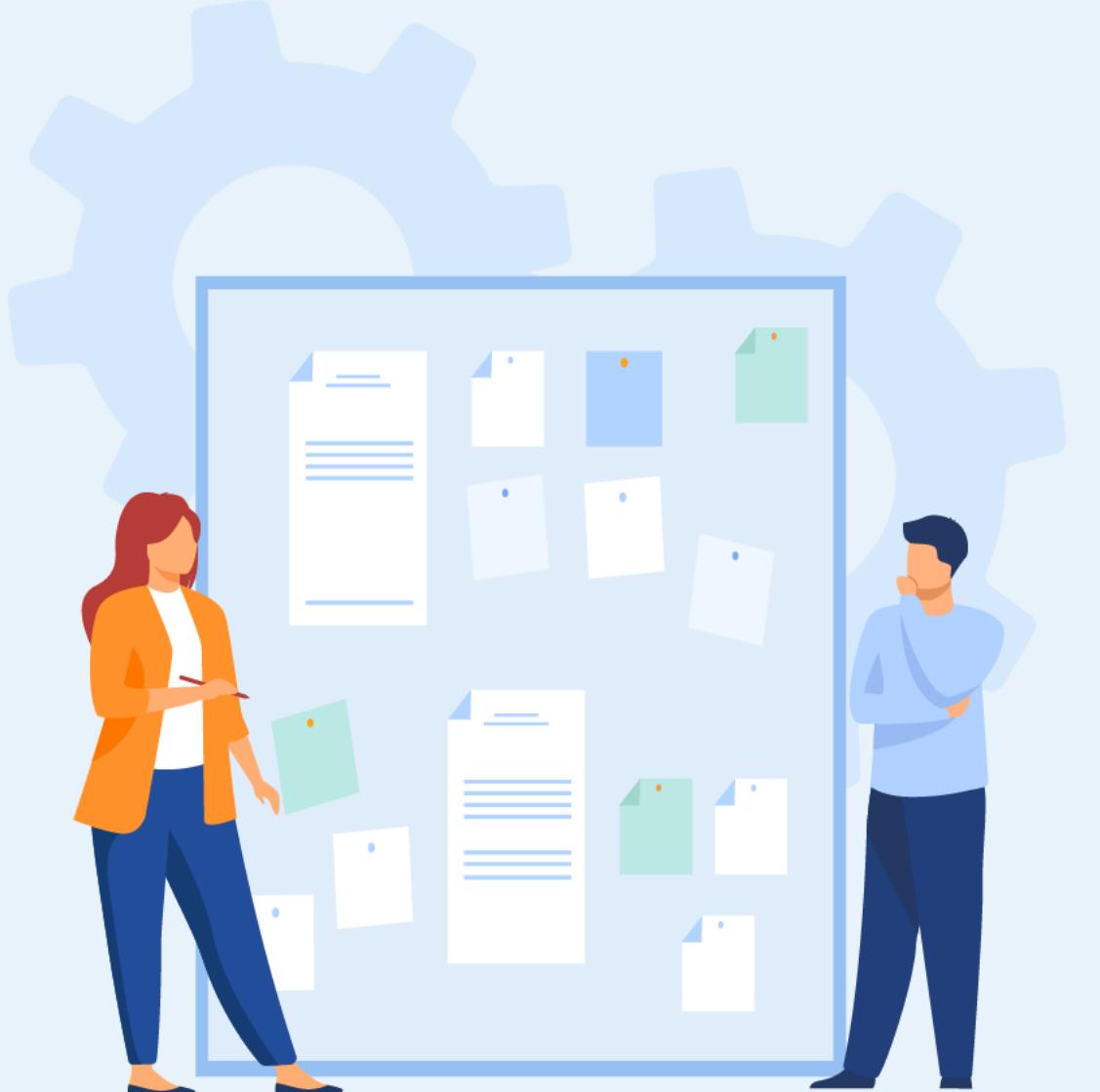
Exemple 1 : intégrer un document HTML dans un autre document HTML

```
<object data = "data/test.html" type = "text/html" width = "300" height = "200">
    alt : <a href = "data/test.html">test.html</a>
</object>
```

L'attribut « alt » apparaitra si le navigateur ne prend pas en charge la balise d'objet.

Exemple 2 : intégrer un document PDF dans un document HTML

```
<object data = "data/test.pdf" type = "application/pdf" width = "300" height = "200">
    alt : <a href = "data/test.pdf">test.html</a>
</object>
```



CHAPITRE 3

Définir les éléments basiques d'une page HTML

1. Normes W3C
2. Squelette d'une page HTML
3. Balises de bases (contenu)
4. Les couleurs
5. Listes en HTML
6. Types de chemin d'accès
7. Liens en HTML
8. Balises multimédia (images, audio, vidéo)
9. **Tableaux avec du HTML**
10. Balises de structuration

03 - Définir les éléments basiques d'une page HTML

Les tableaux



La balise <table>

Pour insérer un tableau, on utilise la balise <table>.

À l'intérieur de la balise <table>, la balise <tr> (table row) signifie une ligne.

Chaque ligne est composée d'une ou plusieurs cellules de données définies par la balise <td>.

Pour l'entête du tableau, on utilise la balise <th> (table header).

L'attribut Border permet de fixer la largeur des bordures des cellules.

Exemple :

```
<table border="1">
    <caption align="bottom">Liste des employés</caption>
    <tr>
        <th>Nom</th><th>Prénom</th><th>Age</th>
    </tr>
    <tr>
        <td>ENAANAI</td><td>Adil</td><td>40</td>
    </tr>
    <tr>
        <td>SAFIR</td><td>Laila</td><td>32</td>
    </tr>
    <tr>
        <td>HAMIM</td><td>Mohamed</td><td>39</td>
    </tr>
</table>
```

Nom	Prénom	Age
ENAANAI	Adil	40
SAFIR	Laila	32
HAMIM	Mohamed	39

Liste des employés

Figure 20 : Exemple de tableau en HTML

03 - Définir les éléments basiques d'une page HTML

Les tableaux

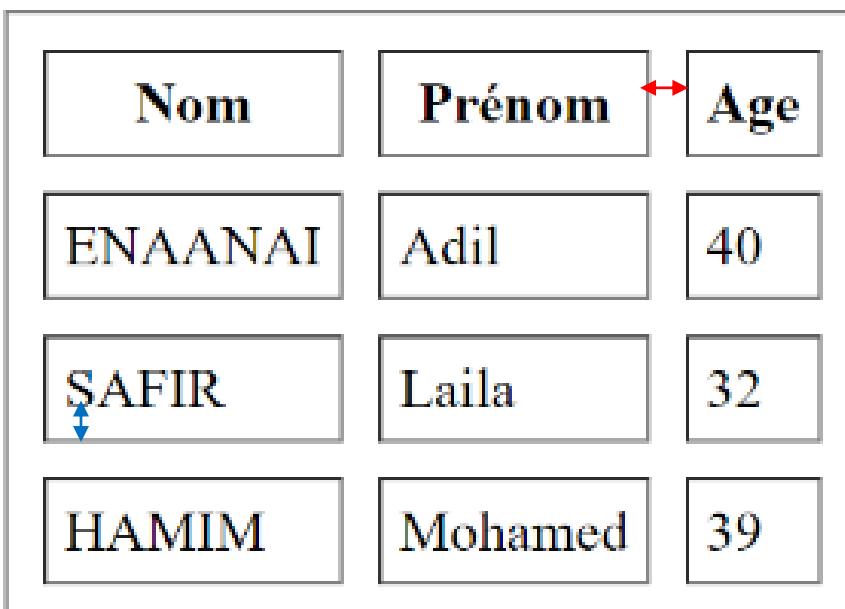
Espacement ses cellules

L'attribut "cellspacing" permet de définir l'espacement entre les cellules du tableau.

l'attribut "cellpadding" permet de définir l'espacement entre le contenu de la cellule et ses bords.

Exemple :

```
<table border="1" cellspacing=10 cellpadding=5>
```



Nom	Prénom	Age
ENAANAI	Adil	40
SAFIR	Laila	32
HAMIM	Mohamed	39

The diagram illustrates the effect of the cellspacing and cellpadding attributes on a 3x3 table. The first row contains the column headers: 'Nom', 'Prénom', and 'Age'. The second row contains the data: 'ENAANAI', 'Adil', and '40'. The third row contains the data: 'SAFIR', 'Laila', and '32'. The fourth row contains the data: 'HAMIM', 'Mohamed', and '39'. Blue arrows labeled 'cellpadding' point from the text content to the borders of the first and third rows. Red arrows labeled 'cellspacing' point between the first and second rows, and between the second and third rows, indicating the space between the cell boundaries.

cellspacing

cellpadding

Figure 21 : Espacement des cellules du tableau

03 - Définir les éléments basiques d'une page HTML

Les tableaux



Division des lignes et des colonnes (fusion des cellules)

Les attributs **colspan** et **rowspan** indiquent sur combien de colonnes et de lignes s'étend la cellule.

Exemple 1: On souhaite que la première cellule du tableau prenne toute la largeur de la ligne. Elle doit donc déborder sur 3 cellules horizontales : Colspan=3

```
<table width=60% border=1>
  <tr>
    <td colspan=3>cellule 1</td>
  </tr>
  <tr>
    <td width=33%>cellule 2</td>
    <td width=33%>cel 2</td>
    <td width=34%>cel 4</td> </tr>
  </table>
```

cellule 1		
cellule 2	cel 3	cel 4

Figure 22 : Exemple de tableau avec fusion des colonnes

Exemple 2: On souhaite que la première colonne du tableau déborde sur 2 lignes : rowspan=2

```
<table width=60% border=1>
  <tr>
    <td width=33% rowspan=2>cellule 1</td>
    <td width=33%>cel 2</td>
    <td width=34%>cel 3</td> </tr>
  </tr>
  <tr>
    <td width=33%>cel 4</td>
    <td width=34%>cel 5</td> </tr>
  </table>
```

cellule 1	cel 2	cel 3
	cel 4	cel 5

Figure 23 : Exemple de tableau avec fusion des lignes



CHAPITRE 3

Définir les éléments basiques d'une page HTML

1. Normes W3C
2. Squelette d'une page HTML
3. Balises de bases (contenu)
4. Les couleurs
5. Listes en HTML
6. Types de chemin d'accès
7. Liens en HTML
8. Balises multimédia (images, audio, vidéo)
9. Tableaux avec du HTML
- 10. Balises de structuration**

03 - Définir les éléments basiques d'une page HTML

Balises de structuration



Les balises <div>

La balise HTML <div> (ou division) est un conteneur générique : elle peut contenir n'importe quel élément HTML.

Cet élément, de type **bloc**, n'a aucun effet sur le contenu ou la mise en page tant qu'il n'est pas mis en forme d'une manière explicite (en utilisant un style CSS).

<div> est très souvent utilisé avec la mise en page CSS d'une page Web.

Les balises

La balise ne produit aucun effet visuel mais sert à appliquer un effet de style CCS aux éléments qu'elle contient (un texte par exemple).

03 - Définir les éléments basiques d'une page HTML

Balises de structuration



Les balises sémantiques

Une page web est souvent composée d'un menu, d'une sidebarre et d'un pied de page.

Tous ces ensembles peuvent (et doivent) être compris dans des balises qui vont structurer la page.

Un élément sémantique décrit clairement son sens au navigateur et au développeur.

Des exemples d'éléments non-sémantiques :

- <div> et - ne disent rien sur leur contenu.

Des exemples d'éléments sémantiques :

- <form>, <table> et - contenu clairement défini.

03 - Définir les éléments basiques d'une page HTML

Balises de structuration

Les balises sémantiques

Plusieurs sites Web utilisent les balises comme : <div id="nav">, <div class="header">, ou <div id="footer">, pour indiquer les liens de navigation, en-tête, pied de page.

HTML5 propose des éléments sémantiques permettant de définir clairement les différentes parties d'une page web :

- <header>
- <nav>
- <section>
- <article>
- <aside>
- <figure> / <figcaption>
- <footer>
- <details> / <summary>
- <mark>
- <time>

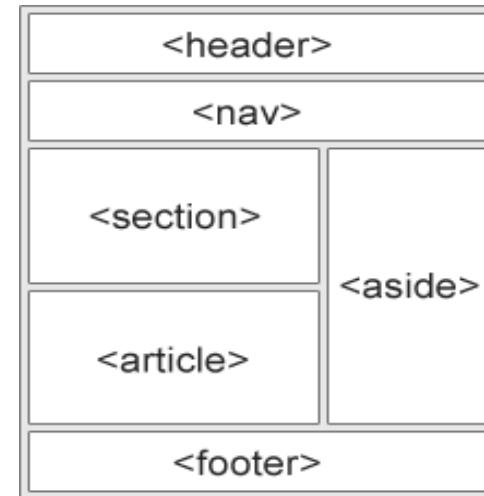


Figure 24 : Les éléments sémantiques [8]

03 - Définir les éléments basiques d'une page HTML

Balises de structuration



Les balises <header> et <footer>

Une page est généralement composée d'un header et d'un footer.

- Le footer reprend en général des liens vers les crédits et les mentions légales, et aussi vers certaines pages qui ne sont pas listées dans le menu :
- Le header peut comprendre (liste non exhaustive) : un menu, un logo, des liens vers des réseaux sociaux, ...

Exemples :

```
<header>
  <h1>Titre de la partie</h1>
  <p>Publié par CDC</p>
  <p>Informations supplémentaires</p>
</header>
<p>....</p>
```

```
<footer>
  <p>Auteur: CDC NTIC<br>
    <a href="mailto:cdc@ofppt.ma">cdc@ofppt.ma</a></p>
</footer>
```

03 - Définir les éléments basiques d'une page HTML

Balises de structuration



La balise <nav>

La balise <nav> (utilisée avec une balise ou) permet de spécifier les éléments de navigation de la page (le menu).

Le menu consiste en une liste d'éléments permettant de naviguer entre plusieurs pages du même site, ou bien vers plusieurs sections de la page.

Exemple :

```
<nav>
  <ul>
    <li><a href="index.html">Accueil</a></li>
    <li><a href="inscription.html">Inscription</a></li>
    <li><a href="a_propos.html">À propos</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</nav>
```

03 - Définir les éléments basiques d'une page HTML

Balises de structuration



Les balises <main> et <section>

La balise <main> spécifie le contenu principal de la page web.

La balise <section> permet de segmenter le contenu en plusieurs sections.

Exemple :

```
<main>
  <p>Texte d'introduction de mon contenu.</p>
  <section>
    <h2>Titre de ma section</h2>
    <p>Texte de ma section.</p>
  </section>
  <section>
    <h2>Titre de ma section</h2>
    <p>Texte de ma section.</p>
  </section>
</main>
```

03 - Définir les éléments basiques d'une page HTML

Balises de structuration



La balise <article>

La balise <article> permet d'incorporer du contenu annexe, comme par exemple :

- des articles de blog,
- des produits,
- des commentaires,
- ...

Exemple :

```
<main>
    <p>Texte de mon contenu. Voilà des articles liés à ma page :</p>
    <div>
        <article>
            <h2>Titre de mon article</h2>
            <p>Texte de mon article.</p>
        </article>
        <article>
            <h2>Titre de mon article</h2>
            <p>Texte de mon article.</p>
        </article>
    </div>
</main>
```

03 - Définir les éléments basiques d'une page HTML

Balises de structuration



La balise `<aside>`

La balise `<aside>` permet d'afficher du contenu qui n'a pas de rapport direct avec le contenu principal, souvent représenté comme une sidebarre :

Exemple :

```
<main>
  <p>Texte de mon contenu.</p>
  <aside>
    <p>. </p>
    Texte sans rapport direct
  </aside>
</main>
```



PARTIE 2

Implémenter une page web statique

Dans ce module, vous allez :

- Avoir une connaissance exacte du fonctionnement d'une page web statique
- Réaliser correctement un site web statique
- Intégration appropriée d'un formulaire dans une page web



20 heures

CHAPITRE 1

Réaliser une page web statique



Ce que vous allez apprendre dans ce chapitre :

- Comprendre le fonctionnement d'une page web statique
- Réaliser un site web statique avec HTML5



02 heures



CHAPITRE 1

Réaliser une page web statique

1. Fonctionnement d'une page web statique
2. Réalisation d'un site web statique avec HTML5

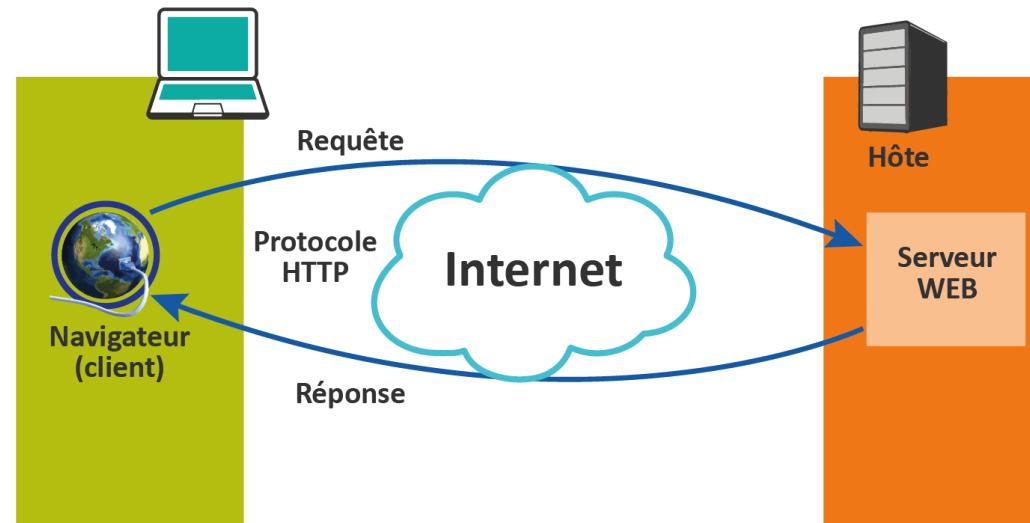
01 - Fonctionnement d'une page web statique

Fonctionnement d'une page web

Fonctionnement d'une page web

Reprends le schéma présenté dans la partie 1 (slide 13) :

Lors de la consultation d'une page web statique, le serveur HTTP envoie le contenu du fichier qui contient les composants de la page à afficher. Le navigateur interprète le contenu (sous forme de balises HTML) et l'affiche au client.





CHAPITRE 1

Réaliser une page web statique

1. Fonctionnement d'une page web statique
2. **Réalisation d'un site web statique avec HTML5**

01 - Réalisation d'un site web statique

Réalisation d'un site web statique avec HTML5



En HTML, il existe cinq façons différentes de créer des dispositions des éléments :

- Tableaux HTML (non recommandé)
- Propriété **float** CSS (méthode ancienne)
- **Flexbox** CSS
- Framework CSS (W3.CSS ou **Bootstrap**)
- Grille CSS

Le **Responsive Web Design** consiste à utiliser HTML et CSS pour redimensionner, masquer, rétrécir ou agrandir automatiquement un site Web afin de le rendre compatible à tous les appareils (ordinateurs de bureau, tablettes et téléphones)

01 - Réalisation d'un site web statique

Réalisation d'un site web statique avec HTML5



Exemple : Afficher différentes images en fonction de la largeur du navigateur

```
<picture>
    <source srcset="img_smallflower.jpg" media="(max-width : 600px)">
    <source srcset="img_flowers.jpg" media="(max-width : 1500px)">
    <source srcset="flowers.jpg">
        
</picture>
```

NB : n'oubliez pas de mettre la balise suivante dans la balise <head>

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

01 - Réalisation d'un site web statique

Réalisation d'un site web statique avec HTML5



Exemple : Taille du texte réactif

La taille du texte peut être définie avec une unité "vw", ce qui signifie la "largeur de la fenêtre".

De cette façon, la taille du texte suivra la taille de la fenêtre du navigateur :

```
<h1 style="font-size :10vw">Bonjour</h1>
```

C'est un pourcentage



CHAPITRE 2

Intégrer des formulaires dans une page web

Ce que vous allez apprendre dans ce chapitre :

- Créer des formulaires HTML
- Gérer les paramètres d'envoi des données



18 heures



CHAPITRE 2

Intégrer des formulaires dans une page web

1. **Balise <form>**
2. Champs de saisie (Input, TextArea)
3. Balises de choix (RadioButton, CheckBox, Select)
4. Balises HTML5 pour les formulaires
5. Validation automatique
6. Boutons

02 - Intégrer des formulaires dans une page web

Balise <form>



Balise <form>

Un formulaire est une section d'un document web qui permet à un utilisateur de fournir des informations via des contrôles interactifs

L'élément HTML <form> permet de déclarer un formulaire

L'attribut "action" spécifie la ressource qui va recevoir les données du formulaire pour les traiter (un script)

L'attribut "method" spécifie la méthode d'envoi des données

On distingue deux méthodes :

- **GET** : permet d'envoyer les données sous forme de chaîne de paramètres à travers l'URL. Les données sont visibles dans la barre d'adresse.
- **POST** : permet d'envoyer les données (invisibles) à travers une requête HTTP.

Exemple :

```
<form action="authentification.php" method="post">
    <!-- Ici on insère les éléments du formulaire-->
</form>
```



CHAPITRE 2

Intégrer des formulaires dans une page web

1. Balise <form>
2. **Champs de saisie (Input, TextArea)**
3. Balises de choix (RadioButton, CheckBox, Select)
4. Balises HTML5 pour les formulaires
5. Validation automatique
6. Boutons

02 - Intégrer des formulaires dans une page web

Champs de saisie <input>



Champ de saisie <input>

L'élément <input> Permet de coder différents types d'entrée pour un formulaire, selon la valeur de l'attribut type.

Exemple :

```
<form action="authentification.php" method="post">
    <input type="text" id="nom" name="nom">
</form>
```

Types d'Input HTML

<input type="**button**">
<input type="**checkbox**">
<input type="**color**">
<input type="**date**">
<input type="**datetime-local**">
<input type="**email**">
<input type="**file**">
<input type="**hidden**">

<input type="**image**">
<input type="**month**">
<input type="**number**">
<input type="**password**">
<input type="**radio**">
<input type="**range**">
<input type="**reset**">
<input type="**search**">

<input type="**submit**">
<input type="**tel**">
<input type="**text**">
<input type="**time**">
<input type="**url**">
<input type="**week**">

02 - Intégrer des formulaires dans une page web

Champs de saisie <input>

Champ de saisie <input>

Type texte

```
<form>
    <label for="pr">Prénom :</label><br>
    <input type="text" id="pr" name="pr"><br>
    <label for="nf">Nom de famille :</label><br>
    <input type="text" id="nf" name="nf">
</form>
```

Pour retourner à la ligne

Prénom:

Nom de famille:

Figure 25 : Champ input

Type password

```
<form>
    <label for="nomutil">Nom d'utilisateur :</label><br>
    <input type="text" id="nomutil" name="nomutil"><br>
    <label for="pwd">Mot de passe :</label><br>
    <input type="password" id="pwd" name="pwd">
</form>
```

Nom d'utilisateur:

Mot de passe:

Figure 26 : Champ de mot de passe

02 - Intégrer des formulaires dans une page web

Champs de saisie <textarea>



Champ de saisie <textarea>

L'élément <textarea> permet d'afficher et saisir un texte en plusieurs ligne.

Exemple :

```
<textarea>
    L'Office de la formation professionnelle et de la promotion du travail (OFPPT) est un opérateur public en
    formation professionnelle.
</textarea>
```

L'Office de la formation professionnelle et de la promotion du travail (OFPPT) est un opérateur public en formation professionnelle.

Figure 27 : Champ textarea



CHAPITRE 2

Intégrer des formulaires dans une page web

1. Balise <form>
2. Champs de saisie (Input, TextArea)
3. **Balises de choix (RadioButton, CheckBox, Select)**
4. Balises HTML5 pour les formulaires
5. Validation automatique
6. Boutons

02 - Intégrer des formulaires dans une page web

Balises de choix (RadioButton)

Type radio

Les boutons radio permettent à un utilisateur d'en sélectionner SEULEMENT UN parmi un nombre limité de choix :

```
<form>
    <input type="radio" id="mas" name="sexe" value="Masculin">
    <label for="mas">Masculin</label><br>
    <input type="radio" id="fem" name="sexe" value="Féminin">
    <label for="fem">Féminin</label><br>
</form>
```

- Masculin
- Féminin

Figure 28 : Champ bouton radio

Type number

```
<form>
    <label for="quant">Quantité :</label>
    <input type="number" id="quant" name="quant" min="0" max="100" step="10" value="30">
</form>
```

Quantité:

Figure 29 : Champ number

02 - Intégrer des formulaires dans une page web

Balises de choix (CheckBox)

Type checkbox

<input type="checkbox"> définit une case à cocher.

Les cases à cocher permettent à un utilisateur de sélectionner ZERO ou PLUS d'options d'un nombre limité de choix.

```
<form>
    <input type="checkbox" id="vehicule1" name="vehicule1" value="velo">
    <label for="vehicule1"> J'ai un vélo</label><br>
    <input type="checkbox" id="vehicule2" name="vehicule2" value="voiture">
    <label for="vehicule2">J'ai une voiture</label><br>
    <input type="checkbox" id="vehicule3" name="vehicule3" value="bateau">
    <label for="vehicule3"> J'ai un bateau</label>
</form>
```

- J'ai un vélo
- J'ai une voiture
- J'ai un bateau

Figure 30 : Champ Checkbox

Type select

L'élément HTML **<select>** représente un contrôle qui fournit un menu d'options :

- L'attribut **id** associe un identifiant unique à la liste
- L'attribut **name** représente le nom du point de données associé lors de la soumission au serveur.
- L'attribut **multiple** spécifie si plusieurs options peuvent être sélectionnées, et **size** spécifie combien d'options doivent être affichées à la fois
- L'élément **<option>** définit une option de la liste

Chaque élément **<option>** doit avoir un attribut **value** contenant la valeur de données à soumettre au serveur lorsque cette option est sélectionnée. L'attribut **selected** sur un élément **<option>** le rend sélectionné par défaut lors du premier chargement de la page.

```
<label for="langage">Sélectionner un langage de programmation:</label>
<select name="lg" id="langage">
    <option value="">--Choisir --</option>
    <option value="Python">Python</option>
    <option value="CSharp">C#</option>
    <option value="java">java</option>
</select>
```



CHAPITRE 2

Intégrer des formulaires dans une page web

1. Balise <form>
2. Champs de saisie (Input, TextArea)
3. Balises de choix (RadioButton, CheckBox, Select)
- 4. Balises HTML5 pour les formulaires**
5. Validation automatique
6. Boutons

02 - Intégrer des formulaires dans une page web

Balises HTML5 pour les formulaires



Type button

Définit un bouton

```
<input type="button" onclick="alert('Bonjour!')" value="Clic ici!">
```

Type color

Définit un sélecteur de couleur

```
<form>
  <label for="coulfav">Sélectionner votre couleur préférée :</label>
  <input type="color" id="coulfav" name="coulfav">
</form>
```

Type range

Définit un contrôle pour entrer un nombre dont la valeur exacte n'est pas importante (comme un contrôle curseur). Plage par défaut est de 0 à 100.

```
<form>
  <label for="vol">Volume (entre 0 and 50) :</label>
  <input type="range" id="vol" name="vol" min="0" max="50">
</form>
```

02 - Intégrer des formulaires dans une page web

Balises HTML5 pour les formulaires



Type date

Définit un sélecteur de date. La valeur comprend l'année, le mois et le jour

```
<form>
  <label for="anniv">Anniversaire :</label>
  <input type="date" id="anniv" name="anniv">
</form>
```

Type time

Définit un contrôle pour saisir une heure (pas de fuseau horaire)

```
<form>
  <label for="rdv">Sélectionner une heure :</label>
  <input type="time" id="rdv" name="rdv">
</form>
```

Type datetime-local

Spécifie un champ de saisie de date et d'heure sans fuseau horaire

```
<form>
  <label for="anniv">Anniversaire (date et heure) :</label>
  <input type="datetime-local" id="anniv" name="anniv">
</form>
```

02 - Intégrer des formulaires dans une page web

Balises HTML5 pour les formulaires



Type week

Définit un contrôle de sélection de la semaine et l'année (sans fuseau horaire)

```
<form>
    <label for="sem">Choisir une semaine :</label>
    <input type="week" id="sem" name="sem">
</form>
```

Type month

Définit un champ de sélection du mois et de l'année. Le format est « AAAA-MM »

```
<form>
    <label for="anniv">Anniversaire (mois et année) :</label>
    <input type="month" id="anniv" name="anniv">
</form>
```

02 - Intégrer des formulaires dans une page web

Balises HTML5 pour les formulaires



Type file

Définit un champ de sélection de fichier et un bouton « Parcourir » pour les téléchargements de fichiers.

```
<form>
    <label for="fichiers">Sélectionner un fichier :</label>
    <input type="file" id="fichiers" name="fichiers">
</form>
```

Type email

Utilisé pour bénéficier de la possibilité de validation d'Email et de voir les boutons raccourcis d'un clavier spécifique si on utilise la page web sur un smartphone.

```
<form>
    <label for="email">Saisir votre email :</label>
    <input type="email" id="email" name="email">
</form>
```

02 - Intégrer des formulaires dans une page web

Balises HTML5 pour les formulaires



Type tel

Utilisé pour les champs de saisie qui doivent contenir un numéro de téléphone.

```
<form>
    <label for="phone">Saisir un numéro de téléphone :</label>
    <input type="tel" id="phone" name="phone" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}">
</form>
```

Type url

Utilisé pour les champs de saisie qui doivent contenir une adresse URL.

```
<form>
    <label for="acc">Ajouter votre page d'accueil :</label>
    <input type="url" id="acc" name="acc">
</form>
```

Type search

Définit un champ de recherche (comme une recherche dans un site ou une recherche Google).

```
<form>
    <label for="gsearch">Recherche Google :</label>
    <input type="search" id="gsearch" name="gsearch">
</form>
```

02 - Intégrer des formulaires dans une page web

Balises HTML5 pour les formulaires



L'attribut value

L'attribut **value** spécifie une valeur initiale pour un champ d'entrée.

```
<form>
  <label for="fil">Filière :</label><br>
  <input type="text" id="fil" name="fil" value="Développement"><br>
  <label for="grp">Groupe :</label><br>
  <input type="text" id="grp" name="grp" value="101">
</form>
```

L'attribut readonly

L'attribut **readonly** spécifie qu'un champ d'entrée est en lecture seule.

```
<form>
  <label for="fil">Filière :</label><br>
  <input type="text" id="fil" name="fil" value="Développement" readonly><br>
  <label for="grp">Groupe :</label><br>
  <input type="text" id="grp" name="grp" value="101">
</form>
```

L'attribut maxlength

L'attribut **maxlength** spécifie le nombre maximum de caractères autorisés.

```
<form>
  <label for="fname">Nom :</label><br>
  <input type="text" id="fname" name="fname" size="50"><br>
  <label for="pin">PIN :</label><br>
  <input type="text" id="pin" name="pin" maxlength="4" size="4">
</form>
```

L'attribut multiple

L'attribut **multiple** spécifie que l'utilisateur est autorisé à saisir plusieurs valeurs. Il fonctionne avec les types d'entrée suivants : e-mail et fichier.

```
<form>
  <label for="fichiers">Sélectionner un fichier :</label>
  <input type="file" id="fichiers" name="fichiers" multiple>
</form>
```

L'attribut pattern

- L'attribut **pattern** spécifie une expression régulière par rapport à laquelle la valeur du champ de saisie est vérifiée lorsque le formulaire est soumis.
- L'attribut **pattern** fonctionne avec les types d'entrée suivants : texte, date, recherche, URL, tél, e-mail et mot de passe.

```
<form>
  <label for="code_pays">Code pays :</label>
  <input type="text" id="code_pays" name="code_pays" pattern="[A-Za-z]{3}" title="Code pays à trois lettres">
</form>
```

L'attribut placeholder

- L'attribut **placeholder** spécifie brièvement un indice qui décrit la valeur attendue d'un champ de saisie (une valeur d'échantillon ou une brève description du format attendu).
- L'attribut **placeholder** fonctionne avec les types d'entrée suivants : texte, recherche, URL, tél, e-mail et mot de passe.

```
<form>
  <label for="phone">Saisir un numéro de téléphone :</label>
  <input type="tel" id="phone" name="phone" placeholder="123-45-678" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}">
</form>
```

02 - Intégrer des formulaires dans une page web

Balises HTML5 pour les formulaires



L'attribut required

- L'attribut **required** spécifie qu'un champ de saisie doit être rempli avant de soumettre le formulaire.
- L'attribut **required** fonctionne avec les types d'entrée suivants : texte, recherche, URL, tél, e-mail, mot de passe, sélecteurs de date, numéro, case à cocher, radio et fichier.

```
<form>
  <label for="username">Nom complet :</label>
  <input type="text" id="username" name="username" required>
</form>
```

L'attribut step

- L'attribut **step** spécifie les intervalles de numéros légaux pour un champ de saisie.
- L'attribut **step** fonctionne avec les types d'entrée suivants : **number, range, date, datetime-local, month, time et week**.

Exemple : si step = "3", les nombres légaux peuvent être -3, 0, 3, 6, etc.

```
<form>
  <label for="points">Points :</label>
  <input type="number" id="points" name="points" step="3">
</form>
```

L'attribut autofocus

L'attribut **autofocus** spécifie qu'un champ de saisie doit automatiquement obtenir le focus lorsque la page est chargée.

```
<form>
  <label for="fname">Nom :</label><br>
  <input type="text" id="fname" name="fname" autofocus><br>
  <label for="lname">Prénom :</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

L'attribut list

L'attribut **list** fait référence à un élément **<datalist>** qui contient des options prédéfinies pour un élément **<input>**.

```
<form>
  <input list="navigateurs">
  <datalist id="navigateurs">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>
```



CHAPITRE 2

Intégrer des formulaires dans une page web

1. Balise <form>
2. Champs de saisie (Input, TextArea)
3. Balises de choix (RadioButton, CheckBox, Select)
4. Balises HTML5 pour les formulaires
- 5. Validation automatique**
6. Boutons

02 - Intégrer des formulaires dans une page web

Validation automatique



Types de validation des formulaires HTML

La validation d'un formulaire consiste à vérifier si les données saisies par l'utilisateur sont correctes et conformes au fonctionnement attendu. Si ce n'est pas le cas, la page doit émettre des messages d'erreur.

Exemples de critères de validation :

- champ de saisi obligatoire
- format de l'email ou du numéro de téléphone correct
- taille et format d'un champ de mot de passe (nombre de caractères)
- présence d'un caractère spécial, lettre majuscule, ...)

Les types de validation :

La validation côté client : effectuée au niveau du navigateur, avant de soumettre les données au serveur. Cette validation peut être mise en œuvre de deux manières :

La validation JavaScript

La validation par les fonctions intégrées dans HTML5

La validation côté serveur : opérée sur le serveur, après soumission des données ont été soumises et avant l'enregistrement des données dans la base de données.

02 - Intégrer des formulaires dans une page web

Validation automatique



Validation des formulaires par les fonctions intégrées dans HTML5

1. Utilisation de l'attribut **required** : rendre une entrée obligatoire et empêcher la soumission du formulaire si l'entrée est vide.
2. Validation selon une expression régulière avec l'attribut **pattern** : utile pour valider le numéro de téléphone par exemple
3. Limitation de la taille des entrées avec les attributs **minlength** et **maxlength** (pour les champs input et textarea) et les attributs **min** et **max** pour le champ number
4. L'attribut **Novalidate** du formulaire (lorsqu'il est présent) précise que les données du form ne doivent pas être validées lors de son envoi



CHAPITRE 2

Intégrer des formulaires dans une page web

1. Balise <form>
2. Champs de saisie (Input, TextArea)
3. Balises de choix (RadioButton, CheckBox, Select)
4. Balises HTML5 pour les formulaires
5. Validation automatique
- 6. Boutons**

02 - Intégrer des formulaires dans une page web

Boutons



Type submit

<input type="submit"> définit un bouton pour soumettre des données de formulaire à un gestionnaire de formulaire.

Le gestionnaire de formulaires est généralement une page de serveur avec un script pour le traitement des données d'entrée.

```
<form action="/action_page.php">
    <label for="fname">Nom :</label><br>
    <input type="text" id="fname" name="fname"><br>
    <label for="lname">Prénom :</label><br>
    <input type="text" id="lname" name="lname"><br>
    <br>
    <input type="submit" value="Envoyer">
</form>
```

Nom :

Prénom :

Envoyer

Figure 31 : Bouton Submit

02 - Intégrer des formulaires dans une page web

Boutons

Type reset

`<input type="reset">` définit un bouton de réinitialisation qui réinitialisera toutes les valeurs du formulaire à leurs valeurs par défaut.

```
<form action="/action_page.php">
  <label for="fname">Nom :</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Prénom :</label><br>
  <input type="text" id="lname" name="lname"><br>
  <br>
  <input type="submit" value="Envoyer">
  <input type="reset" value="Réinitialiser">
</form>
```

Nom :

Prénom :

Envoyer

Réinitialiser

Figure 32 : Bouton reset



PARTIE 3

Mettre en forme une page web avec les feuilles de style CSS

Dans ce module, vous allez :

- Maitriser le CSS pour la mise en forme d'une page web
- Utiliser correctement des positionnements pour l'organisation d'une page web avec CSS
- Ajouter des animations
- Adapter des templates HTML/CSS à un site web statique



 30 heures



CHAPITRE 1

Introduire le CSS

Ce que vous allez apprendre dans ce chapitre :

- Présenter les éléments CSS
- Gérer les types d'intégration du CSS
- Utiliser les différents CSS



10 heures



CHAPITRE 1

Introduire le CSS

1. Présentation du CSS
2. Codification des couleurs
3. Unités de mesure
4. Positions (center, left, right)
5. Fonts
6. Types d'intégration du CSS
7. Sélecteurs simples (element, class, id)
8. Sélecteurs complexes
9. Pseudo classes

Application d'un style à un code HTML

Il existe 3 manières pour introduire le style à une balise donnée :

- Dans une balise, avec l'attribut **style**
- Dans l'en-tête, après la balise title (dans la partie head)
- Dans un fichier d'extension **.css**

La troisième méthode, basée sur l'utilisation du langage CSS dans un fichier séparé, est la technique recommandée

01 - Introduire le CSS

Présentation du CSS (Cascading Style Sheets)

Présentation du CSS

Le langage CSS (**Cascading Style Sheets**) est destiné à gérer le contenu et la structure d'une page web.

CSS permet de gérer la mise en forme du contenu d'une page : le texte (couleur, taille, casse, etc.), les images, les tableaux, les formulaires, etc.

CSS permet également de dimensionner et positionner les éléments dans la page web

CSS est utilisé aussi pour gérer les animations.



Figure 33 : Logo du CSS

Syntaxe du CSS

Une déclaration CSS a la forme suivante :

Le sélecteur est un élément clé qui indique sur quels éléments HTML vont s'appliquer les propriétés CSS définies entre { }

sélecteur {
propriété : valeur ;
propriété : valeur ;
}

C'est l'ensemble de paramètres de mise en forme à appliquer sur l'élément identifié par le sélecteur. Chaque propriété de mise en forme est suivie de deux points et de sa valeur.
Exemple de propriétés : la couleur, la taille, le style, etc.

Exemple :

```
/* Titres de niveau 1 */  
h1 {  
    color : red ;  
}
```

Dans cet exemple, on spécifie la couleur rouge comme couleur par défaut de tous les éléments h1.



CHAPITRE 1

Introduire le CSS

1. Présentation du CSS
2. **Codification des couleurs**
3. Unités de mesure
4. Positions (center, left, right)
5. Fonts
6. Types d'intégration du CSS
7. Sélecteurs simples (élément, class, id)
8. Sélecteurs complexes
9. Pseudo classes

Codification des couleurs (Rappel)

Les navigateurs admettent 3 méthodes possibles pour la déclaration d'une couleur :

Méthode	Exemple	Explication
Couleurs nommées	<u>background-color</u> : grey	La couleur est désignée par son nom en anglais. Il n'y a que 16 noms de couleurs normalisés par le W3C
Couleurs en hexadécimal	<u>background-color</u> : #808080	La couleur est désignée par son code hexadécimal : les deux premiers digits correspondent à la valeur de rouge, les deux suivants le vert et les deux derniers le bleu
Couleurs décomposées	<u>background-color</u> : RGB(128,128,128)	La couleur est définie par trois nombres décimaux qui peuvent prendre les valeurs de 0 à 255 indiquant respectivement le taux de rouge, le taux de vert et le taux de bleu.



CHAPITRE 1

Introduire le CSS

1. Présentation du CSS
2. Codification des couleurs
- 3. Unités de mesure**
4. Positions (center, left, right)
5. Fonts
6. Types d'intégration du CSS
7. Sélecteurs simples (élément, class, id)
8. Sélecteurs complexes
9. Pseudo classes

01 - Introduire le CSS

Unités de mesure



Les unités absolues

Les unités absolues ne sont influencées par aucune autre dimension. Elles sont utilisées pour exprimer une longueur, comme un padding, une hauteur de bloc, la taille des caractères, etc.

- **Les unités absolues sont les suivantes :**

- Les pixels (px)
- Les pouces (in)
- Les centimètres (cm)
- Les millimètres (mm)
- Les picas (pc)
- Les points (pt)

```
div
{
    width : 150px ;
    height : 150px ;
}
```

Le pixel et le point sont les unités les plus utilisées.

- **Relation entre les unités de mesure :**

- 1 in = 96 px
- 1 cm = 37,8 px
- 1 mm = 3,78 px.

01 - Introduire le CSS

Unités de mesure



Les unités relatives au texte

Les unités relatives font varier la taille d'un élément en fonction de la taille de la police ou de l'élément parent.

Les unités dites « relatives au texte » sont utilisées pour harmoniser les éléments d'un design. Ces unités sont les suivantes:

- **Em** : proportionnelle à la taille de la police de l'élément parent ou du document. Par défaut, 1 em = 16 px si aucune taille de police n'est définie.
- **Rem** : fait toujours référence à la taille de la police de l'élément racine. Elle dépend du font-size défini par défaut.
- **Ex** : relative à la hauteur de la police actuelle en minuscule (très rarement utilisée).
- **Ch** : relative à la largeur du caractère "0" (peu utilisée).

Les unités relatives au viewport

Les unités relatives au **viewport** permettent aux éléments de s'adapter à la taille de la fenêtre du navigateur.

Elle sont essentielles pour mettre en place un design responsive :

- La hauteur du viewport (vh)
- La largeur du viewport (vw)
- Le viewport minimum (vmin)
- Le viewport maximum (vmax)



CHAPITRE 1

Introduire le CSS

1. Présentation du CSS
2. Codification des couleurs
3. Unités de mesure
4. **Positions (center, left, right)**
5. Fonts
6. Types d'intégration du CSS
7. Sélecteurs simples (élément, class, id)
8. Sélecteurs complexes
9. Pseudo classes

Alignment du texte

La propriété **text-align** est utilisée pour définir l'alignement horizontal d'un texte.

Un texte peut être aligné à gauche ou à droite, centré ou justifié.

l'alignement à gauche est par défaut si la direction du texte est de gauche à droite, et l'alignement à droite est par défaut si la direction du texte est de droite à gauche

L'exemple suivant montre le texte aligné au centre et aligné à gauche et à droite :

```
h1
{
    text-align : center;
}
h2
{
    text-align : left;
}
h3
{
    text-align : right;
}
```

Alignment du texte

Lorsque la propriété **text-align** est définie sur la valeur "**justify**", toutes les lignes sont étirées de manière à avoir la même largeur et les marges gauche et droite sont droites (texte justifié) :

```
div
{
    text-align : justify;
```

Orientation du texte

Les propriétés **direction** et **unicode-bidi** sont utilisées pour changer la direction du texte d'un élément :

```
p  
{  
    direction : rtl;  
    unicode-bidi : bidi-override;  
}
```

Alignment vertical

La propriété **vertical-align** définit l'alignement vertical d'un élément.

```
img.a
{
    vertical-align : baseline;
}
img.b
{
    vertical-align : text-top;
}
img.c
{
    vertical-align : text-bottom;
}
img.d
{
    vertical-align : sub;
}
img.e
{
    vertical-align : super;
}
```



CHAPITRE 1

Introduire le CSS

1. Présentation du CSS
2. Codification des couleurs
3. Unités de mesure
4. Positions (center, left, right)
- 5. Fonts**
6. Type d'intégration CSS
7. Sélecteurs simples (élément, class, id)
8. Sélecteurs complexes
9. Pseudo classes

Propriétés du texte en CSS3

Taille du texte : font-size

Peut être spécifiée de 4 manières :

- **En pixels** : ou en mm, cm.
- **En valeur relative** : xx-small, x-small, small, medium, large, x-large xx-large.
- **En em** : par exemple, 1em signifie "Taille normale", 1.3em signifie un texte de taille 1,3 fois plus grande, 0.8em signifie que votre texte aura une taille 0,8 fois plus petite.
- **En %** : par exemple, 100% : le texte aura une taille "normale", 130% : le texte aura une taille correspondant à 130% de la taille normale

Exemple :

```
p
{
    font-size : 16px;
}
```

Propriétés du texte en CSS3

Type de la police: **font-family**

C'est une suite de noms de polices.

Cinq familles sont distinguées : serif (Times, etc.), sans-serif (Arial, Helvetica, etc.), cursive, monospace, Fantasy.

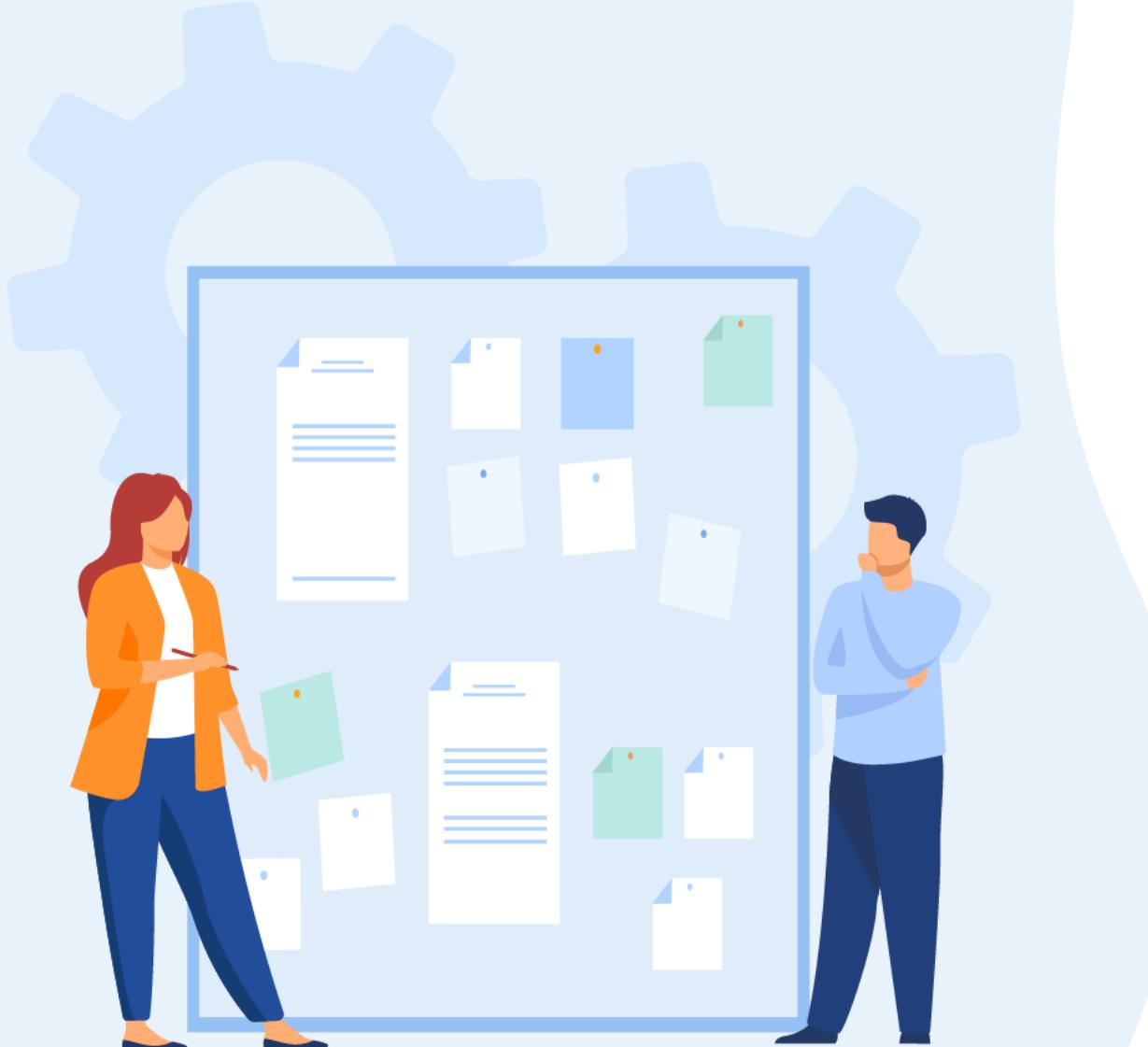
Les polices sont associées de préférence « en pile ». Dans l'exemple suivant, si le visiteur du site n'a pas la police Helvetica Neue sur sa machine, on passe automatiquement à la suivante et ainsi de suite.

```
p
{
    font-family:'Helvetica Neue', Helvetica,sans-serif;
}
```

Règle **@font-face** : permet d'importer un font (extension .ttf)

```
@font-face
{
    font-family: 'newFont';
    src: url('PARCHM.ttf');
}

p
{
    font-family:'newFont';
}
```



CHAPITRE 1

Introduire le CSS

1. Présentation du CSS
2. Codification des couleurs
3. Unités de mesure
4. Positions (center, left, right)
5. Fonts
6. **Types d'intégration du CSS**
7. Sélecteurs simples (élément, class, id)
8. Sélecteurs complexes
9. Pseudo classes

Dans la même page HTML

Pour intégrer une feuilles de style CSS dans une page HTML, il faut placer le code CSS dans la balise `<style>`.

Exemple :

```
<!DOCTYPE html>
<head>
<style>
#myDiv1
{
    width : 100px;
    height : 100px;
    border : 1px solid red;
    background-color : yellow;
}
</style>
</head>
<body>
    <div id="myDiv1"></div>
</body>
</html>
```

Dans un fichier externe

La balise `<link>` permet d'intégrer une feuilles de style CSS dans une page HTML.

La balise `<link>` doit être placée dans la balise `<head>`.

Exemple :

style.css

```
#myDiv1
{
    width : 100px;
    height : 100px;
    border : 1px solid red;
    background-color : yellow;
}
```

index.html

```
<!DOCTYPE html>
<head>
<link rel="stylesheet" href="style.css">
</head>
<body>
    <div id="myDiv1"></div>
</body>
</html>
```



CHAPITRE 1

Introduire le CSS

1. Présentation du CSS
2. Codification des couleurs
3. Unités de mesure
4. Positions (center, left, right)
5. Fonts
6. Types d'intégration du CSS
7. **Sélecteurs simples (élément, class, id)**
8. Sélecteurs complexes
9. Pseudo classes

Les sélecteurs de type

Ce sélecteur permet de cibler les éléments qui correspondent au nom indiqué. (nom de la balise)

Exemple :

Le sélecteur "div" permet de sélectionner tous les éléments "div" de la page web et leur appliquer le style.

```
div
{
    background-color : yellow;
```

Les sélecteurs de classe ":"

Ce sélecteur de type « class » permet de cibler les éléments en fonction de la valeur de leur attribut class.

Exemple :

```
<table border="1">
    <tr><th>Nom</th><th>Prénom</th><th>Age</th></tr>
    <tr class="impaire "><td>MANSOURI</td><td>Hassan</td><td>23</td></tr>
    <tr><td>SAFIR</td><td>Laila</td><td>22</td></tr>
    <tr class="impaire "><td>BICHRI</td><td>Karim</td><td>21</td></tr>
    <tr><td>HAMIM</td><td>Mohamed</td><td>24</td></tr>
</table>
```

Le code CSS suivant signifie que tous les éléments qui appartiennent à la classe "impaire" auront un arrière-plan jaune claire.

```
.impaire
{
    background-color : rgba(255, 255, 0, 0.2);
}
```

Nom	Prénom	Age
MANSOURI	Hassan	23
SAFIR	Laila	22
BICHRI	Karim	21
HAMIM	Mohamed	24

Figure 34: Exemple de classe CSS

Les sélecteurs d'identifiant "#"

Le sélecteur de type identifiant permet de cibler un élément d'un document en fonction de la valeur de son attribut id.

Dans un document, il ne doit y avoir qu'un seul élément pour un identifiant donné.

Exemple :

```
<table border="1">
    <tr><th>Nom</th><th>Prénom</th><th>Age</th></tr>
    <tr><td>MANSOURI</td><td>Hassan</td><td>23</td></tr>
    <tr><td>SAFIR</td><td>Laila</td><td>22</td></tr>
    <tr id="sel"><td>BICHRI</td><td>Karim</td><td>21</td></tr>
    <tr><td>HAMIM</td><td>Mohamed</td><td>24</td></tr>
</table>
```

Nom	Prénom	Age
MANSOURI	Hassan	23
SAFIR	Laila	22
BICHRI	Karim	21
HAMIM	mohamed	24

Le code CSS suivant signifie que l'élément de id="sel" aura un arrière plan jaune.

```
#sel
{
    background-color : rgba(255, 255, 0, 0.2);
}
```

Figure 35: Exemple de ID CSS



CHAPITRE 1

Introduire le CSS

1. Présentation du CSS
2. Codification des couleurs
3. Unités de mesure
4. Positions (center, left, right)
5. Fonts
6. Types d'intégration du CSS
7. Sélecteurs simples (élément, class, id)
- 8. Sélecteurs complexes**
9. Pseudo classes

Le sélecteur universel "*"

Ce sélecteur permet de cibler tous les nœuds d'un document.

Exemple :

le code suivant permet de colorier tous les textes du document en marron.

```
*  
{  
    color : brown;  
}
```

Les sélecteurs d'attribut "[]"

Ce sélecteur permet de cibler des éléments d'un document en fonction de la valeur d'un de leurs attributs.

- **[attr]** : cible un élément qui possède un attribut attr.
- **[attr=valeur]** : cible un élément qui possède un attribut attr dont la valeur est exactement valeur.
- **[attr~=valeur]** : cible un élément qui possède un attribut attr dont la valeur est égale à la valeur spécifiée.
- **[attr|=valeur]** : cible un élément qui possède un attribut attr dont la valeur est exactement égale à valeur ou dont la valeur commence par valeur suivi immédiatement d'un tiret.
- **[attr^=valeur]** : cible un élément qui possède un attribut attr dont la valeur commence par valeur.
- **[attr\$=valeur]** : cible un élément qui possède un attribut attr dont la valeur se termine par valeur.
- **[attr*=valeur]** : cible un élément qui possède un attribut attr et dont la valeur contient au moins une occurrence de valeur dans la chaîne de caractères.

Les sélecteurs d'attribut "[]"

Exemple :

```
/* Les éléments <a> avec un attribut title */
a[title] { color : purple; }

/* Les éléments <a> avec un href qui correspond à "https ://example.org" */
a[href="https ://example.org"] { color : green; }

/* Les éléments <a> dont href contient "example" */
a[href*="example"] { font-size : 2em; }

/* Les éléments <a> dont href finit par ".org" */
a[href$=".org"] { font-style : italic; }

/* Les éléments <a> dont l'attribut class contient le mot logo : comportement identique à a.logo */
a[class~= "logo"] { padding : 2px; }
```

Combinateur de voisin direct "+"

Ce combinateur permet de sélectionner un élément uniquement si :

- Cet élément « suit » un élément donné et que
- Les deux éléments sont les fils d'un même élément parent.

Exemple :

```
li :first-of-type + li
{
    color : red;
}
```

```
<ul>
    <li>Un</li>
    <li>Deux</li>
    <li>Trois</li>
</ul>
```

Résultat :

- Un
- Deux
- Trois

Sélecteurs de voisins généraux "~"

Le combinateur ~ permet de séparer deux sélecteurs et de cibler un élément si celui-ci correspond au second sélecteur et est précédé (mais pas forcément voisin immédiat) d'un autre élément correspondant au premier sélecteur qui partage le même parent.

Les deux éléments sont des fils d'un même parent élément, voisins immédiats ou pas.

Exemple :

```
p ~ span
{
    color : red;
```

```
<span>Ici, ce n'est pas rouge.</span>
<p>Voici un paragraphe.</p>
<code>Un peu de code.</code>
<span>Et un autre span.</span>
<code>Encore du code</code>
<span>Ici aussi, c'est rouge</span>
```

Résultat :

Ici, ce n'est pas rouge.

Voici un paragraphe.

Un peu de code. **Et un autre span.** Encore du code. **Ici aussi, c'est rouge**

Sélecteurs enfant ">"

Le combinateur > sépare deux sélecteurs et cible uniquement les éléments correspondant au second sélecteur qui sont des enfants directs des éléments ciblés par le premier sélecteur.

Exemple :

```
span
{
    background-color : white;
}
div > span
{
    background-color : blue;
```

```
<span>Ici, ce n'est pas rouge.</span>
<p>Voici un paragraphe.</p>
<code>Un peu de code.</code>
<span>Et un autre span.</span>
<code>Encore du code</code>
<span>Ici aussi, c'est rouge</span>
```

Résultat :

Ici, ce n'est pas rouge.

Voici un paragraphe.

Un peu de code. Et un autre span. Encore du code. Ici aussi, c'est rouge

Sélecteurs descendant " " (espace)

Le combinateur de descendance, représenté par un blanc (ou plusieurs blancs à la suite), permet de combiner deux sélecteurs afin de cibler les éléments qui correspondent au second sélecteur uniquement si ceux-ci ont un élément ancêtre qui correspond au premier sélecteur.

Exemple :

```
li
{
    list-style-type : disc;
}
li li
{
    list-style-type : circle;
}
```

Résultat :

- Élément 1
 - Sous-élément A
 - Sous-élément B
- Élément 2
 - Sous-élément A
 - Sous-élément B

```
<ul>
    <li>
        <div>Élément 1</div>
        <ul>
            <li>Sous-élément A</li>
            <li>Sous-élément B</li>
        </ul>
    </li>
    <li>
        <div>Élément 2</div>
        <ul>
            <li>Sous-élément A</li>
            <li>Sous-élément B</li>
        </ul>
    </li>
</ul>
```



CHAPITRE 1

Introduire le CSS

1. Présentation du CSS
2. Codification des couleurs
3. Unités de mesure
4. Positions (center, left, right)
5. Fonts
6. Types d'intégration du CSS
7. Sélecteurs simples (élément, class, id)
8. Sélecteurs complexes
9. **Pseudo classes**

Pseudo-classes

Une **pseudo-classe** est un mot-clé qui peut être ajouté à un sélecteur afin d'indiquer l'état spécifique dans lequel l'élément doit être pour être ciblé par la déclaration.

Exemple :

- **hover** permettra d'appliquer une mise en forme spécifique lorsque l'utilisateur survole l'élément ciblé par le sélecteur (changer la couleur d'un bouton par exemple).

Liste des pseudo-classes

Sélecteur	Exemple	Description
: checked	input :checked	Sélectionne tout élément <input> coché du formulaire
: disabled	input :disabled	Sélectionne chaque élément <input> désactivé
: empty	p :empty	Sélectionne chaque élément <p> qui n'a pas d'enfant
: enabled	input :enabled	Sélectionne chaque élément <input> activé
: first-of-type	p :first-of-type	Sélectionne chaque élément <p> qui est le premier élément <p> de son parent
: in-range	input :in-range	Sélectionne les éléments <input> avec une valeur située dans une plage spécifiée
: invalide	input :invalid	Sélectionne tous les éléments <input> avec une valeur invalide
: last-child	p :last-child	Sélectionne tous les éléments <p> qui sont les derniers enfants de leurs parents
: last-of-type	p :last-of-type	Sélectionne chaque élément <p> qui est le dernier élément <p> de son parent
: not(sélecteur)	:not(p)	Sélectionne tous les éléments qui ne sont pas des paragraphes
: nth-child (n)	p :nth-child(2)	Sélectionne chaque élément <p> qui est le deuxième enfant de son parent
: nth-last-child (n)	p :nth-last-child(2)	Sélectionne chaque élément <p> qui est le deuxième enfant de son parent, à partir du dernier enfant
: nth-last-of-type(n)	p :nth-last-of-type(2)	Sélectionne chaque élément <p> qui est le deuxième élément <p> de son parent, à partir du dernier enfant
: nth-of-type (n)	p :nth-of-type(2)	Sélectionne chaque élément <p> qui est le deuxième élément <p> de son parent
: only-of-type	p :only-of-type	Sélectionne chaque élément <p> qui est le seul élément <p> de son parent
: only-child	p :only-child	Sélectionne chaque élément <p> qui est le seul enfant de son parent
: optional	input :optional	Sélectionne les éléments <input> sans attribut "required"
: out-of-range	input :out-of-range	Sélectionne les éléments <input> qui ont des valeurs en dehors d'une plage spécifiée

Liste des pseudo-classes

Sélecteur	Exemple	Description
:read-only	input :read-only	Sélectionne les éléments <input> avec un attribut "readonly" spécifié
:read-write	input :read-write	Sélectionne les éléments <input> sans attribut "readonly"
:required	input :required	Sélectionne les éléments <input> avec un attribut "required" spécifié
root	root	Sélectionnez l'élément racine du document
:target	#news :target	Sélectionne l'élément #news actuellement actif
:valid	input :valid	Sélectionne tous les éléments <input> avec une valeur valide
:link	a :link	Sélectionne tous les liens non visités
:visited	a :visited	Sélectionnez tous les liens visités
:active	a :active	Sélectionne le lien actif
:hover	a :hover	Sélectionne les liens sur lesquels la souris passe
:focus	input :focus	Sélectionne l'élément <input> qui a le focus
:first-letter	p :first-letter	Sélectionne la première lettre de chaque élément <p>
:first-line	p :first-line	Sélectionne la première ligne de chaque élément <p>
:first-child	p :first-child	Sélectionne tous les éléments <p> qui sont les premiers enfants de leurs parents
:before	p :before	Insérer du contenu avant chaque élément <p>
:after	p :after	Insérer du contenu après chaque élément <p>
:lang(it)	p :lang(it)	Sélectionne chaque élément <p> avec une valeur d'attribut lang commençant par "it"

Pseudo-classes

Exemple : `:visited`

```
<p><a href="https://www.google.com">Cette page est visitée</a></p>
<p><a href="https://www.ofppt.ma">Essayez celle-ci peut-être ?</a></p>
```

```
a :visited
{
    color : #00ff00;
}
```

Cette page est visitée

Essayez celle-ci peut-être ?

Exemple : `:hover`

```
<p>
    <a href="#">Ce lien sera écrit sur un fond doré lors du survol.</a>
</p>
```

```
a :hover
{
    background-color : gold;
}
```

Ce lien sera écrit sur un fond doré lors du survol.



Ce lien sera écrit sur un fond doré lors du survol.

Figure 36: Exemple de Pseudo-classe CSS

01 - Introduire le CSS

Les sélecteurs (pseudo-classes)



Pseudo-classes

Exemple : `:focus`

```
<input class="prenom" value="Rouge si focus">
<input class="nom" value="Vert si focus">
```

Rouge si focus

Vert si focus

Rouge si focus

Vert si focus

Figure 35: Exemple de Pseudo-class CSS

Exemple : `:invalid`

```
<input type="email">
```

adil@gmail.com

Avec Email faux

adil@gmail.com

Avec Email juste

Figure 37: Exemple de Pseudo-classe CSS

```
.prenom :focus
{
    background : yellow;
    color : red;
}
```

```
.nom :focus
{
    background : yellow;
    color : lime;
}
```

```
input :invalid
{
    border : 2px solid red;
}
```

CHAPITRE 2

Utiliser les propriétés CSS



Ce que vous allez apprendre dans ce chapitre :

- Exploiter les propriétés CSS permettant de définir les règles typographiques et de mise en page
- Utiliser les propriétés de positionnement afin de structurer les contenus



10 heures



CHAPITRE 2

Utiliser les propriétés CSS

1. **Typographie**
2. Bordures et ombres
3. Marges et padding
4. Images
5. Couleurs de fond
6. Background
7. Types de positionnement (relatif, absolu, float...)
8. Utilisation des blocs Flex (FlexBox)

Propriétés du texte en CSS3

- Propriété **font-style** : spécifie un texte en italique. Valeurs possibles : **normal, italic, oblique**.
- Propriété **font-variant** : spécifie si un texte doit être affiché en petites capitales. Valeurs possibles : **normal, small-caps**.
- Propriété **font-weight** : permet de mettre en gras un texte. Valeurs possibles : **de 100 à 900, normal, bold, bolder, lighter**.
- Propriété **white-space** : spécifie l'espace blanc à l'intérieur d'un élément. Valeurs possibles : **normal, pre, nowrap, pre-wrap, pre-line**
- Propriété **word-spacing** : l'espace entre les mots (en pixels).
- Propriété **letter-spacing** : l'espace entre deux lettres au sein d'un même mot (en pixels).
- Propriété **text-decoration** : spécifie la décoration du texte. Valeurs possibles : **none, underline, overline , line-through, blink**.
- Propriété **text-transform** : permet de contrôler la capitalisation du texte. Valeurs possible : **none, uppercase (majuscules), lowercase (minuscule), capitalize (la première lettre de chaque mot sera en majuscule)**.
- Propriété **text-align** : permet d'aligner horizontalement le texte. Valeurs possibles: **left, center, right, justify**
- Propriété **text-indent** : taille du retrait de première ligne (en pixels)
- Propriété **line-height** : hauteur de ligne (en pixels).
- Propriété **vertical-align** : permet d'aligner verticalement le texte. Valeurs possibles : **baseline, super, top, text-top, middle, text-bottom, bottom, sub, inherit**

Propriétés du texte en CSS3

Exemple : les liens

```
a {  
    text-decoration: none; /* Les liens ne seront plus soulignés */  
    color: red; /* Les liens seront en rouge au lieu de bleu */  
    font-style: italic; /* Les liens seront en italique */  
}  
  
a:hover /* Quand le visiteur pointe sur le lien */  
{  
    text-decoration: underline; /* Le lien deviendra souligné quand on pointera dessus */  
    color: green; /* Le lien sera écrit en vert quand on pointera dessus */  
    background-color: #CFE1EB; /* Le fond du paragraphe change de couleur */  
}
```



CHAPITRE 2

Utiliser les propriétés CSS

1. Marges et padding
2. **Bordures et ombres**
3. Marges et padding
4. Images
5. Couleurs de fond
6. Background
7. Types de positionnement (relatif, absolu, float...)
8. Utilisation des blocs Flex (FlexBox)

02 - Utiliser les propriétés CSS

Bordures et ombres

Bordures et ombres

```
<div id="myDiv1"></div>
```

```
div
{
    width: 100px;
    height: 100px;
    background-color: yellow;
    border: 1px dashed red;
    border-radius: 10px;
    box-shadow: 10px 10px 5px gray;
}
```

Ligne avec tirets.
Autres types :
Solid, dotted, ...

Rayon de l'arrondi

Résultat :



Distance horizontale

Rayon de l'arrondi

```
.class
{
    text-shadow: 2px 2px 3px #FFF;
```

Distance verticale

Couleur de l'ombre

Figure 38: Exemple de bordure et ombre CSS



CHAPITRE 2

Utiliser les propriétés CSS

1. Typographie
2. Bordures et ombres
- 3. Marges et padding**
4. Images
5. Couleurs de fond
6. Background
7. Types de positionnement (relatif, absolu, float...)
8. Utilisation des blocs Flex (FlexBox)

Marge et padding

La propriété margin (retrait extérieur) définit la taille des marges sur les quatre côtés de l'élément.

C'est une propriété raccourcie qui permet de manipuler les autres propriétés de marges :

- margin-top
- margin-right
- margin-bottom
- margin-left

```
.ex1
{
    margin : auto;
    background : gold;
    width : 66%;
}
.ex2
{
    margin : 20px 0px 0px -20px;
    background : gold;
    width : 66%;
}
```

```
<div class="ex1">
    margin : auto;
    background : gold;
    width : 66%;
</div>

<div class="ex2">
    margin : 20px 0px 0px -20px;
    background : gold;
    width : 66%;
</div>
```

02 - Utiliser les propriétés CSS

Marging et padding



Marging et padding

La propriété **padding** (retrait intérieur) définit les différents écarts de remplissage sur les quatre côtés d'un élément (cf. les boîtes CSS). Elle synthétise **padding-top**, **padding-right**, **padding-bottom**, **padding-left**.

```
h4
{
    background-color : green;
    padding : 50px 20px 20px 50px;
}
h3
{
    background-color : blue;
    padding : 400px 5%;
}
```

Résultat :

```
<h4>Coucou le monde !</h4>
<h3>Le remplissage n'est pas le même ici.</h3>
```

Coucou le monde !

Le remplissage n'est pas le même ici.



CHAPITRE 2

Utiliser les propriétés CSS

1. Typographie
2. Bordures et ombres
3. Marges et padding
- 4. Images**
5. Couleurs de fond
6. Background
7. Types de positionnement (relatif, absolu, float...)
8. Utilisation des blocs Flex (FlexBox)

Propriétés des images

- Propriétés **width** et **height** : définissent la taille de l'image
- Propriété **max-width** : permet à l'image de s'adapter à la taille de l'écran (Exemple 1)
- Propriétés **margin-left** et **margin-right**: ayant la valeur auto, elles permettent de centrer une image (Exemple 2)
- Propriété **opacity** : gère l'opacité de l'image (Exemple 3)
- Propriété **filter** : ajoute des effets visuels (comme le flou et la saturation) à l'image (Exemple 4).

```
#img
{
  max-width: 100%;
  height: auto;
}
```

Exemple 1

```
img
{
  display: block;
  margin-left: auto;
  margin-right: auto;
  width: 50%;
}
```

Exemple 2

```
#img
{
  opacity : 0.5;
}
```

Exemple 3

```
img
{
  filter: grayscale(100%);
}
```

Exemple 4



CHAPITRE 2

Utiliser les propriétés CSS

1. Typographie
2. Bordures et ombres
3. Marges et padding
4. Images
5. **Couleurs de fond**
6. `background`
7. Types de positionnement (relatif, absolu, float...)
8. Utilisation des blocs Flex (FlexBox)

02 - Utiliser les propriétés CSS

Couleurs de fond



Couleurs

Propriété **color** définit la couleur du texte

Propriété **background-color** : définit la couleur du fond

Exemple :

```
p
{
    color : blue;
    font-size : 24px;
    font-family : cursive;
    background-color : yellow;
}
```

```
<p>Ceci est un texte colorié en bleu</p>
```

Ceci est un texte colorié en bleu

Figure 39: propriété color en CSS



CHAPITRE 2

Utiliser les propriétés CSS

1. Typographie
2. Bordures et ombres
3. Marges et padding
4. Images
5. Couleurs de fond
6. **background**
7. Types de positionnement (relatif, absolu, float...)
8. Utilisation des blocs Flex (FlexBox)

02 - Utiliser les propriétés CSS

Background

Background

L'élément image peut être inséré comme arrière plan d'un élément.

Les propriétés associées sont :

- **background-image** : url de l'image de fond
- **background-repeat** : mode de réplication de l'image de fond (**repeat, repeat-x, repeat-y, no-repeat**)
- **background-attachment** : défilement ou non de l'image de fond (**scroll, fixed**)
- **background-position** : position de l'image d'arrière plan (**top, bottom, left, center, right**)
- **background-size** (100 % 100 %)

Exemple :

```
#myDiv1
{
    height : 100px;
    width : 100px;
    border : 1px solid black;
    background-image : url("corona.jpg");
    background-size :100%;
    background-repeat :no-repeat;
}
```

```
<div id="myDiv1"></div>
```



Figure 40: Exemple de background image en CSS



CHAPITRE 2

Utiliser les propriétés CSS

1. Typographie
2. Bordures et ombres
3. Marges et padding
4. Images
5. Couleurs de fond
6. Background
7. **Types de positionnement (relatif, absolu, float...)**
8. Utilisation des blocs Flex (FlexBox)

Types de positionnement

Le principe du positionnement en CSS permet de définir l'emplacement des boîtes générées pour chaque élément de la page

Il existe trois mécanismes de positionnement de base en CSS :

1. **Le flux normal** : consiste à garder le flux normal des éléments selon leur ordre et leur type (bloc , inline)

2. **La propriété « position » :**

- Relative
- Absolue
- Fixe
- Static

3. **La propriété « float » :**

- Définit la position flottante d'un élément à droite (valeur right) ou à gauche (valeur left) dans un document (ou dans la boîte conteneur).
- Nettoyer les flottants : propriété clear

02 - Utiliser les propriétés CSS

Types de positionnement

Le positionnement absolu

Ce type de positionnement permet de placer un élément de la page exactement à l'emplacement souhaité.

- Une balise HTML avec une position absolue ne laisse aucun espace vide après qu'elle est positionnée
- On place une balise HTML en position absolue en fixant la valeur de la propriété position à “**absolute**”
- Les propriétés **right**, **left**, **top** et **bottom** sont alors utilisées pour définir l'emplacement de la balise HTML

Exemple :

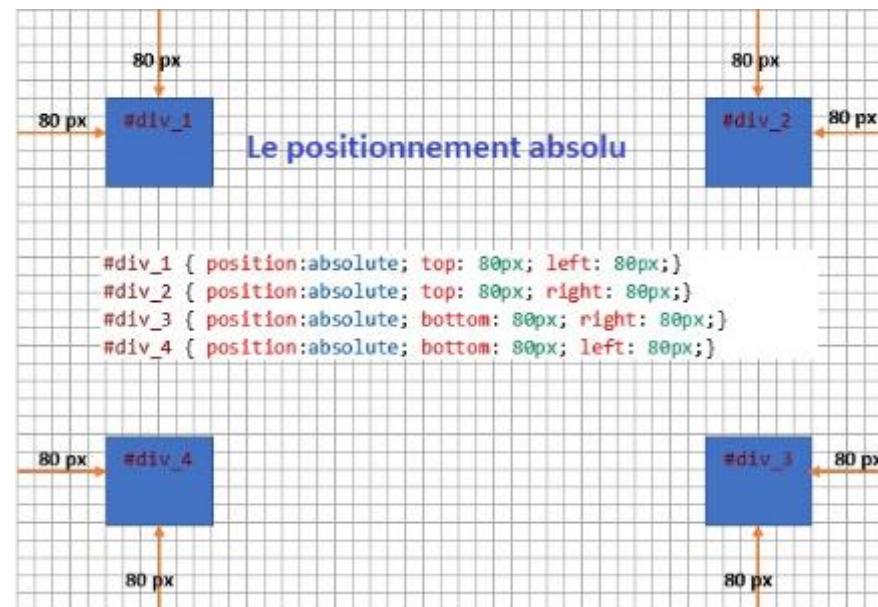


Figure 41: positionnement absolu en CSS

Le positionnement relatif

On place une div en position relative avec la valeur de la propriété : **relative**

La position relative d'un élément (div par exemple) est calculée d'après sa position originale dans la page (sa position dans le flux normal)

l'élément peut être déplacé vers la droite, la gauche, le haut ou le bas (**top, left, bottom et right**)

Le positionnement fixe

Pareil que le positionnement absolu, mais le block reste toujours visible, même si on défile la page

Le positionnement static

C'est le comportement normal (par défaut) pour chaque élément de page. L'élément est alors positionné dans le flux avec sa position.

Les propriétés **top, right, bottom, left** et z-index ne s'appliquent pas.

La valeur **static** est la valeur par défaut de la propriété position

Un élément HTML positionné avec position : static sera positionné selon le flux normal de la page

La propriété CSS float

La propriété **float** est utilisée pour positionner et formater le contenu.

Par exemple, faire flotter une image à gauche du texte dans un conteneur.

- La propriété **float** peut avoir les valeurs suivantes :
 - **left** : L'élément flotte à gauche de son conteneur.
 - **right** : L'élément flotte à droite de son conteneur.
 - **none** : La valeur par défaut, l'élément ne flotte pas.
 - **inherit** : L'élément hérite de la valeur flottante de son parent.

02 - Utiliser les propriétés CSS

Types de positionnement

La propriété CSS float

Exemple 1 : sans float

```
img
{
    width : 100px;
    height : 100px;
}
```



Si la COVID-19 se propage dans votre communauté, protégez-vous en prenant quelques précautions Simples, comme maintenir une distance physique avec autrui, porter un masque, bien ventiler les pièces, éviter les rassemblements, vous laver les mains et tousser dans votre coude replié ou un mouchoir. Suivez Les recommandations locales là où vous vivez et travaillez. Faites tout cela !

Figure 42: flux normal

Exemple 2 : avec float = left

```
img
{
    width : 100px;
    height : 100px;
    float : left;
}
```



Si la COVID-19 se propage dans votre communauté, protégez-vous en prenant quelques précautions Simples, comme maintenir une distance physique avec autrui, porter un masque, bien ventiler les pièces, éviter les rassemblements, vous laver les mains et tousser dans votre coude replié ou un mouchoir. Suivez Les recommandations locales là où vous vivez et travaillez. Faites tout cela !

Figure 43: Propriété float en CSS



CHAPITRE 2

Utiliser les propriétés CSS

1. Typographie
2. Bordures et ombres
3. Marges et padding
4. Images
5. Couleurs de fond
6. Background
7. Types de positionnement (relatif, absolu, float, ...)
8. **Utilisation des blocs Flex (FlexBox)**

Pourquoi utiliser Flexbox ?

Avant l'insertion de la méthode Flexbox, il existait quatre modes de mise en page :

- **Block**, pour les sections d'une page Web
- **Inline**, pour le texte
- **Tableau**, pour les données de tableau à deux dimensions
- **Position**, pour la position explicite d'un élément

La méthode **Flexbox** facilite la structuration et la mise en page souple et réactive, sans utiliser les marges et le positionnement.

La page est décomposée en un agencement de boites qui peuvent se suivre (boites voisines) et s'imbriquer (boites ancêtres).

En définissant les tailles et les interactions entre les différentes boîtes, on obtient une adaptation automatique des éléments de la page lorsqu'elle est redimensionnée.

Comment utiliser Flexbox ?

Le conteneur **flex (flex container)** ou bien la boite dite « parent », contient les éléments **flex (flex items)** ou bien les boites « enfants ».

L'idée est définir, en utilisant les différentes propriétés CSS, l'ordre des "boites enfants" dans la "boite parent".

02 - Utiliser les propriétés CSS

Les propriétés des conteneurs flex (FlexBox)



Définir le conteneur flex (flex container)

Le conteneur est une balise html qui contient d'autres balises html.

Dans l'exemple ci-dessous, La conteneur correspond à la balise <div> pour laquelle on a défini la classe « **.conteneur** ».

Cette balise <div> contient deux balises <div> et une balise <figure>. Ces dernières deviennent alors des "éléments flex" (flex items) mis en évidence ici par la classe **".element"**.

```
<div class="conteneur">
    <div class="element">Élément 1</div>
    <figure class="element">Élément 2</figure>
    <div class="element">Élément 3</div>
</div>
```

La boite se transforme en "conteneur flex" dès lors qu'on lui attribue la propriété CSS "flex" comme montré dans le code CSS suivant :

```
.conteneur
{
    display : flex;
}
```

Définir les axes du conteneur flex avec la propriété flex-direction

Axe principal : main axis

Le conteneur flex possède un axe principal (main axis) sur lequel les éléments flex peuvent se suivre.

L'axe principal est défini par la propriété **flex-direction** qui peut prendre les valeurs suivantes :

- **row** : les éléments flex se suivent sur une ligne, ce qui correspond à la direction normale du sens d'écriture du document
- **row-reverse** : les éléments flex se suivent sur une ligne en ordre inversé. ce qui correspond à la direction inverse du sens d'écriture du document
- **column** : les éléments flex se suivent sur une colonne. **column-reverse** : les éléments flex se suivent sur une colonne en ordre inversé

02 - Utiliser les propriétés CSS

Les propriétés des conteneurs flex (FlexBox)

Définir les axes du conteneur flex avec la propriété flex-direction

Axe secondaire : cross axis

Le choix de l'axe principal impose la direction de l'axe secondaire (**cross axis**) qui lui est toujours perpendiculaire.

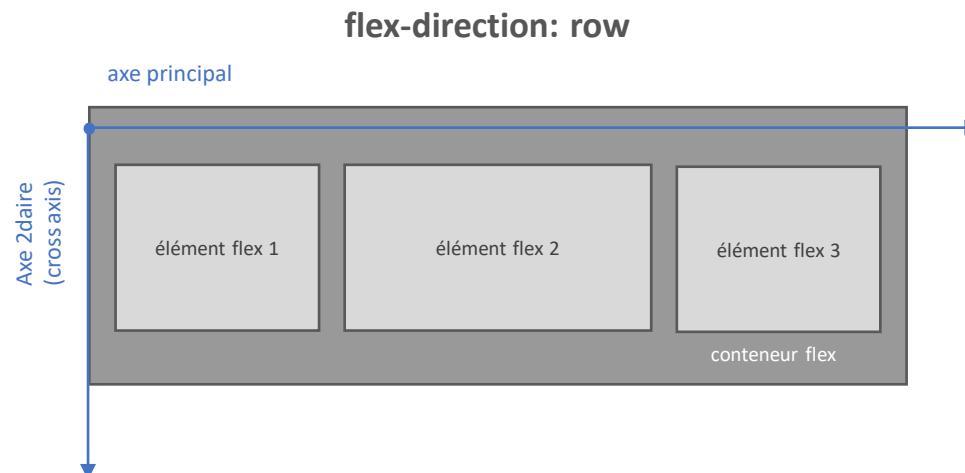


Figure 44: Exemple de flex-direction row

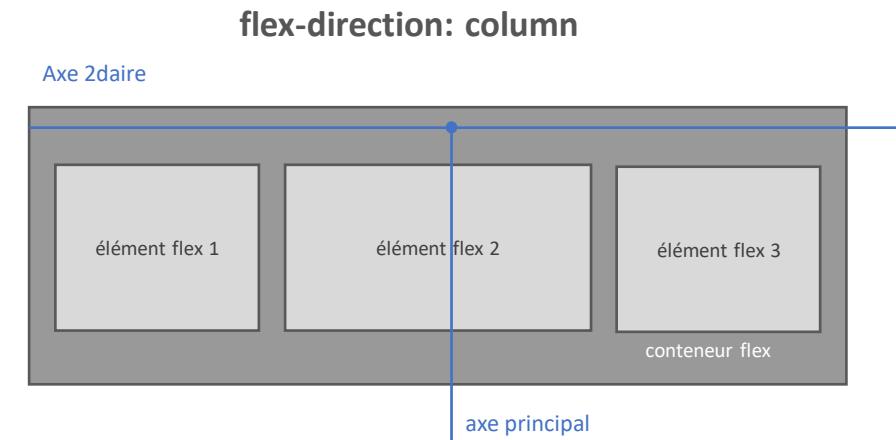


Figure 45: Exemple de flex-direction column

Le choix des axes est essentiel car leur orientation va déterminer le résultat de l'application des propriétés suivantes :

justify-content : définit comment les éléments flex sont positionnés le long de l'axe principal ;

align-items : définit comment les éléments flex sont positionnés le long de l'axe secondaire ;

align-self : définit comment un seul élément flex est positionné le long de l'axe secondaire.

Positionner les éléments à l'intérieur d'un conteneur flex

Les éléments flexibles peuvent être alignés, justifiés et distribués le long des axes de leur conteneur grâce à différentes propriétés.

Dans l'exemple suivant, on a choisi une direction "row", ce qui signifie que l'axe principal est l'axe horizontal et que l'axe secondaire est l'axe vertical.

```
<div class="conteneur">
    <div class="element1">élément 1</div>
    <div class="element2">élément 2</div>
    <div class="element3">élément 3</div>
    <div class="element4">élément 4</div>
</div>
```

```
.conteneur
{
    background-color : blue;
    height : 230px;
    width : 500px;
    display : flex;
    flex-direction : row;
    align-items : stretch | flex-start | flex-end | center | baseline;
    justify-content : flex-start | flex-end | center | space-around | space-between | space-evenly;
}
```

Aligner le long de l'axe principal : la propriété "justify-content"

La propriété "justify-content" permet d'aligner les éléments le long de l'axe principal dans la direction définie par "flex-direction".

Les valeurs de la propriété "justify-content" : flex-start, flex-end, center, space-around, space-between et space-evenly.

Exemple 1 : justify-content : flex-start → les éléments flexibles sont placés à partir de la ligne de début du conteneur sur l'axe principal.



Figure 46: Exemple de la propriété justify-content : flex-start

02 - Utiliser les propriétés CSS

Les propriétés des conteneurs flex (FlexBox)



Aligner le long de l'axe principal : la propriété « justify-content »

Exemple 2 : `justify-content : flex-end` → les éléments flexibles sont placés à partir de la ligne de fin du conteneur sur l'axe principal.



Figure 47: Exemple de la propriété `justify-content : flex-end`

Exemple 3 : `justify-content : center` → les éléments flexibles sont centrés le long de l'axe principal.



Figure 48: Exemple de la propriété `justify-content : center`

02 - Utiliser les propriétés CSS

Les propriétés des conteneurs flex (FlexBox)

Aligner le long de l'axe principal : la propriété "justify-content"

Exemple 4 : justify-content : space-between → l'espace disponible est réparti de façon égale entre chaque élément.



Figure 49: Exemple de la propriété justify-content : space-between

Exemple 5 : justify-content : space-around → l'espace disponible est réparti de façon égale entre chaque élément, y compris au début et à la fin.



Figure 50: Exemple de la propriété justify-content : space-around

02 - Utiliser les propriétés CSS

Les propriétés des conteneurs flex (FlexBox)



WEBFORCE
BE THE CHANGE

Aligner le long de l'axe principal : la propriété "justify-content"

Exemple 6 : justify-content : space-evenly → l'espace disponible est réparti de façon égale entre chaque élément avec un espace entier au début et à la fin.

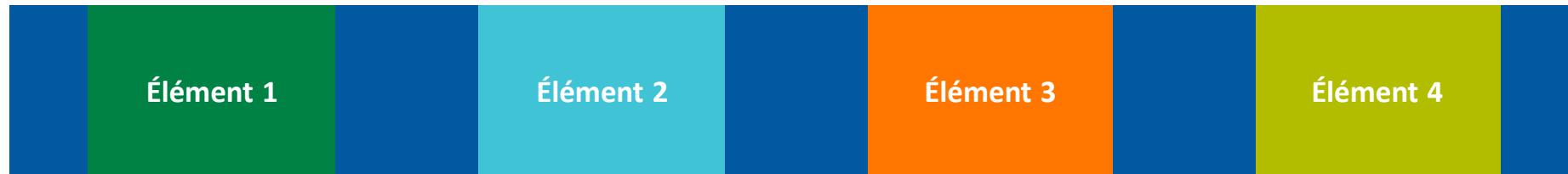


Figure 51: Exemple de la propriété justify-content : space-evenly

02 - Utiliser les propriétés CSS

Les propriétés des conteneurs flex (FlexBox)



WEBFORCE
BE THE CHANGE

Aligner le long de l'axe secondaire : la propriété "align-items"

La propriété "align-items" peut prendre 5 valeurs : **stretch**, **flex-start**, **flex-end**, **center** et **baseline**.

Exemple 1 : align-items : stretch → les éléments flexibles sont étirés le long de l'axe secondaire.

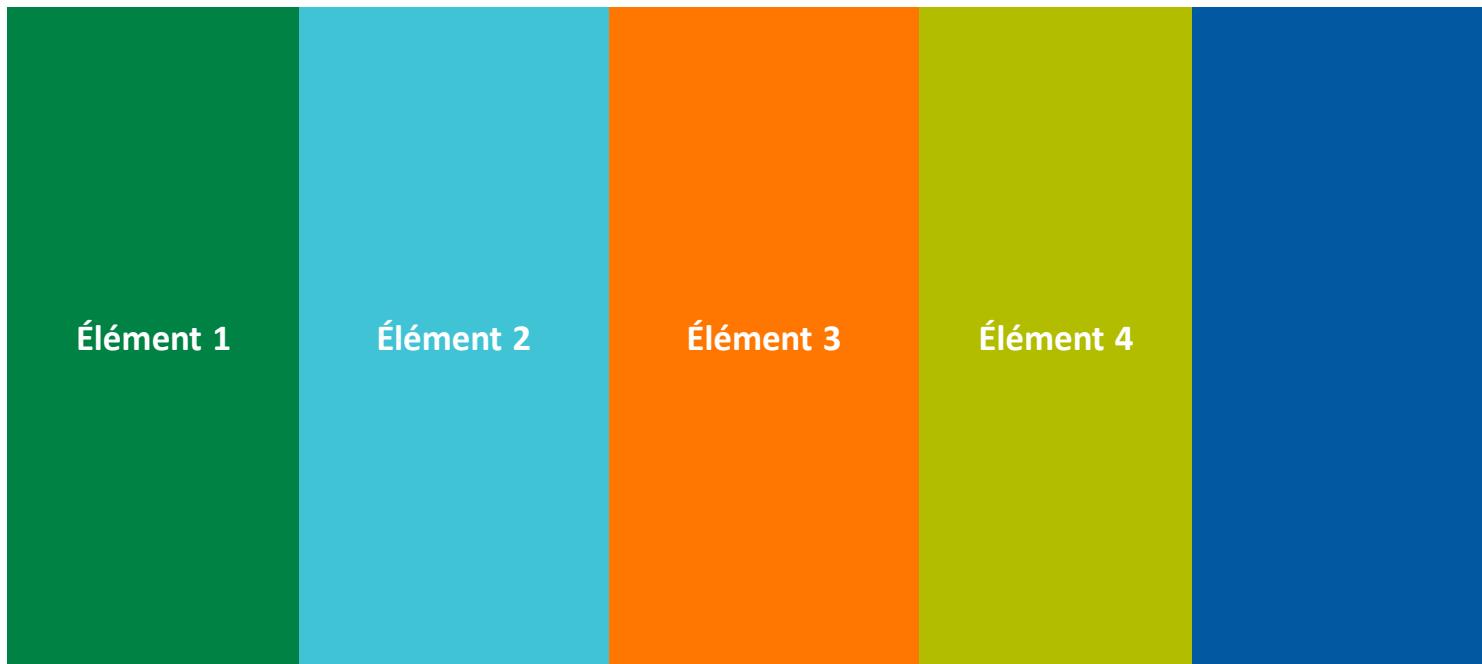


Figure 52: Exemple de la propriété align-items : stretch

02 - Utiliser les propriétés CSS

Les propriétés des conteneurs flex (FlexBox)

Aligner le long de l'axe secondaire : la propriété "align-items"

Exemple 2 : align-items : flex-start → les éléments flexibles sont alignés sur la ligne de début de l'axe secondaire.

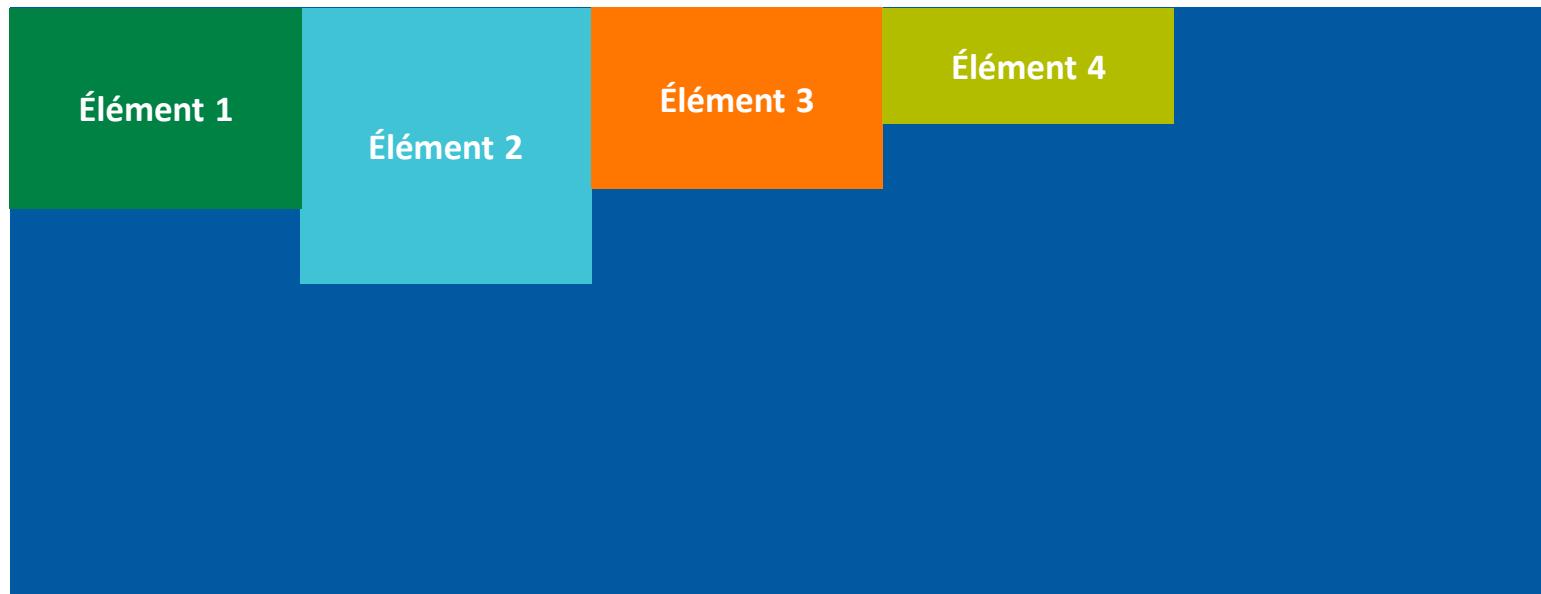


Figure 53: Exemple de la propriété align-items : flex-start

02 - Utiliser les propriétés CSS

Les propriétés des conteneurs flex (FlexBox)

Aligner le long de l'axe secondaire : la propriété "align-items"

Exemple 3 : align-items : flex-end → les éléments flexibles sont alignés sur la ligne de fin de l'axe secondaire.

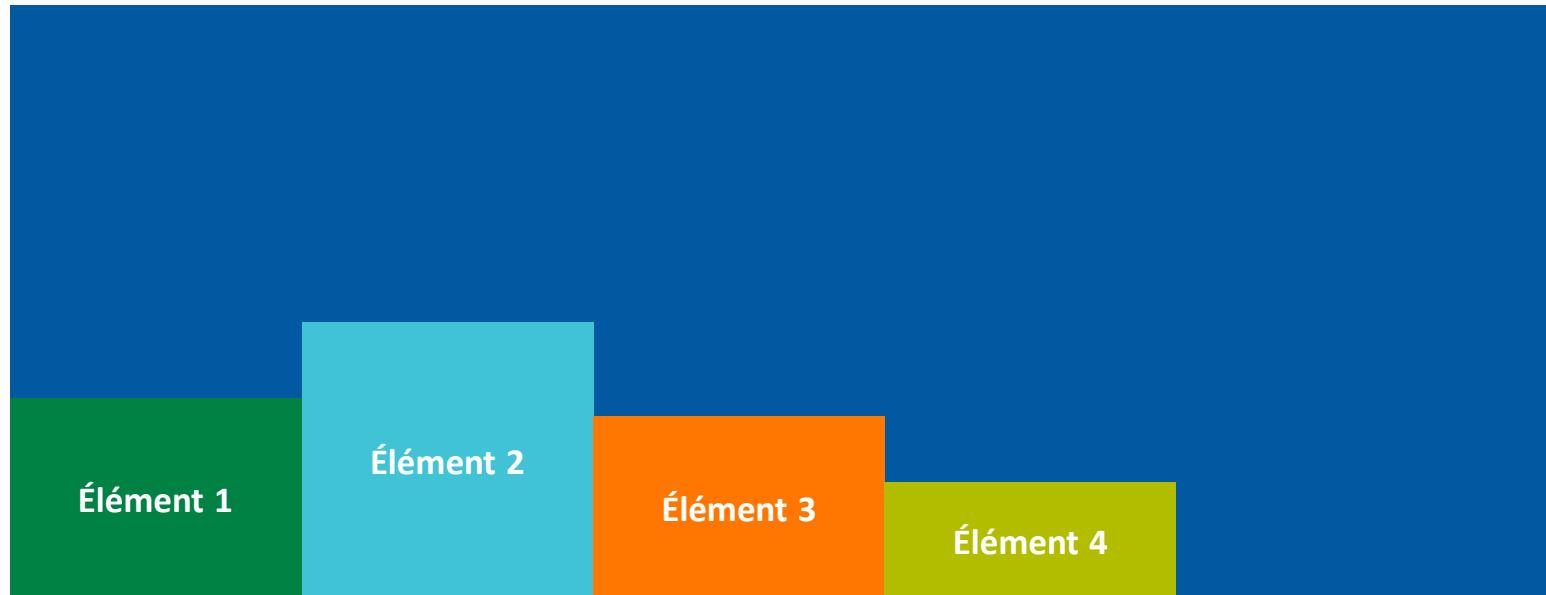


Figure 54: Exemple de la propriété align-items : flex-end

02 - Utiliser les propriétés CSS

Les propriétés des conteneurs flex (FlexBox)

Aligner le long de l'axe secondaire : la propriété "align-items"

Exemple 4 : align-items : center → les éléments flexibles sont centrés sur la ligne d'axe secondaire.

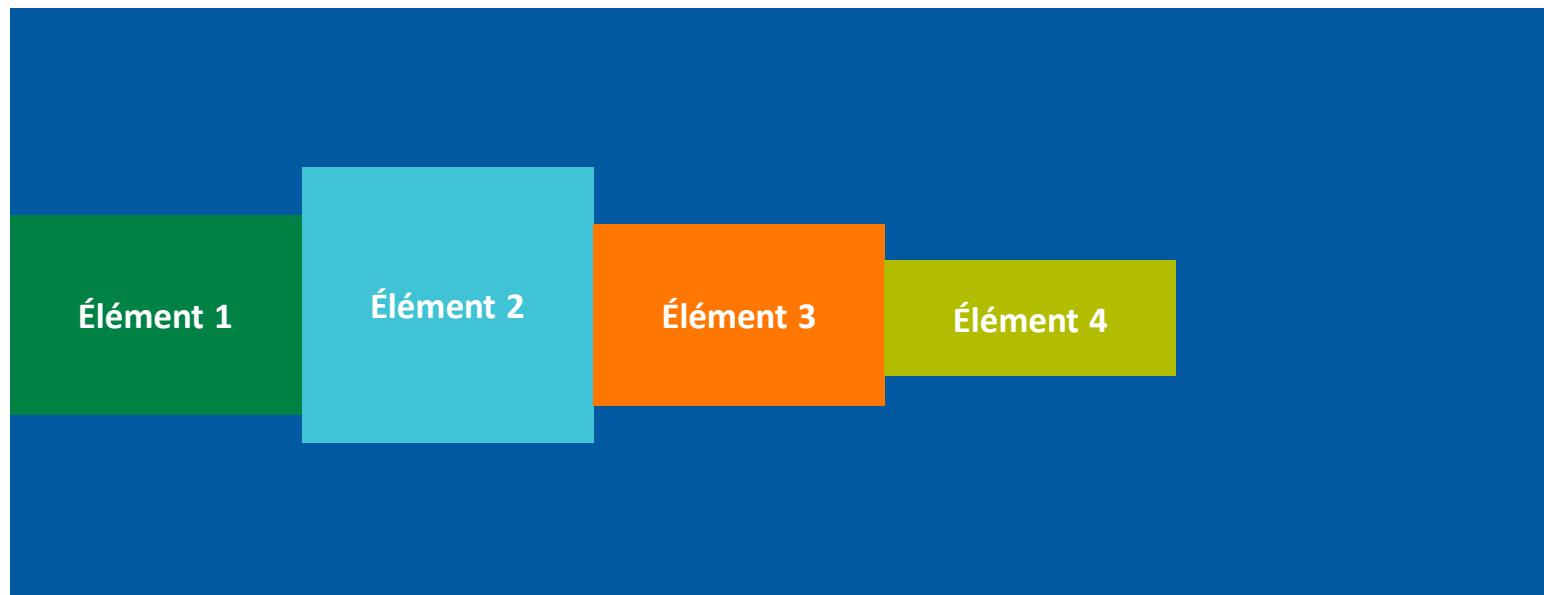


Figure 55: Exemple de la propriété align-items : center

02 - Utiliser les propriétés CSS

Les propriétés des conteneurs flex (FlexBox)

Aligner le long de l'axe secondaire : la propriété "align-items"

Exemple 5 : align-items : baseline → les éléments flexibles sont alignés sur leur ligne de base (les textes sont alignés).

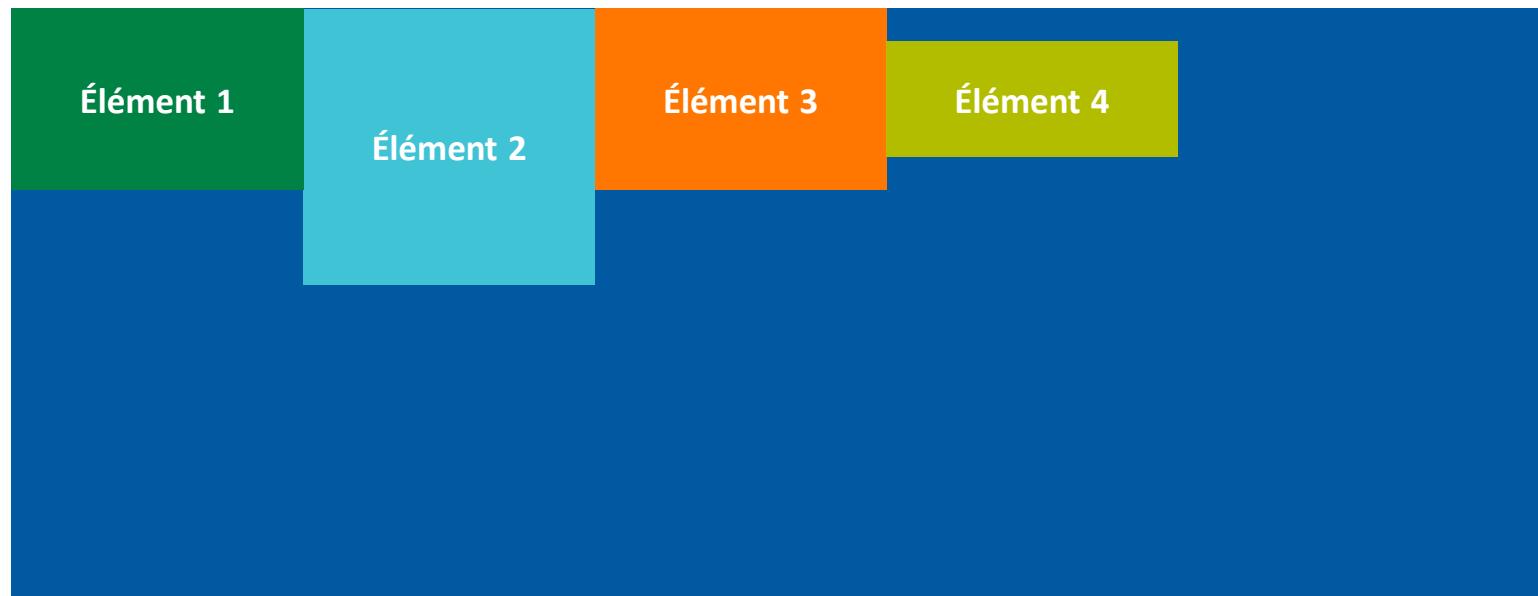


Figure 56: Exemple de la propriété align-items : baseline

Maitriser les passages à la ligne : la propriété "flex-wrap"

La propriété "flex-wrap" peut prendre les valeurs suivantes :

- **flex-wrap : nowrap** - c'est la valeur initiale. Si les éléments flexibles sont trop larges pour tenir dans leur conteneur, ils débordent du conteneur
- **flex-wrap : wrap** - si les éléments flexibles sont trop larges pour tenir dans le conteneur, ils passent automatiquement sur la ligne suivante (du haut vers le bas)
- **flex-wrap : wrap-reverse** - si les éléments flexibles sont trop larges pour le conteneur, ils passent automatiquement sur la ligne suivante (du bas vers le haut)

La propriété raccourcie **flex-flow** combine **flex-direction** et **flex-wrap**. Elle peut, par exemple, prendre les valeurs "row wrap" ou "column wrap".

02 - Utiliser les propriétés CSS

Les propriétés des conteneurs flex (FlexBox)



Les propriétés des éléments flex

- La propriété **order** permet de contrôler l'ordre dans lequel les éléments apparaissent dans le conteneur flex. Les valeurs possibles de cette propriété peuvent être soit des valeurs globales ("inherit", "initial", "unset" ou bien des valeurs entières positives ou négatives)
- La propriété **flex-grow** permet de définir l'agrandissement possible d'un élément proportionnellement aux autres. La valeur de cette propriété est un nombre. Par exemple, si "flex-grow : 2" pour un élément, il occupe (si possible) deux fois plus d'espace que les autres
- La propriété **flex-shrink** définit la possibilité pour un élément de rétrécir. Cette propriété accepte uniquement des valeurs positives
- La propriété **flex-basis** définit la taille par défaut d'un élément. Elle peut prendre des valeurs définies (3 em, 20 px, auto), des valeurs globales (inherit, initial, unset), des valeurs définies par mots-clés (fill, max-content, min-content, fit-content) ou la valeur "content" qui calcule la taille automatiquement en fonction du contenu de l'élément
- La propriété **flex** est une propriété raccourcie combinant les propriétés "flex-grow", "flex-shrink" et "flex-basis"
- La propriété **align-self** correspond à l'application de la propriété "align-items" pour un seul élément



CHAPITRE 3

Adapter une page web au dispositif d'affichage

Ce que vous allez apprendre dans ce chapitre :

- Exploiter les propriétés CSS permettant de définir les règles typographiques et de mise en page
- Utiliser les propriétés de positionnement afin de structurer les contenus



02 heures



CHAPITRE 3

Adapter une page web au dispositif d'affichage

1. Introduction au Responsive Design
2. Media Queries

Introduction au Responsive Design

- Les pages Web peuvent être consultées à l'aide de nombreux types d'appareils : ordinateurs de bureau, tablettes et téléphones. Ainsi, les pages Web ne doivent pas omettre d'informations pour s'adapter aux appareils plus petits, mais plutôt adapter leur contenu pour s'afficher correctement sur n'importe quel appareil utilisé
- La conception Web réactive (Responsive Design) rend la page Web compatible et adaptable à tous les appareils. Elle est basée uniquement sur HTML et CSS (elle n'est pas un programme ou un code JavaScript)



Figure 57: Principe du responsive design

- Le Responsive web design consiste à utiliser les langages CSS et HTML pour redimensionner, masquer, réduire, agrandir ou déplacer le contenu d'une page pour s'adapter à l'écran d'affichage



CHAPITRE 3

Adapter une page web au dispositif d'affichage

1. Introduction au Responsive Design
2. **Media Queries**

03 - Adapter une page web au dispositif d'affichage

Les media queries



Les media queries

La règle **@media**, introduite dans CSS2, permet de définir différentes règles de style pour différents types de médias.

Les **medias queries** dans CSS3 ont étendu l'idée des types de médias CSS2 : examiner la capacité du périphérique au lieu de chercher son type.

Les requêtes multimédias peuvent être utilisées pour vérifier de nombreuses propriétés, telles que :

- Largeur et hauteur de la fenêtre
- Largeur et hauteur de l'appareil
- Orientation (mode paysage ou portrait)
- Résolution

Les media queries

L'exemple suivant change la couleur d'arrière-plan selon la règle suivante :

- Si la largeur de la fenêtre est 480 px au minimum, la couleur est le bleu clair
- Si la largeur de la fenêtre est 200 px au minimum, la couleur est le lightgreen
- Si la largeur de la fenêtre est inférieure à 200 px, la couleur reste blanche

```
@media screen and (min-width : 200px)
{
    body
    {
        background-color :rgb(144, 174, 238);
    }
}

@media screen and (min-width : 480px)
{
    body
    {
        background-color : lightgreen;
    }
}
```

03 - Adapter une page web au dispositif d'affichage

Les media queries



Les media queries

Dans cet exemple, media queries est utilisée pour créer un menu de navigation réactif dont la conception varie selon les tailles d'écran.

```
.demo
{
    overflow : hidden;
    background-color :blue;
}
.demo a
{
    float : left;
    display : block;
    color : white;
    text-align : center;
    padding : 10px;
    text-decoration : none;
}
@media screen and (max-width : 600px)
{
    .demo a
    {
        float : none;
        width : 100%;
    }
}
```

```
<div class="demo">
    <a href="#">Accueil</a>
    <a href="#">A propos</a>
    <a href="#">Tutoriels</a>
    <a href="#">FAQ</a>
    <a href="#">Videos</a>
    <a href="#">Contact</a>
</div>
```



CHAPITRE 4

Créer des animations

Ce que vous allez apprendre dans ce chapitre :

- Manipuler des propriétés d'animation
- Appréhender les types d'animation
- Utiliser des keyFrames



06 heures



CHAPITRE 4

Créer des animations

1. Manipulation des propriétés d'animation
2. Types d'animation (transition, transformation...)
3. Utilisation des keyFrames

Propriétés d'animation

La propriété raccourcie (ou bien les propriétés détaillées correspondantes) est utilisée pour créer une animation CSS. Elle permet de configurer la durée, le minutage et d'autres détails de l'animation.

L'apparence visuelle de l'animation est définie en utilisant des règles CSS de mise en forme au sein de la règle @keyframes.

Les propriétés détaillées rattachées à la propriété raccourcie animation sont :

- **animation-delay** : Cette propriété définit le délai entre le chargement de l'élément et démarrage de l'animation.
- **animation-direction** : Cette propriété précise si l'animation doit alterner entre deux directions de progression (faire des allers-retours) ou recommencer au début à chaque cycle de répétition.
- **animation-duration** : Cette propriété définit la durée d'un cycle de l'animation.
- **animation-fill-mode** : Cette propriété indique les valeurs à appliquer aux propriétés avant et après l'exécution de l'animation.
- **animation-iteration-count** : Cette propriété détermine le nombre de répétition de l'animation. Pour répéter une animation infiniment, On utilise la valeur infinite.
- **animation-name** : Cette propriété déclare un nom à l'animation (utilisé comme référence à l'animation pour la règle @keyframes).
- **animation-play-state** : Cette propriété permet d'interrompre (« pause ») ou de reprendre l'exécution d'une animation.
- **animation-timing-function** : Cette propriété configure la fonction de minutage d'une animation.



CHAPITRE 4

Créer des animations

1. Manipulation des propriétés d'animation
2. **Types d'animation (transition, transformation...)**
3. Utilisation des keyFrames

04 - Créer des animations

Types d'animation



Animations CSS

CSS permet l'animation d'éléments HTML sans utiliser JavaScript

Une animation permet à un élément de passer progressivement d'un style à un autre en modifiant les propriétés CSS

Les types d'animations en CSS :

- Les transformations
- Les transitions

Les transformations 2D

Les transformations CSS 2D permettent de déplacer, faire pivoter, mettre à l'échelle et incliner des éléments.

1. La méthode **translate()** → Exemple : déplacer l'élément <div> de 50 pixels vers la droite et de 100 pixels vers le bas par rapport à sa position actuelle :

```
div {  
    width : 300px;  
    height : 100px;  
    background-color : cyan;  
    border : 1px solid black;  
}  
div :hover {  
    transform : translate(50px,100px);  
}
```

2. La méthode **rotate()** → Exemple : faire pivoter l'élément <div> dans le sens des aiguilles d'une montre de 20 degrés :

```
div {  
    width : 300px;  
    height : 100px;  
    background-color : cyan;  
    border : 1px solid black;  
}  
div :hover {  
    transform : rotate(20deg);  
}
```

Les transformations 2D

3. La méthode **scale()** → Exemple : augmenter la taille de l'élément <div> de deux fois sur sa largeur et trois fois sa hauteur :

```
div {  
    width : 300px;  
    height : 100px;  
    background-color : cyan;  
    border : 1px solid black;  
}  
div :hover {  
    transform : scale(2,3);  
}
```

4. La méthode **skew()** → Exemple : incliner l'élément <div> de 20 degrés le long de l'axe X et de 10 degrés le long de l'axe Y :

```
div {  
    width : 300px;  
    height : 100px;  
    background-color : cyan;  
    border : 1px solid black;  
}  
div :hover {  
    transform : skew(20deg,10deg);  
}
```

Les transformations 2D

5. La méthode **rotateX()** → Permet de faire une rotation par rapport à l'axe X (on peut utiliser aussi **rotateY** et **rotateZ**).

```
div
{
    width : 300px;
    height : 100px;
    background-color : cyan;
    border : 1px solid black;
}

div :hover
{
    transform : rotateX(60deg);
}
```

Transitions CSS

Les transitions CSS permettent de modifier les valeurs des propriétés sur une durée donnée.

Exemple : un élément <div> rouge de 100 px * 100 px subit un effet de transition pour la propriété width, d'une durée de 2 secondes :

```
div
{
    width : 100px;
    height : 100px;
    background : red;
    transition : width 2s;
}
```

Lorsqu'un utilisateur passe la souris sur l'élément <div>, une nouvelle valeur pour la propriété **width** s'applique.

L'effet de transition démarre lorsque la propriété CSS spécifiée (largeur) change de valeur.

```
div :hover
{
    width : 300px;
}
```

```
div
{
    transition :width 2s, height 4s;
}
```



CHAPITRE 4

Créer des animations

1. Manipulation des propriétés d'animation
2. Types d'animation (transition, transformation...)
- 3. Utilisation des keyFrames**

La règle @keyframes

La spécification d'un style CSS à l'intérieur de la règle @keyframes permet à une animation de passer progressivement du style actuel au nouveau style.

Exemple 1 :

```
@keyframes exemple
{
    from {background-color : red;}
    to {background-color : yellow;}
}

/* L'élément sur lequel s'appliquer l'animation */
div
{
    width : 100px;
    height : 100px;
    background-color : red;
    animation-name : exemple;
    animation-duration : 4s;
}
```

définit le nom de l'animation

définit la durée de l'animation

04 - Créer des animations

Utilisation des keyFrames

La règle @keyframes

Exemple 2 :

```
@keyframes exemple
{
    0% {background-color : red;}
    25% {background-color : yellow;}
    50% {background-color : blue;}
    100% {background-color : green;}
}

/* L'élément sur lequel s'appliquer l'animation */
div
{
    width : 100px;
    height : 100px;
    background-color : red;
    animation-name : exemple;
    animation-duration : 4s;
}
```

La règle @keyframes

Exemple 3 :

```
@keyframes example
{
    0%   {background-color :red; left :0px; top :0px;}
    25%  {background-color :yellow; left :200px; top :0px;}
    50%  {background-color :blue; left :200px; top :200px;}
    75%  {background-color :green; left :0px; top :200px;}
    100% {background-color :red; left :0px; top :0px;}
}
```

```
div
{
    width : 100px;
    height : 100px;
    position : relative;
    background-color : red;
    animation-name : example;
    animation-duration : 4s;
}
```



CHAPITRE 5

Adapter des templates HTML/CSS avec un site Web

Ce que vous allez apprendre dans ce chapitre :

- Analyser la structure d'un template
- Adapter le contenu d'un site selon la charte



02 heures



CHAPITRE 5

Adapter des templates HTML/CSS avec un site Web

1. Analyse de la structure d'un template
2. Adaptation du contenu d'un site selon la charte

Notion de emplate

- Un **template**, également appelé **modèle**, **layout** ou **Gabarit** est souvent utilisé de manière répétitive pour créer des documents qui partagent une même structure, abstraction faite de leurs contenus
- C'est une structure représentant la mise en page des documents web passant par le choix des couleurs jusqu'à l'établissement de la structure des différents éléments
- L'utilisation des templates prédefinis est largement adoptée par les créateurs des sites web vu que ça leur permet de centrer leurs efforts sur le contenu du site plutôt que le design

Que contient un template ?

Le template ou gabarit contient notamment :

- **zonage du site** : le pré-positionnement des zones constituant la page à savoir les entêtes, l'identité visuelle, les colonnes et le footer/
- La localisation des **menus de navigation**
- **L'identité visuelle du site** : Les caractères, les formes et les couleurs du contenu textuel
- Le **contenu visuel** tel que les images, les icônes, les fonds...

Pourquoi utiliser un template ?

L'utilisation d'un template permet de créer une cohérence entre les différentes pages d'un site web. En effet, chaque page du site contient des contenus qui lui sont appropriés mais elle partage certains éléments avec l'ensemble des pages. Cela permet aux utilisateurs du site de garder des repères tout au long de la navigation

Comment choisir un template ?

Avant de faire le choix d'un template, il faut définir les fonctionnalités et les contenus du site pour éviter les risques pendant la phase de réalisation. Par exemple se rendre compte que le mode de fonctionnement du template ne correspond pas aux attentes du site.

En choisissant un template, le designer est engagé à rester dans un cadre prédéfini (structure des pages, position des menus de navigation...). Donc ce choix n'est pas adapté au types de projets web qui imposent un design particulier, ou des évolutions fréquentes.



CHAPITRE 5

Adapter des templates HTML/CSS avec un site Web

1. Analyse de la structure d'un template
2. **Adaptation du contenu d'un site selon la charte**

Notion de charte graphique d'un site web

La charte graphique d'un site web est l'ensemble des règles fondamentales gérant l'utilisation des signes graphiques qui constituent l'identité graphique du site.

La charte graphique d'un site se compose de plusieurs éléments essentiels : le logo, les couleurs, la typographie, les icônes, les images/photos/illustrations, ainsi que les menus de navigation et le positionnement des blocs de contenus.

Les éléments de la charte graphique d'un site web

Le logo :

- Le logo du site est l'élément central de l'identité visuelle. La charte web doit définir avec précision ses proportions, ses dimensions, et son emplacement exacte sur le site.

La typographie :

- La police d'écriture du site web est aussi importante dans la communication visuelle. La charte web la définit en s'assurant de sa compatibilité avec les autres supports de communication.

Les couleurs :

- Le choix des couleurs est primordial pour la réussite du site. Les couleurs transmettent des messages différents aux visiteurs du site, en fonction du métier de l'entreprise.
- Afin de maximiser l'impact des couleurs (faciliter la mémorisation par exemple), leur nombre doit être limité à 4 voire 5 maximum.

Les icônes :

- Le choix des icônes est aussi important que les autres éléments de la charte graphique. Les icônes permettent créer l'image de marque du site , et de l'identifier facilement (comme Facebook, Twitter, LinkedIn, icône du "panier", icône de "rechercher« , ...)

Les images :

- La charte graphique du site web doit prévoir une utilisation pertinente des images en fonction de la mission du site.



PARTIE 4

Maitriser Bootstrap

Dans ce module, vous allez :

- Maitriser des différentes composantes de Bootstrap
- Réaliser une production adaptée d'une page web avec Bootstrap
- Maitriser des classes CSS de Bootstrap
- Réaliser une production aisée d'une page web responsive



 25 heures

CHAPITRE 1

Intégrer Bootstrap



Ce que vous allez apprendre dans ce chapitre :

- Découvrir Bootstrap
- Utiliser le système de grille



5 heures



CHAPITRE 1

Intégrer Bootstrap

1. **Introduction au Bootstrap (versions, avantages...)**
2. Système de grille

01 - Intégrer Bootstrap

Introduction au Bootstrap



C'est quoi Bootstrap ?

- **Bootstrap** est un framework gratuit conçu pour permettre un développement plus rapide et plus facile des pages web réactives

Bootstrap comprend des modèles de conception basés sur HTML et CSS pour la typographie, les formulaires, les boutons, les tableaux, la navigation, les images, ...

- **Bootstrap 5** (sortie en 2021) est la dernière version de Bootstrap (première sortie en 2013)

Bootstrap 5 propose de nouveaux composants, une feuille de style plus rapide et plus de réactivité. Il prend également en charge les dernières versions stables des principaux navigateurs et plates-formes.

01 - Intégrer Bootstrap

Introduction au Bootstrap



Avantages Bootstrap

- **Facile à utiliser** : par les développeurs ayant des connaissances de base en HTML et CSS.
- **Fonctionnalités réactives** : le CSS réactif de Bootstrap s'adapte aux différents médias (téléphones, tablettes et ordinateurs de bureau).
- **Approche mobile first** : les styles mobile first font partie du framework de base.
- **Compatibilité du navigateur** : Bootstrap 5 est compatible avec tous les navigateurs modernes (Chrome, Firefox, Edge, Safari et Opera).

Obtenir Bootstrap

Pour utiliser Bootstrap 5 sur le site Web, on peut :

- Inclure Bootstrap 5 à partir d'un CDN (Content Delivery Network).
- Téléchargez Bootstrap 5 sur getbootstrap.com et l'associer aux pages web comme une feuille de style CSS.

Bootstrap 5 à partir de CDN

Ajouter les liens suivants dans le <head> du document HTML. jsDelivr fournit un support CDN pour les CSS et JavaScript de Bootstrap :

```
<!-- Dernier CSS compilé et minifié -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css" rel="stylesheet">

<!-- Dernier JavaScript compilé -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/js/bootstrap.bundle.min.js"></script>
```



CHAPITRE 1

Intégrer Bootstrap

1. Introduction au Bootstrap (versions, avantages...)
2. **Système de grille**

Les conteneurs

Les **conteneurs (containers)** sont l'élément de mise en page le plus basique dans Bootstrap. Ils sont utilisés **par le système de grille (grid)** par défaut.

Le rôle des conteneurs est de contenir, remplir et (parfois) centrer le contenu trouvé à leur intérieur.

Bien que les conteneurs puissent être imbriqués, la plupart des mises en page ne nécessitent pas de conteneur imbriqué [8]

Il existe deux classes de conteneurs disponibles :

La classe **.container** fournit un conteneur de largeur fixe réactif.

La classe **.container-fluid** fournit un conteneur pleine largeur, couvrant toute la largeur de la fenêtre.

Exemple :

```
<div class="container">
    <h1>Bootstrap est utilisé dans cette page.</h1>
    <p>Ceci est un texte</p>
</div>
```

```
<div class="container-fluid">
    <h1>Bootstrap est utilisé dans cette page.</h1>
    <p>Ceci est un texte</p>
</div>
```

01 - Intégrer Bootstrap

Système de grille



Le padding du conteneur

Par défaut, les conteneurs ont un remplissage à gauche et à droite, sans remplissage en haut ou en bas. Par conséquent, les utilitaires d'espacement tels que des marges et des marges supplémentaires sont utilisés pour leur mise en forme.

Par exemple, .pt-5 signifie "ajouter un grand padding supérieur" :

Exemple :

```
<div class="container pt-5"></div>
```

Bordure et couleur du conteneur

D'autres utilitaires, tels que les bordures et les couleurs, sont également utilisés avec les conteneurs.

Exemple :

```
<div class="container p-5 my-5 border"></div>
<div class="container p-5 my-5 bg-dark text-white"></div>
<div class="container p-5 my-5 bg-primary text-white"></div>
```

Les conteneurs réactifs

Il s'agit des classes `.container-sm | md | lg | xl`

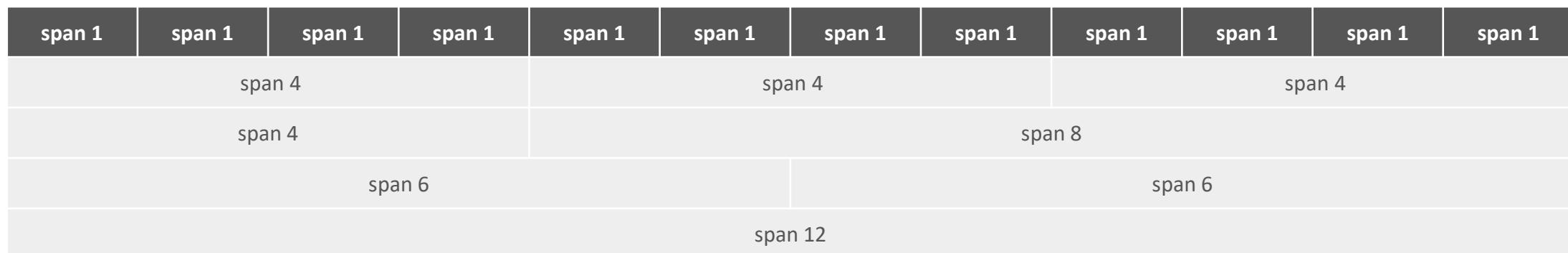
Le **max-width** du conteneur changera selon les différentes tailles d'écran/fenêtre :

Classe	Super petit <code>≥576 px</code>	Petit <code>≥576 px</code>	Moyen <code>768 px</code>	Grand <code>≥992 px</code>	Extra large <code>≥1200 px</code>	XXL <code>≥1400 px</code>
<code>.container-sm</code>	100 %	540 px	720 px	960 px	1140 px	1320 px
<code>.container-md</code>	100 %	100 %	720 px	960 px	1140 px	1320 px
<code>.container-lg</code>	100 %	100 %	100 %	960 px	1140 px	1320 px
<code>.container-xl</code>	100 %	100 %	100 %	100 %	1140 px	1320 px
<code>.container-xxl</code>	100 %	100 %	100 %	100 %	100 %	1320 px

Système de grille Bootstrap 5

Le système de grille de base de Bootstrap est conçu avec **FlexBox** et propose jusqu'à 12 colonnes sur la page.

Il est possible de regrouper les colonnes pour créer des colonnes plus larges :



Le système de grille est réactif et les colonnes se réorganiseront automatiquement en fonction de la taille de l'écran.

01 - Intégrer Bootstrap

Système de grille



Les classes du Système de grille Bootstrap 5

Le système de grille Bootstrap 5 comporte six classes :

- **.col-** (très petits appareils - largeur d'écran inférieure à 576 px)
- **.col-sm-** (petits appareils - largeur d'écran égale ou supérieure à 576 px)
- **.col-md-** (appareils moyens - largeur d'écran égale ou supérieure à 768 px)
- **.col-lg-** (grands appareils - largeur d'écran égale ou supérieure à 992 px)
- **.col-xl-** (appareils xlarge - largeur d'écran égale ou supérieure à 1200 px)
- **.col-xxl-** (appareils xxlarge - largeur d'écran égale ou supérieure à 1400 px)

Exemple 1

```
<!-- Contrôlez la largeur des colonnes et comment elles doivent apparaître sur différents appareils -->
```

```
<div class="row">
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
</div>
<div class="row">
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
</div>
```

La première étoile (*) représente la réactivité : sm, md, lg, xl ou xxl, tandis que la deuxième étoile représente un nombre qui doit totaliser 12 pour chaque ligne.

```
<!-- Ou laissez Bootstrap gérer automatiquement la mise en page -->
```

```
<div class="row">
  <div class="col"></div>
  <div class="col"></div>
  <div class="col"></div>
</div>
```

Au lieu d'ajouter un nombre à chaque col, laissez bootstrap gérer la mise en page pour créer des colonnes de largeur égale.

Exemple 2 : Quatre colonnes de même largeur

L'exemple 2 montre comment créer quatre colonnes de largeur égale en commençant par les tablettes et en évoluant vers de grands bureaux. Sur les téléphones portables ou les écrans de moins de 576 px de large, les colonnes s'empilent automatiquement les unes sur les autres :

```
<div class="row">
    <div class="col-sm-3">.col-sm-3</div>
    <div class="col-sm-3">.col-sm-3</div>
    <div class="col-sm-3">.col-sm-3</div>
    <div class="col-sm-3">.col-sm-3</div>
</div>
```

Exemple 3 : Deux colonnes réactives inégales

L'exemple 3 montre comment obtenir deux colonnes de largeurs différentes en commençant par les tablettes et en évoluant vers les supports plus grands:

```
<div class="row">
    <div class="col-sm-4">.col-sm-4</div>
    <div class="col-sm-8">.col-sm-8</div>
</div>
```

CHAPITRE 2

Maitriser les classes CSS de base



Ce que vous allez apprendre dans ce chapitre :

- Gérer la typographie et les liens
- Mettre en place des tableaux
- Concevoir de formulaires et contrôles supportés
- Créer des boutons



10 heures



CHAPITRE 2

Maitriser les classes CSS de base

1. **Typographie et liens**
2. Tableaux
3. Éléments de formulaires et contrôles supportés
4. Boutons

Le texte avec Bootstrap

Bootstrap 5 utilise par défaut une valeur `font-size` de `1 rem` (16 px par défaut) et sa valeur `line-height` est de 1,5.

De plus, tous les éléments de type `<p>` ont un `margin-top : 0` et un `margin-bottom : 1 rem` (16 px par défaut).

On peut également utiliser des classes `.h1` et `.h6` sur d'autres éléments les forcer à se comporter comme des titres :

```
<p class="h1">h1 Bootstrap heading</p>
<p class="h2">h2 Bootstrap heading</p>
<p class="h3">h3 Bootstrap heading</p>
<p class="h4">h4 Bootstrap heading</p>
<p class="h5">h5 Bootstrap heading</p>
<p class="h6">h6 Bootstrap heading</p>
```

La balise <mark>

L'élément .mark de Bootstrap 5 a pour effet une couleur de fond jaune.

```
<p>Utiliser l'élément mark (ou la classe .mark) pour <mark>mettre en évidence</mark> text.</p>
```

La balise <abbr>

L'élément Bootstrap 5 <abbr> a pour effet une bordure en pointillé en bas et un curseur avec un point d'interrogation au survol :

```
<div class="container mt-3">
  <h1>Abbreviations</h1>
  <p>L'élément abbr est utilisé pour baliser une abréviation ou un acronyme :</p>
  <p>Le <abbr title="Java Script">JS</abbr> est un langage de programmation</p>
</div>
```

Les couleurs du texte

Les classes pour les couleurs du texte sont :

- .text-muted,
- .text-primary
- .text-success
- .text-info
- .text-warning
- .text-danger
- .text-secondary
- .text-white
- .text-dark
- .text-body (couleur du corps par défaut, souvent noir)
- .text-light

```
<div class="conteneur mt-3">
  <h2>Couleurs contextuelles</h2>
  <p class="text-muted">Ce texte est coupé.</p>
  <p class="text-primary">Ce texte est important.</p>
  <p class="text-success">Ce texte indique le succès.</p>
  <p class="text-info">Ce texte représente une information.</p>
  <p class="text-warning">Ce texte représente un avertissement.</p>
  <p class="text-danger">Ce texte représente un danger.</p>
  <p class="text-secondaire">Texte secondaire.</p>
  <p class="text-dark">Ce texte est gris foncé.</p>
  <p class="text-body">Couleur du corps par défaut (souvent noir).</p>
  <p class="text-light">Ce texte est en gris clair (sur fond blanc).</p>
  <p class="text-white">Ce texte est blanc (sur fond blanc).</p>
</div>
```

On peut aussi ajouter 50 % d'opacité pour le texte noir ou blanc avec les classes .text-black-50 ou .text-white-50.

Les couleurs d'arrière plan

Les classes à utiliser pour les couleurs d'arrière-plan sont :

- .bg-primary
- .bg-success
- .bg-info
- .bg-warning
- .bg-danger
- .bg-secondary
- .bg-darket
- .bg-light

```
<div class="conteneur mt-3">
    <h2>Arrière-plans contextuels</h2>
    <p>Ajouter une classe .text-* pour une couleur de texte différente :</p>
    <p class="bg-primary text-white">Ce texte est important.</p>
    <p class="bg-success text-white">Ce texte indique le succès.</p>
    <p class="bg-info text-white">Ce texte représente une information.</p>
    <p class="bg-warning text-white">Ce texte représente un avertissement.</p>
    <p class="bg-danger text-white">Ce texte représente un danger.</p>
    <p class="bg- Secondary text-white">Couleur d'arrière-plan secondaire.</p>
    <p class="bg-dark text-white">Couleur de fond gris foncé.</p>
    <p class="bg-light text-dark">Couleur de fond gris clair.</p>
</div>
```

Remarque : les couleurs d'arrière-plan ne définissent pas la couleur du texte, on peut dans ce cas utiliser la classe .text-*.



CHAPITRE 2

Maitriser les classes CSS de base

1. Typographie et liens
2. **Tableaux**
3. Éléments de formulaires et contrôles supportés
4. Boutons

Les tableaux

Une table Bootstrap 5 de base a un padding léger et des séparateurs horizontaux. La classe .table ajoute un style de base à une table :

```
<table class="table">
  <thead>
    <tr><th>Nom</th><th>Prénom</th><th>Age</th></tr>
  </thead>
  <tbody>
    <tr><td>Mohamed</td><td>KINANI</td><td>25</td></tr>
    <tr><td>Laila</td><td>SAFIR</td><td>23</td></tr>
    <tr><td>Sami</td><td>MANSOURI</td><td>24</td></tr>
  </tbody>
</table>
```

Nom	Prénom	Age
Mohamed	KINANI	25
Laila	SAFIR	23
Sami	MANSOURI	24

Figure 58 : Exemple de table avec Bootstrap

Les lignes rayées

```
<table class="table table-striped">
  <thead>
    <tr><th>Nom</th><th>Prénom</th><th>Age</th></tr>
  </thead>
  <tbody>
    <tr><td>Mohamed</td><td>KINANI</td><td>25</td></tr>
    <tr><td>Laila</td><td>SAFIR</td><td>23</td></tr>
    <tr><td>Sami</td><td>MANSOURI</td><td>24</td></tr>
  </tbody>
</table>
```

Nom	Prénom	Age
Mohamed	KINANI	25
Laila	SAFIR	23
Sami	MANSOURI	24

Figure 59 : Exemple de lignes rayées avec Bootstrap

02 - Maîtriser les classes CSS de base

Les tableaux



Les bordures

```
<table class="table table-bordered">
  <thead>
    <tr><th>Nom</th><th>Prénom</th><th>Age</th></tr>
  </thead>
  <tbody>
    <tr><td>Mohamed</td><td>KINANI</td><td>25</td></tr>
    <tr><td>Laila</td><td>SAFIR</td><td>23</td></tr>
    <tr><td>Sami</td><td>MANSOURI</td><td>24</td></tr>
  </tbody>
</table>
```

Nom	Prénom	Age
Mohamed	KINANI	25
Laila	SAFIR	23
Sami	MANSOURI	24

Figure 60 : Exemple de table avec bordure de lignes

Survoler les lignes

La classe `.table-hover` ajoute un effet de survol (couleur de fond gris) sur les lignes du tableau :

```
<table class="table table-hover">
  <thead>
    <tr><th>Nom</th><th>Prénom</th><th>Age</th></tr>
  </thead>
  <tbody>
    <tr><td>Mohamed</td><td>KINANI</td><td>25</td></tr>
    <tr><td>Laila</td><td>SAFIR</td><td>23</td></tr>
    <tr><td>Sami</td><td>MANSOURI</td><td>24</td></tr>
  </tbody>
</table>
```

Nom	Prénom	Age
Mohamed	KINANI	25
Laila	SAFIR	23
Sami	MANSOURI	24

Figure 61 : Exemple de tableau avec survol des lignes

Tableau avec mode dark (foncé)

La classe `.table-dark` ajoute un fond noir au tableau :

```
<table class="table table-dark">
  <thead>
    <tr><th>Nom</th><th>Prénom</th><th>Age</th></tr>
  </thead>
  <tbody>
    <tr><td>Mohamed</td><td>KINANI</td><td>25</td></tr>
    <tr><td>Laila</td><td>SAFIR</td><td>23</td></tr>
    <tr><td>Sami</td><td>MANSOURI</td><td>24</td></tr>
  </tbody>
</table>
```

Nom	Prénom	Age
Mohamed	KINANI	25
Laila	SAFIR	23
Sami	MANSOURI	24

Figure 62 : Exemple de table mode foncé

Tableau à rayures foncées

Les classes `.table-dark` et `.table-striped` peuvent être combinées pour créer une table sombre et rayée :

```
<table class="table table-dark table-striped">
  <thead>
    <tr><th>Nom</th><th>Prénom</th><th>Age</th></tr>
  </thead>
  <tbody>
    <tr><td>Mohamed</td><td>KINANI</td><td>25</td></tr>
    <tr><td>Laila</td><td>SAFIR</td><td>23</td></tr>
    <tr><td>Sami</td><td>MANSOURI</td><td>24</td></tr>
  </tbody>
</table>
```

Nom	Prénom	Age
Mohamed	KINANI	25
Laila	SAFIR	23
Sami	MANSOURI	24

Figure 63 : Exemple de tableau à rayures foncées

Tableau sans bordure

```
<table class="table table-borderless">
  <thead>
    <tr><th>Nom</th><th>Prénom</th><th>Age</th></tr>
  </thead>
  <tbody>
    <tr><td>Mohamed</td><td>KINANI</td><td>25</td></tr>
    <tr><td>Laila</td><td>SAFIR</td><td>23</td></tr>
    <tr><td>Sami</td><td>MANSOURI</td><td>24</td></tr>
  </tbody>
</table>
```

Nom	Prénom	Age
Mohamed	KINANI	25
Laila	SAFIR	23
Sami	MANSOURI	24

Figure 64 : Exemple de tableau sans bordure

Les classes contextuelles

Les classes contextuelles peuvent être utilisées pour colorer l'ensemble du tableau (`<table>`), les lignes du tableau (`<tr>`) ou les cellules du tableau (`<td>`).

Classe	La description
<code>.table-primary</code>	Bleu : indique une action importante
<code>.table-success</code>	Vert : indique une action réussie ou positive
<code>.table-danger</code>	Rouge : indique une action dangereuse ou potentiellement négative
<code>.table-info</code>	Bleu clair : indique une action ou un changement informatif neutre
<code>.table-warning</code>	Orange : indique un avertissement qui peut nécessiter votre attention
<code>.table-active</code>	Gris : applique la couleur de survol à la ligne ou à la cellule du tableau
<code>.table-secondary</code>	Gris : Indique une action un peu moins importante
<code>.table-light</code>	Fond de table ou de rangée de table gris clair
<code>.table-dark</code>	Fond de table ou de rangée de table gris foncé

Les tableaux réactifs

La classe .table-responsive ajoute une barre de défilement au tableau en cas de besoin (si le tableau est grand horizontalement) :

```
<div class="table-responsive">
  <table class="table">
    ...
  </table>
</div>
```

On peut également décider quand le tableau doit avoir une barre de défilement en fonction de la largeur de l'écran :

Classe	Largeur de l'écran
.table-responsive-sm	< 576 px
.table-responsive-md	< 768 px
.table-responsive-lg	< 992 px
.table-responsive-xl	< 1200 px
.table-responsive-xxl	< 1400 px

02 - Maîtriser les classes CSS de base

Les images

Coups arrondis

La classe **.rounded** ajoute des coups arrondis à une image :

```

```



Cercle

La classe **.rounded-circle** façonne l'image en cercle :

```

```



La vignette

La classe **.img-thumbnail** façonne l'image en une vignette :

```

```



Figure 65 :Coups arrondis

02 - Maîtriser les classes CSS de base

Les images

Alignment des images

Les classes `.float-start` et `.float-end` permettent de faire flotter une image à gauche ou à droite.

```

```



Figure 66 : Alignement à gauche

```

```



Figure 67 : Alignement à droite

Image centrée

Centrer une image en ajoutant les classes utilitaires `.mx-auto`(margin :auto) et `.d-block`(display :block) à l'image :

```

```

Les images réactives

- Les images réactives s'ajustent automatiquement pour s'adapter à la taille de l'écran.
- On peut créer des images responsives en ajoutant une classe `.img-fluid` à la balise ``. L'image s'adaptera alors à l'élément parent.
- La classe `.img-fluid` donne : `max-width : 100 %` et `height : auto` à l'image :

```

```



CHAPITRE 2

Maitriser les classes CSS de base

1. Typographie et liens
2. Tableaux
- 3. Éléments de formulaires et contrôles supportés**
4. Boutons

La forme empilée

La classe `.form-control` permet aux éléments textuels `<input>` et `<textarea>` d'obtenir un style empilé dans un formulaire :

```
<form action="/action_page.php">
    <div class="mb-3 mt-3">
        <label for="email" class="form-label">E-mail :</label>
        <input type="email" class="form-control" id="email" placeholder="Entrez l'email" name="email">
    </div>
    <div class="mb-3">
        <label for="pwd" class="form-label">Mot de passe :</label>
        <input type="password" class="form-control" id="pwd" placeholder="Entez le mot de passe" name="pswd">
    </div>
    <div class="form-check mb-3">
        <label class="form-check-label">
            <input class="form-check-input" type="checkbox" name="remember"> Se souvenir de moi</label>
        </div>
        <button type="submit" class="btn btn-primary">Soumettre</button>
    </form>
```



The screenshot shows a web form with three stacked input fields and a submit button. The first field is labeled "E-mail :" with a placeholder "Entrez l'email". The second field is labeled "Mot de passe :" with a placeholder "Entez le mot de passe". The third field is a checkbox labeled "Se souvenir de moi". A blue "Soumettre" button is at the bottom.

Figure 68 : Forme empilée

02 - Maîtriser les classes CSS de base

Éléments de formulaires et contrôles supportés



Formulaire inline

Les classes `.row` et `.col` permettent aux éléments du formulaire d'apparaître les uns à côté des autres :

```
<form>
  <div class="row">
    <div class="col">
      <input type="text" class="form-control" placeholder="Entrez l'email" name="email">
    </div>
    <div class="col">
      <input type="password" class="form-control" placeholder="Entrer le mot de passe" name="pswd">
    </div>
  </div>
</form>
```

Entrez l'e-mail

Entrer le mot de passe

Figure 69 : Forme inline

02 - Maîtriser les classes CSS de base

Éléments de formulaires et contrôles supportés

Taille du contrôle de formulaire

On peut modifier la taille des entrées en utilisant `.form-control` avec `.form-control-lg` ou `.form-control-sm` :

```
<input type="text" class="form-control form-control-lg" placeholder="Grande entrée">
<input type="text" class="form-control" placeholder="Entrée normale">
<input type="text" class="form-control form-control-sm" placeholder="Petite entrée">
```



Figure 70 : Taille des contrôles

Contrôle désactivé et en lecture seule

Les attributs **disabled** et/ou **readonly** permettent de désactiver le champ de saisie :

```
<input type="text" class="form-control" placeholder="Entrée normale">
<input type="text" class="form-control" placeholder="Entrée désactivée" disabled>
<input type="text" class="form-control" placeholder="Entrée en lecture seule" readonly>
```

Entrée normale

Entrée désactivée

Entrée en lecture seule

Figure 71 : contrôles désactivés / en lecture seule

02 - Maîtriser les classes CSS de base

Éléments de formulaires et contrôles supportés

Les listes

Pour associer un style Bootstrap 5 à une liste de sélection, on utilise la classe `.form-select` à l'élément `<select>` :

```
<select class="form-select">
  <option>1</option>
  <option>2</option>
  <option>3</option>
  <option>4</option>
</select>
```

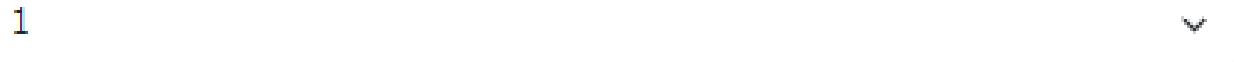


Figure 72 : Liste `<select>` avec choix unique

```
<select multiple class="form-select">
  <option>1</option>
  <option>2</option>
  <option>3</option>
  <option>4</option>
</select>
```

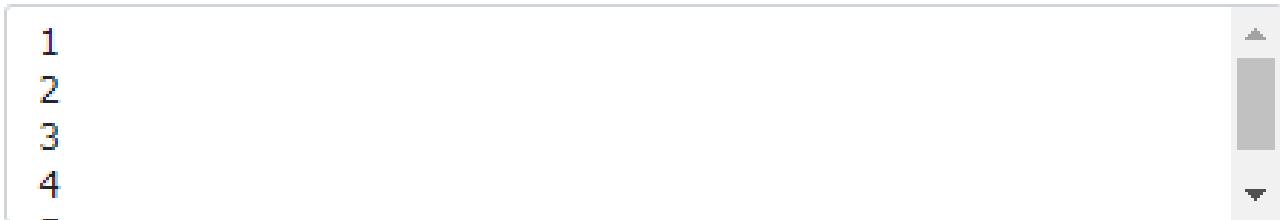
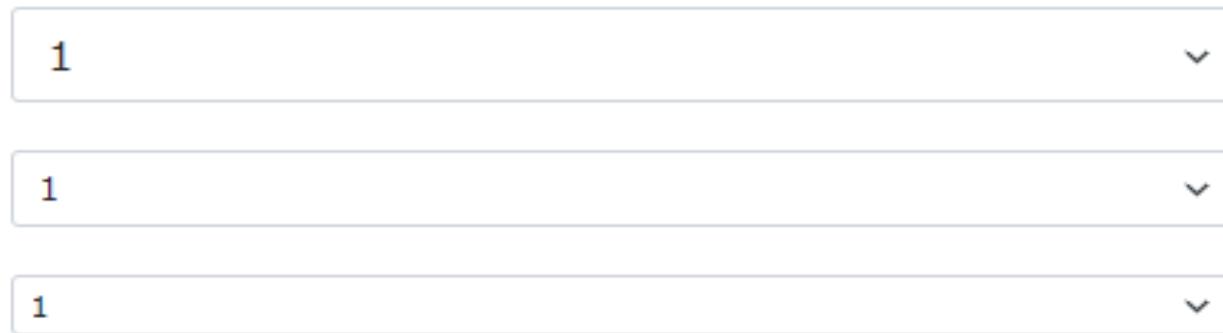


Figure 73 : Liste `<select>` avec choix multiple

La taille de la liste <select>

La classe `.form-select-lg` ou `.form-select-sm` permet de modifier la taille de la liste :

```
<select class="form-select form-select-lg">
    <option>1</option>
</select>
<select class="form-select">
    <option>1</option>
</select>
<select class="form-select form-select-sm">
    <option>1</option>
</select>
```



The figure displays three vertically stacked dropdown menus, each containing a single option '1'. The top menu is large, the middle one is standard size, and the bottom one is small. Each menu has a downward-pointing arrow icon on its right side.

Figure 74 : Taille de la liste <select>

Les listes de données <datalist>

Bootstrap permet d'associer un style aux listes de données, qui sont une liste d'options prédéfinies pour un élément <input> :

```
<label for="browser" class="form-label">Choisissez votre navigateur dans la liste:</label>
<input class="form-control" list="browsers" name="browser" id="browser">
<datalist id="browsers">
  <option value="Edge">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

Choisissez votre navigateur dans la liste :

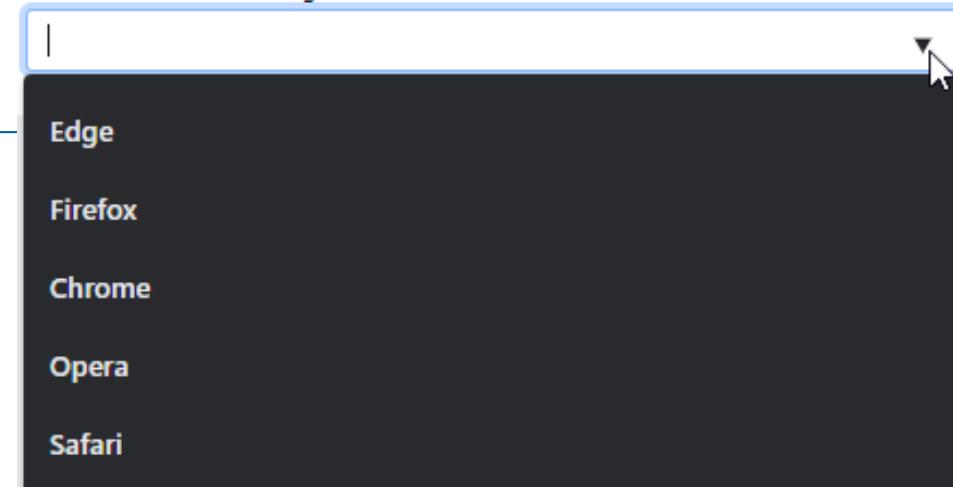


Figure 75 : Style de dataList

Cases à cocher

La classe **.form-switch** avec le conteneur **.form-check** ajoute un style d'interrupteur à bascule pour une case à cocher.

```
<form action="/action.php">
  <div class="form-check form-switch">
    <input class="form-check-input" type="checkbox" id="mySwitch" name="darkmode" value="yes" checked>
    <label class="form-check-label" for="mySwitch">Mode sombre</label>
  </div>
</form>
```

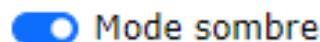


Figure 76 : Style des cases à cocher

02 - Maîtriser les classes CSS de base

Éléments de formulaires et contrôles supportés

Validation du formulaire

```
<form action="/action.php" class="was-validated">
    <div class="mb-3 mt-3">
        <label for="uname" class="form-label">Nom d'utilisateur :</label>
        <input type="text" class="form-control" id="uname" placeholder="Saisir le nom d'utilisateur"
name="uname" required>
            <div class="valid-feedback">Valide.</div>
            <div class="invalid-feedback">Merci de remplir ce champ.</div>
    </div>
    <div class="mb-3">
        <label for="pwd" class="form-label">Mot de passe :</label>
        <input type="password" class="form-control" id="pwd" placeholder="Saisir le mot de passe"
name="pswd" required>
            <div class="valid-feedback">Valide.</div>
            <div class="invalid-feedback">Merci de remplir ce champ.</div>
    </div>
    <button type="submit" class="btn btn-primary">Envoyer</button>
</form>
```

Nom d'utilisateur :

!

Merci de remplir ce champ.

Mot de passe :

!

Merci de remplir ce champ.

Envoyer

Figure 77 : Validation des formulaires



CHAPITRE 2

Maitriser les classes CSS de base

1. Typographie et liens
2. Tableaux
3. Éléments de formulaires et contrôles supportés
- 4. Boutons**

Styles de boutons

Bootstrap 5 propose différents styles de boutons. Les classes de boutons peuvent être utilisées sur les éléments `<a>`, `<button>` ou `<input>`.

```
<button type="button" class="btn">De base</button>
<button type="button" class="btn btn-primary">Primaire</button>
<button type="button" class="btn btn-secondary">Secondaire</button>
<button type="button" class="btn btn-success">Succès</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-warning">Avertissement</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-dark">Sombre</button>
<button type="button" class="btn btn-light">Léger</button>
<button type="button" class="btn btn-link">Relier</button>
```

[De base](#)[Primaire](#)[Secondaire](#)[Succès](#)[Info](#)[Avertissement](#)[Danger](#)[Sombre](#)[Léger](#)[Relier](#)

Figure 78 : Styles de boutons

Contour du bouton

Bootstrap 5 fournit huit types de contours des boutons :

```
<button type="button" class="btn btn-outline-primary">Primaire</button>
<button type="button" class="btn btn-outline-secondary">Secondaire</button>
<button type="button" class="btn btn-outline-success">Succès</button>
<button type="button" class="btn btn-outline-info">Info</button>
<button type="button" class="btn btn-outline-warning">Avertissement</button>
<button type="button" class="btn btn-outline-danger">Danger</button>
<button type="button" class="btn btn-outline-dark">Sombre</button>
<button type="button" class="btn btn-outline-light text-dark">Léger</button>
```

Primaire

Secondaire

Succès

Info

Avertissement

Danger

Sombre

Léger

Figure 79 : Contour des boutons

CHAPITRE 3

Utiliser les composants Bootstrap



Ce que vous allez apprendre dans ce chapitre :

- Créer des barres de navigation et des menus déroulants
- Mettre en place la pagination
- Réaliser des barres de progression et des cartes



10 heures



CHAPITRE 3

Utiliser les composants Bootstrap

- 1. Barres de navigation**
2. Menus déroulants
3. Pagination, badges et alertes
4. Barres de progression
5. Cartes (cards)

03- Utiliser les composants Bootstrap

Barres de navigation



Barre de navigation standard

La classe `.navbar`, suivie de la classe de `.navbar-expand-xxl|xl|lg|md|sm` permet de créer une barre de navigation standard

Les éléments de barre de navigation seront empilés verticalement sur des écrans **xx large, extra large, large, moyen ou petit**.

Pour ajouter des liens dans la barre de navigation, on utilise un élément `` (ou un `<div>`) avec la `class="navbar-nav"`.

Les éléments `` avec une classe `.nav-item` suivis d'un élément `<a>` avec une classe `.nav-link` permettent de définir les options de navigation :

```
<nav class="navbar navbar-expand-sm bg-light">
  <div class="container-fluid">
    <!-- Liens -->
    <ul class="navbar-nav">
      <li class="nav-item"><a class="nav-link" href="#">Link 1</a></li>
      <li class="nav-item"><a class="nav-link" href="#">Link 2</a></li>
      <li class="nav-item"><a class="nav-link" href="#">Link 3</a></li>
    </ul>
  </div>
</nav>
```

Lien 1 Lien 2 Lien 3

Figure 80 : Barre de navigation standard

Barre de navigation verticale

Une barre de navigation verticale peut être obtenue en supprimant la classe `.navbar-expand-*` :

```
<nav class="navbar bg-light">
  <ul class="navbar-nav">
    <li class="nav-item"><a class="nav-link" href="#">Lien 1</a></li>
    <li class="nav-item"><a class="nav-link" href="#">Lien 2</a></li>
    <li class="nav-item"><a class="nav-link" href="#">Lien 3</a></li>
  </ul>
</nav>
```

Lien 1
Lien 2
Lien 3

Figure 81 : Barre de navigation verticale

03 - Utiliser les composants Bootstrap

Barres de navigation



Barre de navigation centrée

La classe `.justify-content-center` permet centrer la barre de navigation :

```
<nav class="navbar navbar-expand-sm bg-light justify-content-center">
  <ul class="navbar-nav">
    <li class="nav-item"><a class="nav-link" href="#">Lien1</a></li>
    <li class="nav-item"><a class="nav-link" href="#">Lien2</a></li>
    <li class="nav-item"><a class="nav-link" href="#">Lien3</a></li>
  </ul>
</nav>
```

Lien 1 Lien 2 Lien 3

Figure 82 : Barre de navigation centrée

Barre de navigation coloriée

Les classes `.navbar-light` et `navbar-dark` associées aux classes `bg` permettent de colorer la barre de navigation :

```
<nav class="navbar navbar-expand-sm bg-light navbar-light">
  <ul class="navbar-nav">
    <li class="nav-item active"><a class="nav-link" href="#">Accueil</a></li>
    <li class="nav-item"><a class="nav-link" href="#">Lien1</a></li>
    <li class="nav-item"><a class="nav-link" href="#">Lien2</a></li>
    <li class="nav-item"><a class="nav-link disabled" href="#">Desactivé</a></li>
  </ul>
</nav>
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">...</nav>
<nav class="navbar navbar-expand-sm bg-primary navbar-dark">...</nav>
<nav class="navbar navbar-expand-sm bg-success navbar-dark">...</nav>
<nav class="navbar navbar-expand-sm bg-info navbar-dark">...</nav>
<nav class="navbar navbar-expand-sm bg-warning navbar-dark">...</nav>
<nav class="navbar navbar-expand-sm bg-danger navbar-dark">...</nav>
<nav class="navbar navbar-expand-sm bg-secondary navbar-dark">...</nav>
```



CHAPITRE 3

Utiliser les composants Bootstrap

1. Barres de navigation
2. **Menus déroulants**
3. Pagination, badges et alertes
4. Barres de progression
5. Cartes (cards)

03 - Utiliser les composants Bootstrap

Les menus déroulants

Menu déroulant avec bouton simple

```
<div class="btn-group">
    <button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false"> Menu déroulant <span class="caret"></span></button>
    <ul class="dropdown-menu">
        <li><a href="#" title="Lien 1">Lien 1</a></li>
        <li><a href="#" title="Lien 2">Lien 2</a></li>
        <li><a href="#" title="Lien 3">Lien 3</a></li>
        <li role="separator" class="divider"></li>
        <li><a href="#" title="Lien 4">Lien 4</a></li>
    </ul>
</div>
```

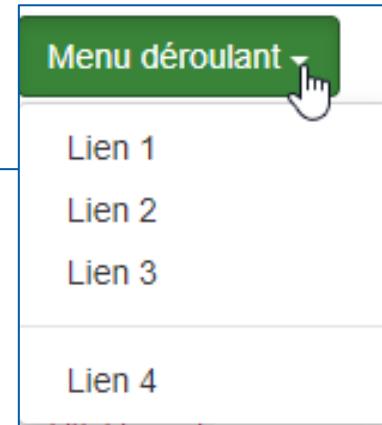


Figure 83 : Menu déroulant avec bouton simple

03 - Utiliser les composants Bootstrap

Les menus déroulants

Menu déroulant avec bouton divisé

```
<div class="btn-group">
    <button type="button" class="btn btn-default">Menu déroulant</button>
    <button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
        <span class="caret"></span>
        <span class="sr-only">Toggle Dropdown</span>
    </button>
    <ul class="dropdown-menu">
        <li><a href="#" title="Lien 1">Lien 1</a></li>
        <li><a href="#" title="Lien 2">Lien 2</a></li>
        <li><a href="#" title="Lien 3">Lien 3</a></li>
        <li role="separator" class="divider"></li>
        <li><a href="#" title="Lien 4">Lien 4</a></li>
    </ul>
</div>
```

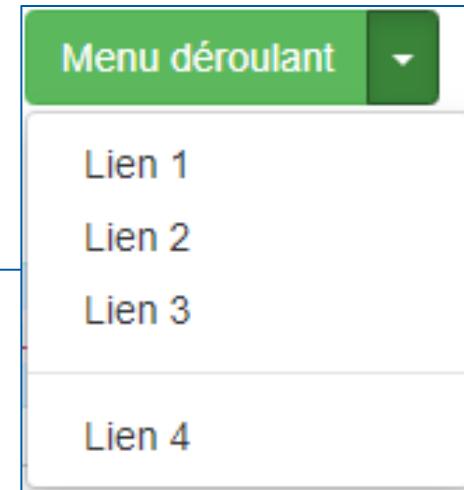


Figure 84 : Menu déroulant avec bouton divisé



CHAPITRE 3

Utiliser les composants Bootstrap

1. Barres de navigation
2. Menus déroulants
- 3. Pagination, badges et alertes**
4. Barres de progression
5. Cartes (cards)

Les alertes

Les alertes sont créées avec la classe `.alert`, suivie de l'une des classes contextuelles `.alert-success`, `.alert-info`, `.alert-warning`, `.alert-danger`, `.alert-primary`, `.alert-secondary`, `.alert-light` ou `.alert-dark` :

```
<div class="alert alert-success">
  <strong> Succès ! </strong> Indique une action réussie ou positive.
</div>
```

Succès ! Cette boîte d'alerte peut indiquer une action réussie ou positive.

Info ! Cette boîte d'alerte pourrait indiquer un changement ou une action informative.

Avertissement ! Cette boîte d'alerte peut indiquer un avertissement qui peut nécessiter votre attention.

Figure 85 : Les alertes

Les liens d'alertes

La classe `.alert-link`, ajoutée aux liens à l'intérieur de la boîte d'alerte, permet de créer des "liens de couleur assortie" :

```
<div class="alert alert-info">
    <strong> Information! </strong> Vous deviez<a href="#" class="alert-link"> lire ce message</a>.
</div>
<div class="alert alert-warning">
    <strong> Warning! </strong> Vous deviez<a href="#" class="alert-link"> lire ce message</a>.
</div>
<div class="alert alert-danger">
    <strong> Danger! </strong> Vous deviez<a href="#" class="alert-link"> lire ce message</a>.
</div>
```

Information! Vous deviez lire ce message.

Warning! Vous deviez lire ce message.

Danger! Vous deviez lire ce message.

Figure 86 : Liens d'alertes

Fermeture d'alertes

Pour fermer le message d'alerte, on utilise la classe **.alert-dismissible** au conteneur d'alerte.

Pour fermer l'alerte, utilisez les classes **class="btn-close"** et **data-bs-dismiss="alert"** à un lien ou à un élément de bouton.

```
<div class="alert alert-success alert-dismissible">
  <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
  <strong>Succès!</strong> Cette boîte d'alerte peut indiquer une action réussie ou positive.
</div>
```



Succès! Cette boîte d'alerte peut indiquer une action réussie ou positive.



Figure 87 : fermeture d'alerte

Alertes animées

Les classes **.fade** et **.show** ajoutent un effet de décoloration lors de la fermeture du message d'alerte :

```
<div class="alert alert-danger alert-dismissible fade show">
```

Pagination

La pagination permet de structurer le contenu des pages (les résultats d'une opération de recherche par exemple)

Pour créer une pagination de base, ajouter la classe **.pagination** à un élément ``. Ajouter aussi la classe **.page-item** (associée aux éléments ``) et la classe **.page-link** associée à chaque lien à l'intérieur `` :

```
<ul class="pagination">
    <li class="page-item"><a class="page-link" href="#">Précédent</a></li>
    <li class="page-item"><a class="page-link" href="#">1</a></li>
    <li class="page-item"><a class="page-link" href="#">2</a></li>
    <li class="page-item"><a class="page-link" href="#">3</a></li>
    <li class="page-item"><a class="page-link" href="#">Suivant</a></li>
</ul>
```

Précédent 1 2 3 Suivant

Figure 88 : Pagination

Pagination (état actif)

La classe **.active** est utilisée pour « surligner » la page en cours :

```
<ul class="pagination">
  <li class="page-item"><a class="page-link" href="#">Précédent</a></li>
  <li class="page-item"><a class="page-link" href="#">1</a></li>
  <li class="page-item active"><a class="page-link" href="#">2</a></li>
  <li class="page-item"><a class="page-link" href="#">3</a></li>
  <li class="page-item"><a class="page-link" href="#">Suivant</a></li>
</ul>
```

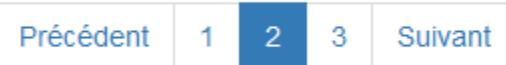


Figure 89 : Etat actif d'une pagination

03 - Utiliser les composants Bootstrap

Pagination, badges et alertes

Dimensionnement de la pagination

Les blocs de pagination peuvent également être dimensionnés à une taille plus grande ou plus petite.

Ajouter la classe `.pagination-lg` pour les grands blocs ou `.pagination-sm` pour les petits blocs.

```
<ul class="pagination pagination-lg">
    <li class="page-item"><a class="page-link" href="#">Précédent</a></li>
    <li class="page-item"><a class="page-link" href="#">1</a></li>
    <li class="page-item"><a class="page-link" href="#">2</a></li>
    <li class="page-item"><a class="page-link" href="#">3</a></li>
    <li class="page-item"><a class="page-link" href="#">Suivant</a></li>
</ul>
    <ul class="pagination pagination-sm">
        <li class="page-item"><a class="page-link" href="#">Précédent</a></li>
        <li class="page-item"><a class="page-link" href="#">1</a></li>
        <li class="page-item"><a class="page-link" href="#">2</a></li>
        <li class="page-item"><a class="page-link" href="#">3</a></li>
        <li class="page-item"><a class="page-link" href="#">Suivant</a></li>
</ul>
```

Précédent 1 2 3 Suivant

Précédent 1 2 3 Suivant

Figure 90 : Dimensionnement d'une pagination

Alignment de la pagination

Les classes utilitaires permettent de modifier l'alignement de la pagination :

```
<!-- Aligné à gauche (par défaut) -->
<ul class="pagination" style="margin :20px 0">
    <li class="page-item">...</li>
</ul>
<!-- Centré -->
<ul class="pagination justify-content-center" style="margin :20px 0">
    <li class="page-item">...</li>
</ul>
<!-- Aligné à droite -->
<ul class="pagination justify-content-end" style="margin :20px 0">
    <li class="page-item">...</li>
</ul>
```

Précédent 1 2 3 Suivant

Précédent 1 2 3 Suivant

Précédent 1 2 3 Suivant

Figure 91 : Alignement d'une pagination

03 - Utiliser les composants Bootstrap

Pagination, badges et alertes



Breadcrumbs

- Une autre forme de pagination est le fil d'Ariane :

HTML / CSS / Bootstrap / Exemples

- Les classes **.breadcrumb** et **.breadcrumb-item** indiquent l'emplacement de la page actuelle dans une hiérarchie de navigation :

```
<ul class="breadcrumb">
  <li class="breadcrumb-item"><a href="#">HTML</a></li>
  <li class="breadcrumb-item"><a href="#">CSS</a></li>
  <li class="breadcrumb-item"><a href="#">Bootstrap</a></li>
  <li class="breadcrumb-item active">Exemples</li>
</ul>
```

Breadcrumbs

La classe **.badge** en combinaison avec une classe contextuelle (comme **.bg-secondary**) dans les éléments **** permet de créer des badges rectangulaires. Ces badges s'adaptent à la taille de l'élément parent :

```
<h1>Exemple<span class="badge bg-secondary">Nouveau</span></h1>
```

Badges contextuels

Utilisation des classes contextuelles (**.bg-***) pour changer la couleur d'un badge :

```
<span class="badge bg-primary">Primaire</span>
<span class="badge bg-secondary">Secondaire</span>
<span class="badge bg-success">Succès</span>
<span class="badge bg-danger">Danger</span>
<span class="badge bg-warning">Avertissement</span>
<span class="badge bg-info">Info</span>
<span class="badge bg-light">Léger</span>
<span class="badge bg-dark">Sombre</span>
```

Primaire Secondaire Succès Danger Avertissement Info Léger Sombre

Figure 92 : Breadcrumbs



CHAPITRE 3

Utiliser les composants Bootstrap

1. Barres de navigation
2. Menus déroulants
3. Pagination, badges et alertes
4. **Barres de progression**
5. Cartes (cards)

Barre de progression

Pour créer une barre de progression par défaut, on utilise la classe `.progress` à un élément conteneur et on ajoute la classe `.progress-bar` à son élément enfant :

```
<div class="progress">
    <div class="progress-bar" style="width :70%"></div>
</div>
```



Figure 93 : Barre de progression

Hauteur de la barre de progression

```
<div class="progress" style="height :20px">
    <div class="progress-bar" style="width :40%;height :20px"></div>
</div>
```



Figure 94 : hauteur de la barre de progression

Étiquettes de barre de progression

Ajouter un texte à l'intérieur de la barre de progression pour afficher le pourcentage visible :

```
<div class="progress">
    <div class="progress-bar" style="width :70%">70%</div>
</div>
```

A horizontal progress bar consisting of two colored segments. The first segment is blue and spans 70% of the bar's length, with the text "70 %" centered inside it in white. The second segment is light gray and spans the remaining 30% of the bar's length.

Figure 95 : étiquette de la barre de progression

Barres de progression colorées

```
<!-- Bleu -->
<div class="progress">
    <div class="progress-bar" style="width :10%"></div>
</div>
<!-- Vert -->
<div class="progress">
    <div class="progress-bar bg-success" style="width :20%"></div>
<!-- Orange -->
<div class="progress">
    <div class="progress-bar bg-warning" style="width :40%"></div>
</div>
```

Barre de progression rayée

La classe `.progress-bar-striped` permet d'ajouter des rayures aux barres de progression :

```
<div class="progress">
  <div class="progress-bar progress-bar-striped" style="width :40%"></div>
</div>
```

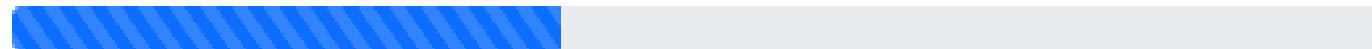


Figure 96: barre de progression rayée

Barre de progression animée

```
<div class="progress-bar progress-bar-striped progress-bar-animated" style="width :40%"></div>
```



CHAPITRE 3

Utiliser les composants Bootstrap

1. Barres de navigation
2. Menus déroulants
3. Pagination, badges et alertes
4. Barres de progression
5. **Cartes (cards)**

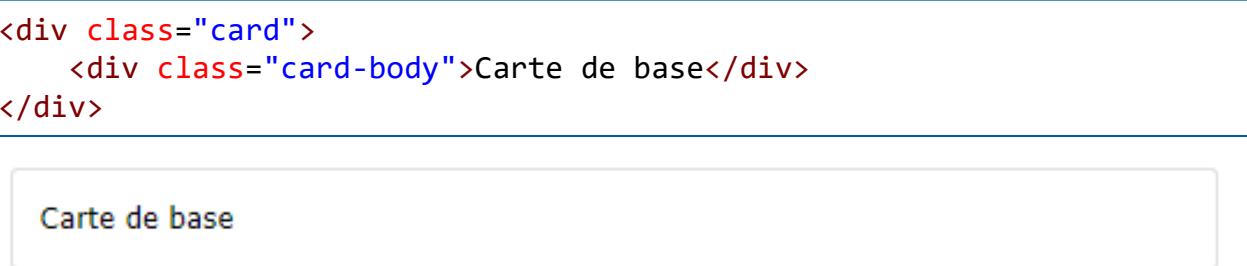
Cartes

Une carte dans Bootstrap 5 est une boîte bordée avec un peu de rembourrage autour de son contenu. Il comprend des options pour les entêtes, les pieds de page, le contenu, les couleurs, etc.

Carte de base

Une carte de base est créée avec la classe `.card` et le contenu à l'intérieur de la carte a une classe `.card-body` :

```
<div class="card">
    <div class="card-body">Carte de base</div>
</div>
```

A screenshot of a web browser showing a single card component. The card has a light gray background and a thin black border. Inside, the text "Carte de base" is centered in a standard black font.

Carte de base

Figure 97 : carte de base

Entête et pied de carte

La classe `.card-header` ajoute une entête à la carte et la classe `.card-footer` ajoute un pied de page à la carte.

Pour ajouter une couleur d'arrière-plan à la carte, utilisez des classes contextuelles (`.bg-primary`, `.bg-success`, `.bg-info`, `.bg-warning`, `.bg-danger`, `.bg-secondary`, `.bg-dark`, `.bg-light`).

```
<div class="card">
    <div class="card-header">Entête</div>
    <div class="card-body">Teneur</div>
    <div class="card-footer">Bas de page</div>
</div>
```

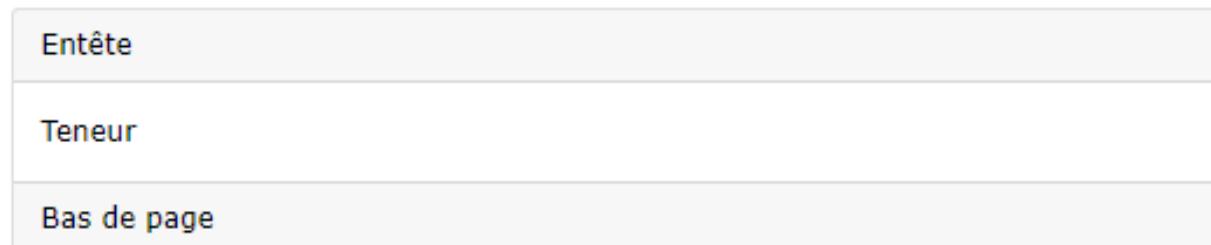


Figure 98 : entête et pieds de carte

Titres, texte et liens des cartes

Pour ajouter des titres de carte à n'importe quel élément de titre on utilise la classe `.card-title`.

La classe `.card-text` est utilisée pour supprimer les marges inférieures d'un élément `<p>` s'il s'agit du dernier enfant (ou du seul) à l'intérieur de `.card-body`.

La classe `.card-link` ajoute une couleur bleue à n'importe quel lien et un effet de survol :

```
<div class="card">
  <div class="card-body">
    <h4 class="card-title">Titre de la carte</h4>
    <p class="card-text">Quelques exemples de texte. Quelques exemples de texte. </p>
    <a href="#" class="card-link">Lien carte</a>
    <a href="#" class="card-link">Autre lien</a>
  </div>
</div>
```

Titre de la carte

Quelques exemples de texte. Quelques exemples de texte.

[Lien carte](#) [Autre lien](#)

Figure 99 : titres, liens et texte des cartes

Images des cartes

On utilise les classes `.card-img-top` ou `.card-img-bottom` avec une balise `` pour placer l'image en haut ou en bas à l'intérieur de la carte.

```
<div class="card" style="width :400px">
    
    <div class="card-body">
        <h4 class="card-title">Sami LAMRINI</h4>
        <p class="card-text">Quelques exemples de textes.</p>
        <a href="#" class="btn btn-primary">Voir Profil</a>
    </div>
</div>
```

Superposition des images des cartes

Il est possible de transformer une image en arrière-plan de carte et utiliser la classe `.card-img-overlay` pour ajouter du texte au-dessus de l'image :

```
<div class="card" style="width :500px">
    
    <div class="card-img-overlay">
        <h4 class="card-title">Sami LAMRINI</h4>
        <p class="card-text">Quelques exemples de textes.</p>
        <a href="#" class="btn btn-primary">Voir Profil</a>
    </div>
</div>
```



PARTIE 5

Héberger un site web

Dans ce module, vous allez :

- Héberger correctement un site statique intranet
- Héberger correctement un site statique en ligne
- Maîtriser le transfert de fichiers (FTP)



05 heures

CHAPITRE 1

Déployer en intranet



Ce que vous allez apprendre dans ce chapitre :

- Définir un réseau local
- Installer un serveur Web
- Utiliser les protocoles de transfert de fichiers
- Créer des comptes FTP
- Mettre à jour du contenu



03 heures



CHAPITRE 1

Déployer en intranet

- 1. Définition d'un réseau local**
2. Installation d'un serveur Web
3. Protocoles de transfert de fichiers
4. Création des comptes FTP
5. Transfert des fichiers
6. Mise à jour du contenu

01 - Déployer en intranet

Définition d'un réseau local



Réseau local

Un réseau local, appelé aussi réseau local d'entreprise (RLE) ou en anglais LAN (Local Area Network), est un réseau informatique à une échelle géographique relativement restreinte, par exemple une salle informatique, un bâtiment ou un site d'entreprise [9]. Grâce à ce concept, datant de 1970, les employés d'une entreprise ont à disposition un système permettant :

- D'échanger des informations
- De communiquer
- D'avoir accès à des services divers



CHAPITRE 1

Déployer en intranet

1. Définition d'un réseau local
2. **Installation d'un serveur Web**
3. Protocoles de transfert de fichiers
4. Création des comptes FTP
5. Transfert des fichiers
6. Mise à jour du contenu

01 - Déployer en intranet

Installation d'un serveur Web



Installer le serveur Apache

L'installation du serveur Apache sur une machine Windows ne nécessite aucune installation préalable.

Voici la procédure à suivre :

- Enregistrez le pack de logiciel ZIP dans un dossier local sur votre ordinateur
- Décompressez le fichier Zip en effectuant un double-clic sur l'icône concernée
- Sélectionnez le dossier Apache24 et copiez-le sur C :\
- Désormais, tous les fichiers nécessaires à l'installation se trouvent sous C :\Apache24

01 - Déployer en intranet

Installation d'un serveur Web



Tester le serveur Apache

À la suite de l'installation du serveur Apache, il est recommandé d'effectuer un test :

1. Aller dans le répertoire C:\Apache24
2. Ouvrir le dossier "bin".
3. Démarrez le fichier exécutable "httpd" en effectuant un double-clic sur ce dernier afin de démarrer le serveur Apache.

En cas d'erreur, le système affiche deux rapports d'erreur pour indiquer les composants à installer ou le blocage du serveur Apache par le pare-feu.

01 - Déployer en intranet

Installation d'un serveur Web

Que faire en cas d'alerte de sécurité Windows ?

- Les ordinateurs disposent d'un pare-feu empêchant les programmes d'accéder au réseau de manière injustifiée. Afin de pouvoir rendre le serveur Apache disponible sur l'environnement de travail ou sur le réseau local, il est nécessaire d'autoriser la communication d'Apache.
- Une fenêtre s'ouvrira au démarrage du serveur Apache en informant que le pare-feu a bloqué certaines fonctions du serveur Web. Il est nécessaire de cliquer sur "Autoriser l'accès" pour permettre l'accès au serveur.

Que faire en cas d'alerte de sécurité Windows ?

- Le serveur Apache est fonctionnel : on peut le démarrer en effectuant un double-clic sur le fichier "httpd". Le lancement ouvrira automatiquement une nouvelle fenêtre de commande noire. Pour arrêter le serveur, il faut fermer cette fenêtre.
- Pour accéder au serveur Web par le biais d'un navigateur: taper "localhost" ou l'adresse ip locale "127.0.0.1" dans la barre de recherche.



CHAPITRE 1

Déployer en intranet

1. Définition d'un réseau local
2. Installation d'un serveur Web
- 3. Protocoles de transfert de fichiers**
4. Création des comptes FTP
5. Transfert des fichiers
6. Mise à jour du contenu

FTP (File Transfer Protocol)

- Le protocole de transferts de fichiers FTP est une méthode populaire de transfert de fichiers. FTP échange des données en utilisant deux canaux distincts : le canal de commande pour authentifier l'utilisateur et le canal de données pour transférer les fichiers.
- Cependant, ces deux canaux FTP ne sont pas chiffrés, ce qui implique que les données envoyées ont un risque d'être exploitées même s'il nécessite un nom d'utilisateur et un mot de passe pour authentifier l'accès.

SFTP

- SFTP est un protocole de transferts de fichiers sécurisé qui permet de transférer des fichiers en toute sécurité par SSL/TLS (Secure Sockets Layer/Transport Layer Security). Les transferts SFTP peuvent être authentifiés par des méthodes de prise en charge comme les certificats de clients, les certificats de serveurs et les mots de passe.

FTPS (Secure File Transfer Protocol)

- SFTP signifie FTP part SSH (Secure Shell). Il s'agit d'un protocole FTP sécurisé représentant une excellente alternative aux outils FTP non sécurisés ou aux scripts manuels. SFTP permet d'échanger des données via une connexion SSH qui offre un niveau de protection élevé pour les partages de fichiers entre systèmes, cloud, collaborateurs, etc.

HTTP (Hyper Text Transfer Protocol)

- HTTP (Hyper Text Transfer Protocol) est le pilier de la communication des données. Il définit le format des messages par lesquels les navigateurs et les serveurs web communiquent et détermine comment un navigateur doit répondre à une requête.
- HTTP utilise TCP (Transmission Control Protocol) comme protocole secondaire, qui est aussi un protocole apatriide. Cela signifie que chaque commande est exécutée indépendamment et qu'aucune information de session n'est conservée par le destinataire.

HTTPS (Hyper Text Transfer Protocol Secure)

- HTTPS (Hyper Text Transfer Protocol Secure) est la variante sécurisée du protocole HTTP où les communications sont chiffrées par TLS ou SSL.



CHAPITRE 1

Déployer en intranet

1. Définition d'un réseau local
2. Installation d'un serveur Web
3. Protocoles de transfert de fichiers
- 4. Création des comptes FTP**
5. Transfert des fichiers
6. Mise à jour du contenu

01 - Déployer en intranet

Création des comptes FTP

Création des comptes FTP

- Les comptes FTP permettent de transférer et récupérer rapidement des fichiers depuis l'ordinateur personnel vers le serveur (distant) et vice versa. La solution logicielle à utiliser est nommée "**client ftp**" qui, installé sur l'ordinateur, se connecte au "serveur ftp"
- Sous Windows, il est conseillé d'utiliser le logiciel libre comme "**FileZilla**" pour se connecter en FTP :
- Télécharger FileZilla depuis le site SourceForge
- Lancer son installation
- Une fois l'installation achevée, lancer FileZilla en cliquant sur son icône :

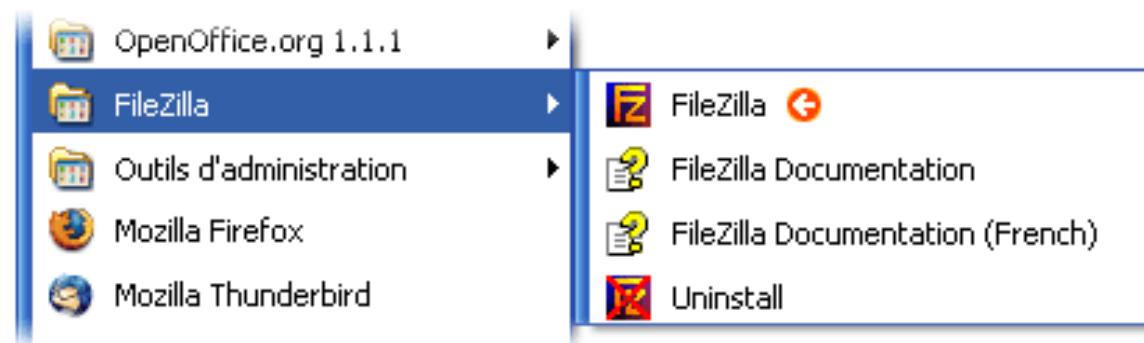


Figure 100 : Se connecter à FileZilla

01 - Déployer en intranet

Création des comptes FTP

Se connecter à son compte FTP avec FileZilla

L'interface par défaut de FileZilla est la suivante :

- L'écran est divisé en plusieurs parties :
 1. en haut, la barre de connexion rapide
 2. à gauche, le contenu du disque dur
 3. à droite, le listing de ce que contient le compte FTP pour l'hébergement du site internet
 4. un historique des messages pour les diagnostics d'erreur
 5. les transferts en cours.

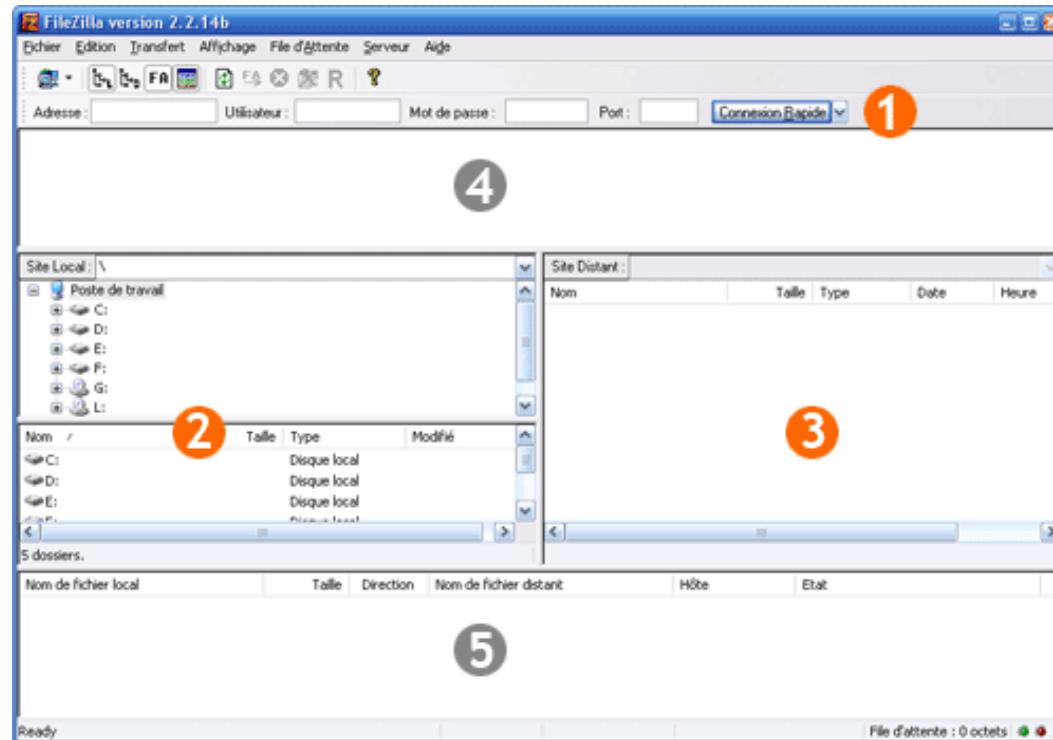


Figure 101 : Interface de FileZilla

01 - Déployer en intranet

Création des comptes FTP

Créer un compte FTP avec FileZilla

Lors de la première connexion au bureau, il n'existe aucun compte FTP. Il faut en créer un :

Cliquer sur le lien Comptes FTP dans le menu du bureau (cadre de gauche).

Cliquer sur le menu **edit** puis **users** à droite, cliquer sur **add** puis saisir le nom de l'utilisateur ftp.

Dans **password**, saisir le mot de passe d'accès pour cet utilisateur.

Puis dans **Group membership**, sélectionner le groupe auquel cet utilisateur sera rattaché.

Autoriser FileZilla sur le pare-feu de Windows Defender

Le pare-feu « Windows Defender » bloque les connexions entrantes vers le serveur FTP FileZilla.

Pour l'autoriser, il faut autoriser le programme : C:\Program Files (x86)\FileZilla Server\FileZilla Server.exe



Figure 102 : Création d'un compte sur FileZilla



CHAPITRE 1

Déployer en intranet

1. Définition d'un réseau local
2. Installation d'un serveur Web
3. Protocoles de transfert de fichiers
4. Création des comptes FTP
5. **Transfert des fichiers**
6. Mise à jour du contenu

01 - Déployer en intranet

Transfert des fichiers

Se connecter au serveur FTP

Une fois connecté au serveur FTP, on remarque deux panneaux à l'écran. Celui de gauche affiche les fichiers et les dossiers stockés sur l'ordinateur local, celui de droit affiche les contenus des ordinateurs à distance.

Dans le panneau de gauche, sélectionner les fichiers à charger sur le serveur. La plupart des clients FTP permettent de glisser-déposer les fichiers d'un panneau à l'autre pour initier le chargement.

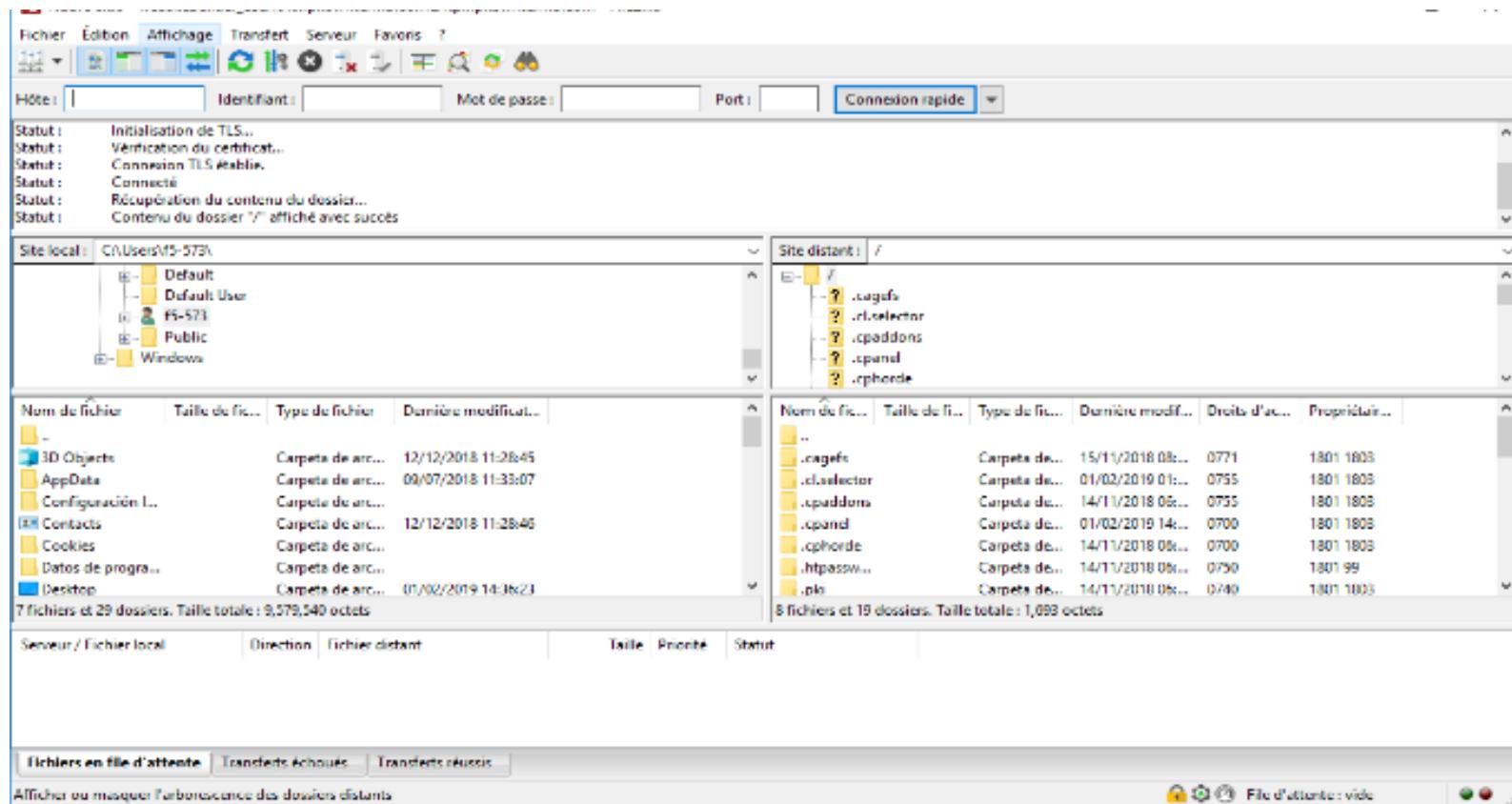


Figure 103 : Transfert des fichiers



CHAPITRE 1

Déployer en intranet

1. Définition d'un réseau local
2. Installation d'un serveur Web
3. Protocoles de transfert de fichiers
4. Création des comptes FTP
5. Transfert des fichiers
- 6. Mise à jour du contenu**

01 - Déployer en intranet

Mise à jour du contenu

Mise à jour du contenu

Pour mettre à jour le contenu de votre site web déjà transféré avec FileZilla, il suffit d'envoyer les nouveaux fichiers et remplacer les anciens.

Dans FileZilla, sélectionner **Fichier > gestionnaire de site** pour ouvrir un panneau dans lequel on peut saisir les détails du compte FTP.

Nom de l'hébergeur/du serveur : correspond au nom de domaine avec un préfixe FTP (par exemple, mydomain.com).

Port : le numéro du port utilisé pour les connexions FTP au serveur. En général, le port 21 est utilisé.

Type de login : Choisir Normal, plutôt que sur Anonyme.

Utilisateur : nom d'utilisateur de compte FTP.

Mot de passe : mot de passe de compte FTP.

Une fois connecté, il sera possible de copier des fichiers depuis le disque dur vers le site

Le résultat sur le site peut être consulté sur le navigateur web.



Figure 104 : Se connecter à un serveur

CHAPITRE 2

Déployer en ligne



Ce que vous allez apprendre dans ce chapitre :

- Identifier les critères de choix d'un hébergeur
- Effectuer un hébergement (gratuit, payant)



02 heures



CHAPITRE 2

Déployer en ligne

1. Critères de choix d'un hébergeur
2. Hébergement (gratuit, payant)

02 - Déployer en ligne

Critères de choix d'un hébergeur

Choisir un hébergeur

1. **La vitesse d'accès au site** : Ce critère sera aussi déterminant pour assurer un bon référencement.
2. **La limite de bande passante** : Les hébergeurs proposent différentes options – selon le type de site que vous lancez, choisissez celle qui vous convient le mieux.
3. **Les services et options proposées** : Certains hébergeurs proposent des services et options d'installation de certificats SSL (sécurité), de multiple d'adresses emails, etc. À vous de choisir l'offre qui vous convient et de comparer parmi les choix proposés.
4. **La qualité et la rapidité du support** : Pour en juger, vous pouvez vous baser sur les avis et les témoignages des clients des hébergeurs que vous avez identifiés.
5. **Les services courants** : Certains hébergeurs proposent une installation facilitée de WordPress, Woo Commerce et d'autres services courants.
6. **La sécurité du site** : Quel niveau de sécurité offre cet hébergeur ? Comparez les différents niveaux de sécurité de chaque prestataire.
7. **Les backups** : Il s'agit de la sauvegarde régulière de toutes les données de votre site internet, de sorte qu'en cas de problème, vous puissiez rapidement récupérer vos données.
8. **Le prix** : Enfin, il s'agit de comparer le prix de tous les prestataires en fonction des prestations qu'ils proposent ou que vous avez choisies selon vos besoins.



CHAPITRE 2

Déployer en ligne

1. Critères de choix d'un hébergeur
2. **Hébergement (gratuit, payant)**

02 - Déployer en ligne

Hébergement (gratuit / payant)

Hébergement d'un site Web

Un hébergement web permet de publier le site sur internet et de le rendre accessible au monde entier. La mise en ligne du site se fait en quelques étapes :

- Choix d'un hébergeur de qualité proposant des services à moindre coût.
- Choix du nom de domaine offert avec l'hébergement.
- Téléchargement des fichiers en ligne.

02 - Déployer en ligne

Hébergement (gratuit / payant)

Hébergement d'un site Web

- Un site web est un ensemble de fichiers reliés entre eux et écrits en utilisant des langages web (HTML, CSS...). Ces fichiers peuvent être déposés dans l'espace web via un logiciel utilisant un protocole de transfert (FTP), comme FileZilla.
- Mais vous pouvez aussi utiliser le service de transfert rapide de l'hébergeur.

 images	31/10/2018 13:49
 js	26/10/2018 14:51
 index.html	23/11/2018 16:34
 manifest.json	23/11/2018 11:04
 service-worker.js	25/10/2018 11:48
 tutoriel.txt	14/11/2018 17:02

Figure 105 : Exemple de contenu d'un site web

02 - Déployer en ligne

Hébergement (gratuit / payant)



Héberger un site statique en intranet

- L'hébergement Web est un service qui permet de publier votre site en direct sur Internet. Un hébergeur, également appelé fournisseur d'hébergement, est une entreprise qui fournit les outils dont vous avez besoin pour accomplir cela.
- Plus particulièrement, cela inclut un serveur. Les serveurs stockent les fichiers de site Web et les navigateurs Internet peuvent communiquer avec eux pour transmettre les pages de votre site aux visiteurs.
- Il est important de noter qu'il existe deux manières d'héberger un site Web : localement, à l'extérieur/publiquement.
- Un site Web hébergé localement est stocké sur votre propre ordinateur tandis qu'un site hébergé en externe est stocké sur le serveur d'un fournisseur (hébergeur).

02 - Déployer en ligne

Hébergement (gratuit, payant)

Bande passante et espace de stockage

- La bande passante d'hébergement est le total des données qui seront transférées à l'utilisateur sur demande, du serveur au site Web. L'espace de stockage est l'espace total dont dispose un serveur pour stocker toutes ces données.
- L'hébergement payant n'a aucune restriction en matière de stockage et de bande passante, tandis que les hébergeurs gratuits limitent ces ressources.

Caractéristiques et mises à niveau

- Les fonctionnalités de l'hébergement gratuit sont limitées. On ne peut pas par exemple avoir accès au cPanel et ainsi configurer des éléments tels que des sauvegardes, des certificats SSL, des comptes de messagerie, etc...

Nom de domaine et image de marque

- Les services gratuits et payants fournissent un nom de domaine mais avec une différence :
- Les hébergeurs gratuits ajoutent leur nom dans l'URL du domaine
- Les plans d'hébergements payants, en revanche, offrent la possibilité de créer un nom de domaine personnalisé qui aura une validité d'un an.

Confidentialité et sécurité

- La sécurité est l'un des aspects les plus importants d'un hébergement. Des fonctionnalités de sécurité faibles risquent de générer une fuite d'informations ou de vol d'informations sensibles des utilisateurs et des visiteurs (comme les informations personnelles, des informations financières, etc).

02 - Déployer en ligne

Hébergement (gratuit, payant)



Support technique et assistance

- Un service d'hébergement offre également le support technique. Le rôle de l'équipe d'assistance est d'assurer la fiabilité du service aux utilisateurs.

Performances et fiabilité

- Les hébergeurs gratuits n'utilisent pas suffisamment de ressources, ce qui peut limiter les performances du site Web.
- Les fournisseurs d'hébergement payant doivent mettre à niveau leurs propres systèmes pour offrir les meilleures performances : par exemple, utiliser des disques SSD au lieu de disques durs, qui sont plus rapides.

02 - Déployer en ligne

Hébergement correct d'un site web statique en intranet



Hébergement d'un site Web en utilisant votre PC Windows comme serveur WAMP

- L'approche d'héberger un site sur Windows signifie transformer l'ordinateur en serveur WAMP.
- **WAMP** est un acronyme qui signifie **Windows, Apache, MySQL et PHP**. Ce sont les éléments qui composent le serveur.
- Il existe un certain nombre de programmes gratuits disponibles (comme WampServer et XAMPP).
- Ces logiciels permettent à l' ordinateur d'agir comme un serveur qui héberge les fichiers du site Web.

02 - Déployer en ligne

Hébergement (gratuit / payant)

Hébergement d'un site Web en utilisant votre PC Windows comme serveur WAMP

- Après l'installation du logiciel WAMP, il faut lui ajouter un site.
- Sur le tableau de bord, ajouter les fichiers du site dans le « répertoire www ». C'est là que les fichiers du site Web sont hébergés.
- Le site auto-hébergé demeurera en privé. Donc, il est complètement inaccessible.
- Accéder à « Apache» et sélectionner le fichier « httpd.conf». Noter les paramètres suivants :

```
Order Allow,Deny  
Allow from all
```

02 - Déployer en ligne

Hébergement (gratuit / payant)

Bien choisir son nom de domaine

Le nom de domaine est l'adresse du site.

Le nom de domaine est unique. Lors de l'achat d'un pack d'hébergement web, un nom de domaine est offert. Il est important de bien réfléchir à ce nom : il doit bien représenter le site.

Les différentes extensions

Le nom de domaine comprend aussi l'extension (.com, .ma...).

Si le site a une portée nationale, il faut choisir l'extension « .ma », si sa portée est internationale, il faut opter pour l'extension « .com ». Pour une association ou une organisation, utilisez l'extension « .org ».

Si le .com et le .ma ne sont pas disponibles, vous pouvez vous orienter vers un .net même si ce n'est pas, contrairement au .com, le suffixe de marque sur internet.

Les pièges à éviter

- Ne pas utiliser un nom composé avec trop de mots : limitez-vous à 3 - 4 maximum
- Éviter un nom imprononçable
- Faites attention aux mots difficilement orthographiable
- Évitez les traits d'union.