

# Travaux Pratiques HTML et CSS pour Débutants

## Introduction

Ce document regroupe une série de travaux pratiques destinés aux étudiants débutants en développement web. Ces TP couvrent les fondamentaux du HTML et du CSS, depuis la structure de base d'une page web jusqu'à la création d'un site web personnel complet.

Chaque TP est conçu pour être progressif, permettant d'acquérir de nouvelles compétences tout en consolidant les notions précédemment apprises. Les énoncés sont accompagnés de conseils pratiques et de solutions détaillées pour faciliter l'apprentissage autonome.

Ces travaux pratiques vous guideront à travers les concepts essentiels du développement web front-end, vous donnant les bases nécessaires pour créer vos propres sites web et applications.

Bonne découverte et bon apprentissage !

## TP 1 : Structure de base d'une page HTML

### Objectif

Ce premier travail pratique a pour but de vous familiariser avec la structure fondamentale d'une page HTML. Vous apprendrez à créer votre première page web en utilisant les balises HTML essentielles, à organiser le contenu avec des titres et des paragraphes, et à ajouter des commentaires pour améliorer la lisibilité de votre code.

### Énoncé

#### Partie 1 : Création d'un document HTML de base

1. Créez un nouveau fichier nommé `index.html`
2. Ajoutez la déclaration de type de document (doctype) HTML5

3. Créez la structure de base avec les balises `<html>`, `<head>` et `<body>`
4. Dans la section `<head>`, ajoutez :
5. Un titre de page "Ma première page web"
6. Une métadonnée pour définir l'encodage en UTF-8
7. Dans la section `<body>`, ajoutez un titre principal (h1) "Bienvenue sur ma première page web"

## Partie 2 : Enrichissement du contenu

1. Ajoutez un paragraphe de présentation sous le titre principal
2. Créez une section "À propos" avec un titre de niveau 2 (h2)
3. Dans cette section, ajoutez deux paragraphes décrivant vos intérêts
4. Créez une autre section "Mes compétences" avec un titre de niveau 2
5. Ajoutez au moins trois compétences sous forme de titres de niveau 3 (h3), chacun suivi d'un paragraphe explicatif

## Partie 3 : Utilisation des commentaires

1. Ajoutez un commentaire au début du document expliquant son objectif
2. Ajoutez des commentaires avant chaque section principale pour expliquer sa structure
3. Commentez une ligne de code que vous pourriez vouloir modifier plus tard

## Conseils

- Veillez à bien indenter votre code pour le rendre plus lisible
- Utilisez des noms significatifs pour vos éléments
- N'oubliez pas de fermer toutes vos balises
- Vérifiez régulièrement votre page dans un navigateur pour voir le résultat

## Solution

```
<!DOCTYPE html>
<!-- Ce document est ma première page web créée dans le cadre du
TP1 HTML -->
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Ma première page web</title>
</head>
<body>
  <h1>Bienvenue sur ma première page web</h1>
```

```
<p>Cette page est le résultat de mon premier travail pratique en HTML. Je vais vous présenter un peu qui je suis et quelles sont mes compétences.</p>
```

```
<!-- Section de présentation personnelle -->
```

```
<section>
```

```
<h2>À propos</h2>
```

```
<p>Je suis étudiant(e) en informatique et je découvre le développement web. J'aime apprendre de nouvelles technologies et créer des projets innovants.</p>
```

```
<p>Dans mon temps libre, j'apprécie la lecture, les jeux vidéo et les randonnées en pleine nature. Ces activités m'aident à rester créatif(ve) et à trouver de l'inspiration pour mes projets.</p>
```

```
</section>
```

```
<!-- Section des compétences -->
```

```
<section>
```

```
<h2>Mes compétences</h2>
```

```
<h3>Programmation</h3>
```

```
<p>J'ai des connaissances de base en Python et je commence à apprendre le JavaScript. J'aime résoudre des problèmes logiques et créer des algorithmes efficaces.</p>
```

```
<h3>Design</h3>
```

```
<p>Je m'intéresse au design d'interfaces utilisateur et j'essaie d'appliquer les principes de l'UX dans mes projets. J'utilise des outils comme Figma pour créer des maquettes.</p>
```

```
<h3>Communication</h3>
```

```
<p>Je suis capable de travailler en équipe et de communiquer efficacement mes idées. Je peux également rédiger des documentations claires et précises.</p>
```

```
<!-- À développer plus tard : ajouter une section sur les projets réalisés -->
```

```
</section>
```

```
</body>
```

```
</html>
```

## Points importants à retenir

La structure de base d'une page HTML5 comprend toujours les éléments suivants : - La déclaration `<!DOCTYPE html>` qui indique au navigateur qu'il s'agit d'un document HTML5 - L'élément `<html>` qui englobe tout le contenu de la page - L'élément `<head>` qui contient les métadonnées et informations sur le document - L'élément `<body>` qui contient tout le contenu visible de la page

Les balises de titre ( `<h1>` à `<h6>` ) permettent de structurer hiérarchiquement le contenu de votre page, avec `<h1>` étant le titre principal et les autres niveaux représentant des sous-sections de plus en plus spécifiques.

Les commentaires HTML sont invisibles pour l'utilisateur mais très utiles pour les développeurs. Ils permettent d'expliquer le code, de le documenter et de faciliter sa maintenance future.

N'oubliez pas que l'indentation et l'organisation claire du code ne sont pas obligatoires pour le fonctionnement de la page, mais elles sont essentielles pour la lisibilité et la maintenance du code.

## TP 2 : Mise en forme de texte

### Objectif

Ce travail pratique vous permettra d'explorer les différentes possibilités de mise en forme du texte en HTML. Vous apprendrez à utiliser les balises de formatage, à créer différents types de listes, à insérer des liens hypertextes et à utiliser les attributs pour enrichir vos pages web.

### Énoncé

#### Partie 1 : Formatage de texte

1. Créez un nouveau fichier nommé `formatage.html`
2. Ajoutez la structure de base d'une page HTML5
3. Donnez comme titre à votre page "Techniques de formatage HTML"
4. Dans le corps de la page, créez un titre principal "Les bases du formatage de texte en HTML"
5. Rédigez un paragraphe d'introduction sur l'importance du formatage de texte
6. Dans ce paragraphe, utilisez les balises suivantes :
7. `<strong>` pour mettre en évidence une phrase importante
8. `<em>` pour mettre en italique certains mots
9. `<mark>` pour surligner un terme technique
10. `<sub>` et `<sup>` pour ajouter des indices et exposants
11. `<del>` pour barrer un texte incorrect
12. `<ins>` pour souligner un texte ajouté

## Partie 2 : Création de listes

1. Ajoutez une section "Types de listes en HTML" avec un titre de niveau 2
2. Créez une liste non ordonnée des différents types de listes en HTML
3. Pour chaque type de liste, ajoutez une brève description
4. Créez une liste ordonnée des étapes pour réaliser une recette de cuisine simple
5. Créez une liste de définitions avec au moins trois termes HTML et leurs définitions
6. Créez une liste imbriquée (une liste à l'intérieur d'une autre liste) présentant une structure hiérarchique

## Partie 3 : Liens hypertextes

1. Ajoutez une section "Navigation et liens" avec un titre de niveau 2
2. Créez un lien vers le site du W3C (<https://www.w3.org/>)
3. Ajoutez l'attribut `title` à ce lien pour afficher une info-bulle
4. Créez un lien qui s'ouvre dans un nouvel onglet en utilisant l'attribut `target="_blank"`
5. Créez un lien vers une section de votre page en utilisant des ancres (id et #)
6. Créez un lien vers une adresse email

## Conseils

- Testez régulièrement votre page dans différents navigateurs pour voir comment le formatage s'affiche
- Utilisez les balises sémantiques appropriées pour chaque type de contenu
- N'abusez pas des formatages, privilégiez la lisibilité et la cohérence
- Pensez à l'accessibilité : les liens doivent avoir un texte descriptif

## Solution

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Techniques de formatage HTML</title>
</head>
<body>
  <h1>Les bases du formatage de texte en HTML</h1>

  <p>
    Le formatage de texte est <strong>essentiel pour
    améliorer la lisibilité</strong> de vos pages web.
    Il permet de <em>mettre en valeur certaines
```

informations

 et de structurer votre contenu de manière claire.

Les balises HTML comme `<mark>strong`, `em`, et `mark</mark>` vous aident à attirer l'attention sur des éléments importants.

Vous pouvez également utiliser des notations scientifiques comme `H<sub>2</sub>O` ou `E=mc<sup>2</sup>`.

Il est important de ne pas confondre les balises `<del>HTML4</del>` `<ins>HTML5</ins>` pour un code moderne.

`</p>`

`<h2 id="listes">Types de listes en HTML</h2>`

`<ul>`

`<li><strong>Liste non ordonnée (ul)</strong>` : Utilisée pour présenter des éléments sans ordre particulier

`<li><strong>Liste ordonnée (ol)</strong>` : Utilisée pour présenter des éléments dans un ordre spécifique

`<li><strong>Liste de définitions (dl)</strong>` : Utilisée pour présenter des termes et leurs définitions

`</ul>`

`<h3>Recette de crêpes</h3>`

`<ol>`

`<li>Mélanger 250g de farine avec 4 œufs</li>`

`<li>Ajouter progressivement 50cl de lait</li>`

`<li>Incorporer 50g de beurre fondu</li>`

`<li>Laisser reposer la pâte pendant 1 heure</li>`

`<li>Cuire les crêpes dans une poêle chaude</li>`

`</ol>`

`<h3>Glossaire HTML</h3>`

`<dl>`

`<dt>HTML</dt>`

`<dd>HyperText Markup Language, le langage standard pour créer des pages web</dd>`

`<dt>Balise</dt>`

`<dd>Élément de syntaxe qui permet de structurer et formater le contenu d'une page web</dd>`

`<dt>Attribut</dt>`

`<dd>Information supplémentaire ajoutée à une balise pour modifier son comportement ou fournir des métadonnées</dd>`

`</dl>`

`<h3>Structure d'un site web</h3>`

`<ul>`

`<li>Page d'accueil`

`<ul>`

`<li>En-tête`

`<ul>`

`<li>Logo</li>`

```

        <li>Menu de navigation</li>
    </ul>
</li>
<li>Contenu principal
    <ul>
        <li>Articles</li>
        <li>Images</li>
    </ul>
</li>
<li>Pied de page
    <ul>
        <li>Informations de contact</li>
        <li>Liens sociaux</li>
    </ul>
</li>
</ul>
</li>
</ul>
</ul>

<h2>Navigation et liens</h2>

<p>
    Visitez le <a href="https://www.w3.org/" title="World
Wide Web Consortium - Organisation de standardisation du Web">site
officiel du W3C</a> pour en savoir plus sur les standards du
web.
</p>

<p>
    Consultez la <a href="https://developer.mozilla.org/fr/"
target="_blank">documentation MDN</a> pour approfondir vos
connaissances (s'ouvre dans un nouvel onglet).
</p>

<p>
    <a href="#listes">Retourner à la section sur les
listes</a> pour revoir les différents types de listes.
</p>

<p>
    Pour toute question, <a
href="mailto:contact@example.com">contactez-nous par email</a>.
</p>
</body>
</html>

```

## Points importants à retenir

Les balises de formatage de texte en HTML permettent de mettre en valeur certaines parties de votre contenu. Il est important de les utiliser de manière sémantique : - `<strong>` indique une importance forte (pas seulement du texte en gras) - `<em>` indique une emphase (pas seulement du texte en italique) - `<mark>` met en évidence un texte pertinent dans le contexte actuel

Les listes sont des éléments fondamentaux pour structurer l'information : - Les listes non ordonnées (`<ul>`) sont parfaites pour les énumérations sans ordre particulier - Les listes ordonnées (`<ol>`) conviennent aux séquences, instructions ou classements - Les listes de définitions (`<dl>`) sont idéales pour les glossaires et les paires clé-valeur

Les liens hypertextes sont au cœur du web et permettent la navigation : - L'attribut `href` définit la destination du lien - L'attribut `title` fournit des informations supplémentaires (info-bulle) - L'attribut `target="_blank"` ouvre le lien dans un nouvel onglet - Les ancres permettent de naviguer à l'intérieur d'une même page

N'oubliez pas que le formatage doit servir le contenu et non l'inverse. Un bon formatage améliore la lisibilité et l'accessibilité de votre page.

# TP 3 : Intégration d'images et de médias

## Objectif

Ce travail pratique vous permettra d'apprendre à intégrer des éléments multimédias dans vos pages web. Vous découvrirez comment insérer des images, gérer leurs attributs, créer des figures avec légendes et transformer des images en liens cliquables.

## Énoncé

### Partie 1 : Insertion d'images basique

1. Créez un nouveau fichier nommé `images.html`
2. Ajoutez la structure de base d'une page HTML5
3. Donnez comme titre à votre page "Galerie d'images"
4. Dans le corps de la page, créez un titre principal "Ma collection d'images"
5. Insérez trois images différentes (vous pouvez utiliser des URLs d'images libres de droits comme celles de Unsplash ou Pexels)

6. Pour chaque image, assurez-vous d'ajouter :
7. Un attribut `alt` descriptif
8. Des attributs `width` et `height` appropriés

## Partie 2 : Figures et légendes

1. Créez une section "Images avec légendes" avec un titre de niveau 2
2. Utilisez la balise `<figure>` pour encadrer une image
3. Ajoutez une légende à cette image avec la balise `<figcaption>`
4. Créez au moins deux autres figures avec des légendes différentes
5. Organisez ces figures de manière à ce qu'elles s'affichent côte à côte (vous pouvez utiliser des attributs de style inline pour cet exercice)

## Partie 3 : Images cliquables

1. Créez une section "Images cliquables" avec un titre de niveau 2
2. Insérez une image qui, lorsqu'on clique dessus, redirige vers un site web externe
3. Créez une image map avec au moins deux zones cliquables différentes
4. Ajoutez une petite galerie d'images miniatures qui, lorsqu'on clique dessus, affichent une version plus grande de l'image (utilisez des liens vers d'autres pages ou des images plus grandes)

## Conseils

- Utilisez des images de taille raisonnable pour ne pas ralentir le chargement de votre page
- Pensez toujours à l'accessibilité en fournissant des descriptions alternatives pertinentes
- Vérifiez que vos images s'affichent correctement dans différents navigateurs
- Attention aux droits d'auteur : utilisez des images libres de droits ou créditez les auteurs

## Solution

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Galerie d'images</title>
  <style>
    figure {
```

```

        display: inline-block;
        margin: 10px;
        width: 300px;
        border: 1px solid #ccc;
        padding: 10px;
        text-align: center;
    }

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

    .thumbnails {
        display: flex;
        flex-wrap: wrap;
        gap: 10px;
    }

    .thumbnails img {
        width: 100px;
        height: 100px;
        object-fit: cover;
        cursor: pointer;
    }
</style>
</head>
<body>
    <h1>Ma collection d'images</h1>

    <p>Voici quelques exemples d'intégration d'images dans une
page HTML.</p>

    

    

    

    <h2>Images avec légendes</h2>

    <figure>
        
        <figcaption>Un développeur web en plein travail sur son

```

```

code HTML</figcaption>
  </figure>

  <figure>
    
    <figcaption>Structure HTML vue dans un éditeur de code</
figcaption>
  </figure>

  <figure>
    
    <figcaption>Maquette de site web en cours de
développement</figcaption>
  </figure>

  <h2>Images cliquables</h2>

  <p>Cliquez sur l'image ci-dessous pour visiter le site du
W3C :</p>

  <a href="https://www.w3.org/" target="_blank">
    
  </a>

  <h3>Image avec zones cliquables</h3>

  

  <map name="workdesk">
    <area shape="rect" coords="50,50,200,170" href="https://
developer.mozilla.org/fr/docs/Web/HTML" alt="Ordinateur"
title="En savoir plus sur HTML">
    <area shape="circle" coords="350,200,80" href="https://
developer.mozilla.org/fr/docs/Web/CSS" alt="Tasse de café"
title="En savoir plus sur CSS">
    <area shape="poly"
coords="450,100,500,150,450,200,400,150" href="https://
developer.mozilla.org/fr/docs/Web/JavaScript" alt="Carnet de
notes" title="En savoir plus sur JavaScript">
  </map>

  <h3>Galerie de miniatures</h3>

  <div class="thumbnails">

```

```
<a href="https://images.unsplash.com/
photo-1593642532400-2682810df593?w=1200" target="_blank">
  
</a>

<a href="https://images.unsplash.com/
photo-1517336714731-489689fd1ca8?w=1200" target="_blank">
  
</a>

<a href="https://images.unsplash.com/
photo-1516387938699-a93567ec168e?w=1200" target="_blank">
  
</a>

<a href="https://images.unsplash.com/
photo-1531297484001-80022131f5a1?w=1200" target="_blank">
  
</a>
</div>
</body>
</html>
```

## Points importants à retenir

L'intégration d'images est essentielle pour rendre vos pages web plus attrayantes et informatives. Voici les points clés à retenir :

La balise `<img>` est auto-fermante et nécessite au minimum l'attribut `src` qui spécifie l'URL de l'image. L'attribut `alt` est également crucial pour l'accessibilité et le référencement, car il fournit une description textuelle de l'image pour les lecteurs d'écran et les situations où l'image ne peut pas être affichée.

Les attributs `width` et `height` permettent de définir les dimensions de l'image. Il est recommandé de les spécifier pour éviter le réarrangement de la page pendant le chargement des images. Cependant, il est souvent préférable de gérer ces dimensions via CSS pour une meilleure réactivité.

La combinaison des balises `<figure>` et `<figcaption>` permet d'associer une légende à une image de manière sémantique. Cette structure est particulièrement utile pour les illustrations, diagrammes, photos, ou tout contenu qui bénéficie d'une explication.

Les images peuvent être rendues cliquables en les plaçant à l'intérieur d'une balise `<a>`. Cela permet de créer des liens visuellement attrayants ou des galeries d'images.

L'élément `<map>` associé à l'attribut `usemap` permet de créer des images avec plusieurs zones cliquables. Chaque zone est définie par une balise `<area>` qui spécifie sa forme, ses coordonnées et sa destination.

Pour les galeries d'images, une approche courante consiste à afficher des miniatures qui, lorsqu'on clique dessus, affichent une version plus grande de l'image. Cette technique permet d'économiser de la bande passante tout en offrant une expérience utilisateur agréable.

## TP 4 : Création de tableaux

### Objectif

Ce travail pratique vous permettra de maîtriser la création et la stylisation de tableaux en HTML. Vous apprendrez à structurer des données tabulaires, à fusionner des cellules et à appliquer des styles CSS pour améliorer la présentation de vos tableaux.

### Énoncé

#### Partie 1 : Structure de base d'un tableau

1. Créez un nouveau fichier nommé `tableaux.html`
2. Ajoutez la structure de base d'une page HTML5
3. Donnez comme titre à votre page "Tableaux en HTML"
4. Dans le corps de la page, créez un titre principal "Utilisation des tableaux en HTML"
5. Créez un premier tableau simple représentant un calendrier mensuel avec :
  6. Une ligne d'en-tête pour les jours de la semaine
  7. Quatre lignes pour représenter les semaines du mois
  8. Des cellules contenant les numéros des jours

## Partie 2 : Fusion de cellules

1. Créez une section "Emploi du temps" avec un titre de niveau 2
2. Réalisez un emploi du temps hebdomadaire avec :
3. Une colonne pour les heures de la journée
4. Une colonne pour chaque jour de la semaine
5. Des cellules fusionnées horizontalement pour représenter des cours de durées différentes
6. Des cellules fusionnées verticalement pour représenter des activités sur plusieurs jours

## Partie 3 : Stylisation de tableaux

1. Créez une section "Tableau de données" avec un titre de niveau 2
2. Réalisez un tableau présentant des statistiques (par exemple, population de différentes villes)
3. Structurez correctement votre tableau avec `<thead>`, `<tbody>` et `<tfoot>`
4. Utilisez des attributs de style ou une feuille CSS interne pour :
5. Ajouter des bordures au tableau
6. Alternier les couleurs de fond des lignes
7. Centrer le texte dans certaines colonnes
8. Mettre en évidence l'en-tête du tableau

## Conseils

- Utilisez les balises `<caption>` pour donner un titre à vos tableaux
- N'oubliez pas les attributs `scope` pour améliorer l'accessibilité
- Évitez d'utiliser les tableaux pour la mise en page générale de votre site
- Préférez les styles CSS aux attributs HTML obsolètes comme `border` ou `cellpadding`

## Solution

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Tableaux en HTML</title>
  <style>
    table {
      border-collapse: collapse;
```

```
        width: 100%;
        margin-bottom: 20px;
    }

    caption {
        font-weight: bold;
        margin-bottom: 10px;
        font-size: 1.2em;
    }

    th, td {
        border: 1px solid #ddd;
        padding: 8px;
        text-align: center;
    }

    th {
        background-color: #f2f2f2;
        font-weight: bold;
    }

    .calendar td:first-child, .calendar td:last-child {
        background-color: #ffebee;
    }

    .schedule th {
        background-color: #4CAF50;
        color: white;
    }

    .schedule td {
        height: 50px;
        vertical-align: middle;
    }

    .course {
        background-color: #e6f7ff;
    }

    .break {
        background-color: #ffffcc;
    }

    .data-table tbody tr:nth-child(even) {
        background-color: #f9f9f9;
    }

    .data-table tfoot {
        font-weight: bold;
        background-color: #e6e6e6;
    }
</style>
```

```
</head>
```

```
<body>
```

```
  <h1>Utilisation des tableaux en HTML</h1>
```

```
  <p>Les tableaux sont utilisés pour présenter des données  
  sous forme tabulaire. Voici différents exemples d'utilisation.</  
  p>
```

```
  <table class="calendar">
```

```
    <caption>Calendrier - Mai 2025</caption>
```

```
    <thead>
```

```
      <tr>
```

```
        <th>Lun</th>
```

```
        <th>Mar</th>
```

```
        <th>Mer</th>
```

```
        <th>Jeu</th>
```

```
        <th>Ven</th>
```

```
        <th>Sam</th>
```

```
        <th>Dim</th>
```

```
      </tr>
```

```
    </thead>
```

```
    <tbody>
```

```
      <tr>
```

```
        <td></td>
```

```
        <td></td>
```

```
        <td></td>
```

```
        <td>1</td>
```

```
        <td>2</td>
```

```
        <td>3</td>
```

```
        <td>4</td>
```

```
      </tr>
```

```
      <tr>
```

```
        <td>5</td>
```

```
        <td>6</td>
```

```
        <td>7</td>
```

```
        <td>8</td>
```

```
        <td>9</td>
```

```
        <td>10</td>
```

```
        <td>11</td>
```

```
      </tr>
```

```
      <tr>
```

```
        <td>12</td>
```

```
        <td>13</td>
```

```
        <td>14</td>
```

```
        <td>15</td>
```

```
        <td>16</td>
```

```
        <td>17</td>
```

```
        <td>18</td>
```

```
      </tr>
```

```
      <tr>
```

```
        <td>19</td>
```

```
        <td>20</td>
```

```

        <td>21</td>
        <td>22</td>
        <td>23</td>
        <td>24</td>
        <td>25</td>
    </tr>
    <tr>
        <td>26</td>
        <td>27</td>
        <td>28</td>
        <td>29</td>
        <td>30</td>
        <td>31</td>
        <td></td>
    </tr>
</tbody>
</table>

```

<h2>Emploi du temps</h2>

```

<table class="schedule">
  <caption>Emploi du temps hebdomadaire</caption>
  <thead>
    <tr>
      <th>Horaire</th>
      <th>Lundi</th>
      <th>Mardi</th>
      <th>Mercredi</th>
      <th>Jeudi</th>
      <th>Vendredi</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>09:00 - 10:30</td>
      <td class="course">HTML Bases</td>
      <td class="course">CSS Avancé</td>
      <td class="course">JavaScript</td>
      <td class="course">Projet Web</td>
      <td class="course">Révisions</td>
    </tr>
    <tr>
      <td>10:30 - 11:00</td>
      <td class="break" colspan="5">Pause</td>
    </tr>
    <tr>
      <td>11:00 - 12:30</td>
      <td class="course">CSS Bases</td>
      <td class="course" rowspan="2">Atelier Design</
td>
      <td class="course">HTML Avancé</td>
      <td class="course" rowspan="4">Projet

```

```

Personnel</td>
    <td class="course">Évaluation</td>
</tr>
<tr>
    <td>12:30 - 14:00</td>
    <td class="break">Déjeuner</td>
    <td class="break">Déjeuner</td>
    <td class="break">Déjeuner</td>
</tr>
<tr>
    <td>14:00 - 15:30</td>
    <td class="course" colspan="2">Workshop
Responsive Design</td>
    <td class="course" rowspan="2">Projet
Collaboratif</td>
</tr>
<tr>
    <td>15:30 - 17:00</td>
    <td class="course">Tutorat</td>
    <td class="course">Conférence</td>
</tr>
</tbody>
</table>

```

<h2>Tableau de données</h2>

```

<table class="data-table">
  <caption>Population des plus grandes villes françaises</
caption>
  <thead>
    <tr>
      <th scope="col">Rang</th>
      <th scope="col">Ville</th>
      <th scope="col">Population (2023)</th>
      <th scope="col">Superficie (km²)</th>
      <th scope="col">Densité (hab/km²)</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Paris</td>
      <td>2 165 423</td>
      <td>105,4</td>
      <td>20 545</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Marseille</td>
      <td>870 731</td>
      <td>240,6</td>
      <td>3 619</td>
    </tr>
  </tbody>
</table>

```

```

        </tr>
        <tr>
            <td>3</td>
            <td>Lyon</td>
            <td>522 969</td>
            <td>47,9</td>
            <td>10 918</td>
        </tr>
        <tr>
            <td>4</td>
            <td>Toulouse</td>
            <td>493 465</td>
            <td>118,3</td>
            <td>4 171</td>
        </tr>
        <tr>
            <td>5</td>
            <td>Nice</td>
            <td>340 017</td>
            <td>71,9</td>
            <td>4 729</td>
        </tr>
    </tbody>
    <tfoot>
        <tr>
            <td colspan="2">Total</td>
            <td>4 392 605</td>
            <td>584,1</td>
            <td>7 520</td>
        </tr>
    </tfoot>
</table>
</body>
</html>

```

## Points importants à retenir

Les tableaux HTML sont des outils puissants pour présenter des données structurées. Voici les éléments essentiels à comprendre :

La structure de base d'un tableau comprend les balises `<table>`, `<tr>` (table row) pour les lignes, `<th>` (table header) pour les cellules d'en-tête et `<td>` (table data) pour les cellules de données. Cette structure permet d'organiser l'information de manière claire et lisible.

Pour améliorer la sémantique et l'accessibilité, il est recommandé d'utiliser les balises `<thead>`, `<tbody>` et `<tfoot>` qui permettent de distinguer les différentes parties

du tableau. La balise `<caption>` permet quant à elle d'ajouter un titre descriptif au tableau.

La fusion de cellules est possible grâce aux attributs `colspan` (fusion horizontale) et `rowspan` (fusion verticale). Ces attributs indiquent combien de colonnes ou de lignes la cellule doit occuper. Cette fonctionnalité est particulièrement utile pour créer des emplois du temps ou des plannings.

La stylisation des tableaux avec CSS permet d'améliorer considérablement leur lisibilité. Les propriétés comme `border-collapse`, `border`, `padding`, `background-color` et `text-align` sont couramment utilisées. L'alternance des couleurs de fond pour les lignes (`tr:nth-child(even/odd)`) est une technique efficace pour faciliter la lecture des données.

Il est important de noter que les tableaux doivent être utilisés uniquement pour présenter des données tabulaires et non pour la mise en page générale d'un site web. Pour la mise en page, il est préférable d'utiliser des techniques modernes comme Flexbox ou Grid.

## TP 5 : Introduction au CSS

### Objectif

Ce travail pratique vous permettra de découvrir les bases du CSS (Cascading Style Sheets), le langage qui permet de styliser les pages web. Vous apprendrez à lier une feuille de style externe à votre document HTML, à utiliser différents types de sélecteurs et à modifier l'apparence visuelle de votre page.

### Énoncé

#### Partie 1 : Liaison d'une feuille de style externe

1. Créez un nouveau fichier HTML nommé `style.html`
2. Ajoutez la structure de base d'une page HTML5
3. Donnez comme titre à votre page "Introduction au CSS"
4. Dans le corps de la page, créez un titre principal "Styliser avec CSS"
5. Ajoutez plusieurs sections avec des titres, paragraphes, listes et autres éléments HTML
6. Créez un fichier CSS externe nommé `styles.css`

7. Liez ce fichier CSS à votre document HTML en utilisant la balise `<link>`

## Partie 2 : Sélecteurs CSS de base

1. Dans votre fichier CSS, utilisez les types de sélecteurs suivants :
2. Sélecteur d'élément (ex: `p`, `h1`, `div`)
3. Sélecteur de classe (ex: `.ma-classe`)
4. Sélecteur d'identifiant (ex: `#mon-id`)
5. Sélecteur descendant (ex: `article p`)
6. Sélecteur d'enfant direct (ex: `nav > ul`)
7. Pseudo-classes (ex: `:hover`, `:first-child`)
8. Appliquez ces sélecteurs à différents éléments de votre page HTML

## Partie 3 : Modification des couleurs et des polices

1. Modifiez les couleurs de texte et d'arrière-plan de différents éléments
2. Changez les polices de caractères en utilisant `font-family`
3. Modifiez la taille des textes avec `font-size`
4. Ajoutez des effets de texte comme `font-weight`, `font-style` et `text-decoration`
5. Utilisez des dégradés de couleur pour l'arrière-plan d'un élément

## Partie 4 : Gestion des marges et des rembourrages

1. Appliquez des marges externes avec la propriété `margin`
2. Ajoutez des rembourrages internes avec la propriété `padding`
3. Utilisez les versions spécifiques (`margin-top`, `padding-left`, etc.)
4. Ajoutez des bordures à certains éléments avec la propriété `border`

## Conseils

- Utilisez les outils de développement de votre navigateur pour inspecter et modifier les styles en temps réel
- Commencez par des styles simples avant de passer à des effets plus complexes
- Organisez votre CSS de manière logique, en regroupant les styles par type d'élément ou par section
- N'hésitez pas à commenter votre CSS pour expliquer vos choix de style

# Solution

Fichier HTML ( style.html ) :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Introduction au CSS</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <h1>Styliser avec CSS</h1>
    <nav>
      <ul>
        <li><a href="#introduction">Introduction</a></li>
        <li><a href="#selecteurs">Sélecteurs</a></li>
        <li><a href="#couleurs">Couleurs et polices</a></li>
        <li><a href="#boites">Modèle de boîte</a></li>
      </ul>
    </nav>
  </header>

  <main>
    <section id="introduction" class="section-claire">
      <h2>Introduction au CSS</h2>
      <p class="important">CSS (Cascading Style Sheets)
est un langage qui décrit la présentation des documents HTML. Il
permet de contrôler l'apparence des éléments sur une page web.</p>
      <p>Avec CSS, vous pouvez définir les couleurs, les
polices, la mise en page et bien d'autres aspects visuels de
votre site.</p>
    </section>

    <section id="selecteurs" class="section-foncee">
      <h2>Les sélecteurs CSS</h2>
      <p>Les sélecteurs permettent de cibler des éléments
HTML spécifiques pour leur appliquer des styles.</p>
      <div class="exemple">
        <h3>Types de sélecteurs</h3>
        <ul>
          <li>Sélecteur d'élément: <code>p { color:
blue; }</code></li>
          <li>Sélecteur de classe: <code>.important
{ font-weight: bold; }</code></li>
          <li>Sélecteur d'ID: <code>#introduction
```

```

{ background-color: #f0f0f0; }</code></li>
    <li>Sélecteur descendant: <code>section p
{ line-height: 1.5; }</code></li>
</ul>
</div>
</section>

<section id="couleurs" class="section-claire">
  <h2>Couleurs et polices</h2>
  <p>Le CSS offre de nombreuses façons de spécifier
les couleurs:</p>
  <div class="exemple">
    <p class="couleur-nommee">Cette couleur est
définie par son nom.</p>
    <p class="couleur-hex">Cette couleur est définie
en hexadécimal.</p>
    <p class="couleur-rgb">Cette couleur est définie
en RGB.</p>
    <p class="couleur-hsl">Cette couleur est définie
en HSL.</p>
  </div>

  <h3>Polices de caractères</h3>
  <div class="exemple">
    <p class="police-serif">Ceci est une police avec
empattements (serif).</p>
    <p class="police-sans-serif">Ceci est une police
sans empattements (sans-serif).</p>
    <p class="police-monospace">Ceci est une police
à chasse fixe (monospace).</p>
  </div>
</section>

<section id="boites" class="section-foncee">
  <h2>Le modèle de boîte</h2>
  <p>Chaque élément HTML est représenté comme une
boîte rectangulaire.</p>
  <div class="boite">
    <p>Cette boîte a une bordure, une marge externe
et un rembourrage interne.</p>
  </div>
  <div class="boite-sans-marge">
    <p>Cette boîte n'a pas de marge externe mais a
un rembourrage interne.</p>
  </div>
</section>
</main>

<footer>
  <p>&copy; 2025 - Cours d'introduction au HTML et CSS</p>
</footer>

```

```
</body>
</html>
```

Fichier CSS ( styles.css ):

```
/* Styles généraux */
body {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  line-height: 1.6;
  color: #333;
  margin: 0;
  padding: 0;
  background-color: #f8f8f8;
}

/* En-tête et navigation */
header {
  background-color: #2c3e50;
  color: white;
  padding: 1rem;
  text-align: center;
}

h1 {
  margin-top: 0;
  font-size: 2.5rem;
}

nav ul {
  list-style-type: none;
  padding: 0;
  display: flex;
  justify-content: center;
  gap: 1.5rem;
}

nav > ul > li {
  margin: 0;
}

nav a {
  color: white;
  text-decoration: none;
  padding: 0.5rem 1rem;
  border-radius: 4px;
  transition: background-color 0.3s;
}

nav a:hover {
```

```
    background-color: #1abc9c;
}

/* Sections */
main {
    max-width: 1200px;
    margin: 0 auto;
    padding: 2rem;
}

section {
    margin-bottom: 2rem;
    padding: 1.5rem;
    border-radius: 8px;
}

.section-claire {
    background-color: white;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}

.section-foncee {
    background-color: #ecf0f1;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}

h2 {
    color: #2c3e50;
    border-bottom: 2px solid #1abc9c;
    padding-bottom: 0.5rem;
}

h3 {
    color: #3498db;
}

/* Classes spécifiques */
.important {
    font-weight: bold;
    font-size: 1.1rem;
    color: #e74c3c;
}

.exemple {
    background-color: #f9f9f9;
    border-left: 4px solid #3498db;
    padding: 1rem;
    margin: 1rem 0;
}

code {
    background-color: #f0f0f0;
```

```
padding: 0.2rem 0.4rem;
border-radius: 3px;
font-family: 'Courier New', Courier, monospace;
}

/* Couleurs */
.couleur-nomme {
  color: red;
}

.couleur-hex {
  color: #00aa55;
}

.couleur-rgb {
  color: rgb(64, 99, 255);
}

.couleur-hsl {
  color: hsl(330, 75%, 60%);
}

/* Polices */
.police-serif {
  font-family: Georgia, 'Times New Roman', Times, serif;
  font-size: 1.1rem;
}

.police-sans-serif {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 1.1rem;
}

.police-monospace {
  font-family: 'Courier New', Courier, monospace;
  font-size: 1.1rem;
}

/* Modèle de boîte */
.boite {
  background-color: #d5f5e3;
  border: 3px solid #1abc9c;
  padding: 20px;
  margin: 30px;
}

.boite-sans-marge {
  background-color: #d5e8f5;
  border: 3px dashed #3498db;
  padding: 20px;
  margin: 0;
}
```

```
/* Pied de page */
footer {
  background-color: #2c3e50;
  color: white;
  text-align: center;
  padding: 1rem;
  margin-top: 2rem;
}
```

## Points importants à retenir

Le CSS (Cascading Style Sheets) est un langage fondamental pour contrôler l'apparence visuelle des pages web. Voici les concepts clés à comprendre :

La liaison d'une feuille de style externe à un document HTML se fait via la balise `<link>` placée dans l'en-tête ( `<head>` ) du document. Cette approche permet de séparer le contenu (HTML) de la présentation (CSS), ce qui facilite la maintenance et permet de réutiliser les styles sur plusieurs pages.

Les sélecteurs CSS sont des motifs qui déterminent quels éléments HTML seront affectés par les règles de style. Il existe plusieurs types de sélecteurs, chacun ayant sa propre spécificité : - Les sélecteurs d'éléments ciblent tous les éléments d'un type donné (ex: `p` , `h1` ) - Les sélecteurs de classe ( `.ma-classe` ) ciblent les éléments ayant l'attribut `class` correspondant - Les sélecteurs d'ID ( `#mon-id` ) ciblent l'élément unique ayant l'attribut `id` correspondant - Les sélecteurs combinés permettent de cibler des éléments selon leur position dans la structure du document

Les propriétés de couleur et de police permettent de contrôler l'apparence du texte et des arrière-plans. Les couleurs peuvent être spécifiées de différentes manières (noms, codes hexadécimaux, RGB, HSL), tandis que les polices peuvent être définies par famille, taille, style et poids.

Le modèle de boîte CSS est un concept fondamental qui considère chaque élément HTML comme une boîte rectangulaire composée de : - Le contenu (content) : l'espace où le texte et les images apparaissent - Le rembourrage (padding) : l'espace transparent autour du contenu - La bordure (border) : la ligne qui entoure le padding - La marge (margin) : l'espace transparent autour de la bordure

La compréhension de ces concepts de base vous permettra de créer des pages web visuellement attrayantes et professionnelles. N'oubliez pas que la pratique est essentielle pour maîtriser le CSS, alors n'hésitez pas à expérimenter avec différentes propriétés et valeurs.

# TP 6 : Mise en page avec CSS

## Objectif

Ce travail pratique vous permettra d'approfondir vos connaissances en CSS en vous concentrant sur les techniques de mise en page. Vous découvrirez le modèle de boîte CSS, les différentes méthodes de positionnement et l'utilisation de Flexbox pour créer des layouts modernes et responsifs.

## Énoncé

### Partie 1 : Le modèle de boîte CSS

1. Créez un nouveau fichier HTML nommé `layout.html`
2. Ajoutez la structure de base d'une page HTML5
3. Donnez comme titre à votre page "Mise en page avec CSS"
4. Créez un fichier CSS externe nommé `layout.css` et liez-le à votre document HTML
5. Dans votre HTML, créez plusieurs `<div>` avec des classes différentes
6. Dans votre CSS, appliquez à ces `<div>` différentes propriétés du modèle de boîte :
7. Largeur et hauteur (`width`, `height`)
8. Marges externes (`margin`)
9. Rembourrages internes (`padding`)
10. Bordures (`border`)
11. Expérimentez avec `box-sizing: border-box` et `box-sizing: content-box`

### Partie 2 : Positionnement des éléments

1. Créez une section "Techniques de positionnement" dans votre page
2. Expérimentez avec les différentes valeurs de la propriété `position` :
3. `static` (positionnement normal)
4. `relative` (positionnement relatif à sa position normale)
5. `absolute` (positionnement par rapport à l'ancêtre positionné le plus proche)
6. `fixed` (positionnement par rapport à la fenêtre du navigateur)
7. `sticky` (positionnement hybride entre relative et fixed)
8. Utilisez les propriétés `top`, `right`, `bottom` et `left` pour positionner vos éléments
9. Créez un exemple de menu de navigation fixe en haut de la page

10. Créez un bouton "retour en haut" fixé en bas à droite de la page

## Partie 3 : Création d'un layout simple

1. Créez une section "Layout de page" dans votre document
2. Réalisez une mise en page classique avec :
3. Un en-tête ( `<header>` )
4. Une barre de navigation ( `<nav>` )
5. Une section principale ( `<main>` ) contenant du contenu
6. Une barre latérale ( `<aside>` )
7. Un pied de page ( `<footer>` )
8. Utilisez les propriétés `float` et `clear` pour positionner ces éléments
9. Ajoutez des marges et des rembourrages appropriés pour espacer les éléments

## Partie 4 : Utilisation de Flexbox

1. Créez une section "Flexbox" dans votre document
2. Créez un conteneur flex avec plusieurs éléments enfants
3. Expérimentez avec les propriétés suivantes sur le conteneur :
4. `display: flex`
5. `flex-direction`
6. `justify-content`
7. `align-items`
8. `flex-wrap`
9. `gap`
10. Appliquez les propriétés suivantes sur les éléments enfants :
11. `flex-grow`
12. `flex-shrink`
13. `flex-basis`
14. `align-self`
15. Créez une galerie d'images responsive avec Flexbox

## Conseils

- Utilisez les outils de développement de votre navigateur pour visualiser le modèle de boîte
- Commencez par des mises en page simples avant de passer à des structures plus complexes
- N'hésitez pas à utiliser des couleurs d'arrière-plan différentes pour mieux visualiser vos éléments

- Testez votre mise en page en redimensionnant la fenêtre du navigateur

## Solution

Fichier HTML ( layout.html ) :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Mise en page avec CSS</title>
  <link rel="stylesheet" href="layout.css">
</head>
<body>
  <nav class="menu-fixe">
    <ul>
      <li><a href="#modele-boite">Modèle de boîte</a></li>
      <li><a href="#positionnement">Positionnement</a></
li>
      <li><a href="#layout">Layout de page</a></li>
      <li><a href="#flexbox">Flexbox</a></li>
    </ul>
  </nav>

  <header class="page-header">
    <h1>Mise en page avec CSS</h1>
    <p>Découvrez les techniques modernes de mise en page
web</p>
  </header>

  <section id="modele-boite" class="section">
    <h2>Le modèle de boîte CSS</h2>
    <p>Chaque élément HTML est représenté comme une boîte
rectangulaire.</p>

    <div class="demo-container">
      <div class="box content-box">
        <p>box-sizing: content-box</p>
      </div>

      <div class="box border-box">
        <p>box-sizing: border-box</p>
      </div>
    </div>

    <div class="info-box">
      <p>Le modèle de boîte standard (content-box) calcule
```

la taille d'un élément en ajoutant le padding et la bordure à la largeur/hauteur spécifiée.</p>

<p>Le modèle alternatif (border-box) inclut le padding et la bordure dans la largeur/hauteur spécifiée.</p>

</div>

</section>

<section id="positionnement" class="section">

<h2>Techniques de positionnement</h2>

<div class="demo-container position-demo">

<div class="pos-box static">

<p>position: static</p>

</div>

<div class="pos-box relative">

<p>position: relative</p>

</div>

<div class="pos-container">

<div class="pos-box absolute">

<p>position: absolute</p>

</div>

<p class="note">Cet élément est positionné par rapport à son parent</p>

</div>

<div class="pos-box sticky">

<p>position: sticky</p>

</div>

</div>

<div class="info-box">

<p>La propriété <code>position</code> détermine comment un élément est positionné dans le document.</p>

<p>Les valeurs les plus courantes sont <code>static</code> (par défaut), <code>relative</code>, <code>absolute</code>, <code>fixed</code> et <code>sticky</code>.</p>

</div>

</section>

<section id="layout" class="section">

<h2>Layout de page</h2>

<div class="page-layout">

<header class="layout-header">

<h3>En-tête de page</h3>

</header>

<nav class="layout-nav">

<h3>Navigation</h3>

```

        <li><a href="#">Accueil</a></li>
        <li><a href="#">À propos</a></li>
        <li><a href="#">Services</a></li>
        <li><a href="#">Contact</a></li>
    </ul>
</nav>

<main class="layout-main">
    <h3>Contenu principal</h3>
    <p>Cette section contient le contenu principal
de la page. Dans un site réel, c'est ici que vous placeriez vos
articles, produits ou autres informations importantes.</p>
    <p>Le contenu principal occupe généralement la
plus grande partie de la page.</p>
</main>

<aside class="layout-sidebar">
    <h3>Barre latérale</h3>
    <p>Cette section contient des informations
complémentaires, comme des liens connexes, des publicités ou des
widgets.</p>
</aside>

<footer class="layout-footer">
    <h3>Pied de page</h3>
    <p>© 2025 - Tous droits réservés</p>
</footer>
</div>
</section>

<section id="flexbox" class="section">
    <h2>Utilisation de Flexbox</h2>

    <div class="flex-demo">
        <h3>Propriétés du conteneur flex</h3>

        <div class="flex-container direction-row">
            <div class="flex-item">1</div>
            <div class="flex-item">2</div>
            <div class="flex-item">3</div>
            <div class="flex-item">4</div>
        </div>
        <p class="code">flex-direction: row;</p>

        <div class="flex-container direction-column">
            <div class="flex-item">1</div>
            <div class="flex-item">2</div>
            <div class="flex-item">3</div>
            <div class="flex-item">4</div>
        </div>
        <p class="code">flex-direction: column;</p>
    </div>

```

```

<div class="flex-container justify-between">
  <div class="flex-item">1</div>
  <div class="flex-item">2</div>
  <div class="flex-item">3</div>
  <div class="flex-item">4</div>
</div>
<p class="code">justify-content: space-between;</p>

<div class="flex-container align-center">
  <div class="flex-item">1</div>
  <div class="flex-item tall">2</div>
  <div class="flex-item">3</div>
  <div class="flex-item">4</div>
</div>
<p class="code">align-items: center;</p>
</div>

<h3>Galerie d'images avec Flexbox</h3>
<div class="flex-gallery">
  <div class="gallery-item">
    
    <p>Image 1</p>
  </div>
  <div class="gallery-item">
    
    <p>Image 2</p>
  </div>
  <div class="gallery-item">
    
    <p>Image 3</p>
  </div>
  <div class="gallery-item">
    
    <p>Image 4</p>
  </div>
  <div class="gallery-item">
    
    <p>Image 5</p>
  </div>
  <div class="gallery-item">
    
    <p>Image 6</p>
  </div>
</div>
</section>

```

```
<a href="#" class="back-to-top">↑</a>

<footer class="page-footer">
  <p>© 2025 - Cours d'introduction au HTML et CSS</p>
</footer>
</body>
</html>
```

Fichier CSS ( layout.css ):

```
/* Styles généraux */
* {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
}

body {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  line-height: 1.6;
  color: #333;
  padding-top: 60px; /* Espace pour le menu fixe */
}

.section {
  padding: 2rem;
  margin-bottom: 2rem;
  background-color: #f9f9f9;
  border-radius: 8px;
  max-width: 1200px;
  margin-left: auto;
  margin-right: auto;
}

h1, h2, h3 {
  margin-bottom: 1rem;
  color: #2c3e50;
}

p {
  margin-bottom: 1rem;
}

code {
  background-color: #f0f0f0;
  padding: 0.2rem 0.4rem;
  border-radius: 3px;
  font-family: 'Courier New', Courier, monospace;
}
```

```

}

.info-box {
  background-color: #e8f4f8;
  border-left: 4px solid #3498db;
  padding: 1rem;
  margin: 1rem 0;
}

.demo-container {
  margin: 2rem 0;
  overflow: hidden;
}

/* Menu fixe */
.menu-fixe {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  background-color: #2c3e50;
  color: white;
  padding: 1rem;
  z-index: 1000;
}

.menu-fixe ul {
  display: flex;
  justify-content: center;
  list-style-type: none;
}

.menu-fixe li {
  margin: 0 1rem;
}

.menu-fixe a {
  color: white;
  text-decoration: none;
  padding: 0.5rem;
  border-radius: 4px;
  transition: background-color 0.3s;
}

.menu-fixe a:hover {
  background-color: #1abc9c;
}

/* En-tête de page */
.page-header {
  text-align: center;
  padding: 3rem 1rem;
}

```

```
background-color: #3498db;
color: white;
margin-bottom: 2rem;
}

.page-header h1 {
font-size: 2.5rem;
color: white;
}

/* Modèle de boîte */
.box {
width: 300px;
height: 150px;
padding: 20px;
border: 10px solid #3498db;
margin: 20px;
background-color: #e8f4f8;
float: left;
}

.content-box {
box-sizing: content-box;
background-color: #d5f5e3;
border-color: #2ecc71;
}

.border-box {
box-sizing: border-box;
background-color: #fadbd8;
border-color: #e74c3c;
}

/* Positionnement */
.position-demo {
position: relative;
height: 400px;
background-color: #f0f0f0;
padding: 20px;
margin-bottom: 2rem;
}

.pos-box {
width: 200px;
height: 100px;
padding: 15px;
margin: 10px;
background-color: #d5f5e3;
border: 2px solid #2ecc71;
}

.static {
```

```
/* position: static est la valeur par défaut */
background-color: #d5f5e3;
border-color: #2ecc71;
}

.relative {
position: relative;
top: 20px;
left: 50px;
background-color: #fadbd8;
border-color: #e74c3c;
}

.pos-container {
position: relative;
height: 150px;
width: 80%;
background-color: #ebdef0;
margin: 20px;
padding: 10px;
}

.absolute {
position: absolute;
top: 40px;
right: 30px;
background-color: #d6eaf8;
border-color: #3498db;
}

.sticky {
position: sticky;
top: 70px;
background-color: #fef9e7;
border-color: #f1c40f;
}

.note {
margin-top: 120px;
font-style: italic;
}

/* Layout de page */
.page-layout {
background-color: #f5f5f5;
padding: 20px;
margin-bottom: 2rem;
overflow: hidden; /* Pour contenir les éléments flottants */
}

.layout-header {
background-color: #3498db;
```

```
    color: white;
    padding: 20px;
    margin-bottom: 20px;
    text-align: center;
}

.layout-header h3 {
    color: white;
}

.layout-nav {
    background-color: #2c3e50;
    color: white;
    padding: 20px;
    margin-bottom: 20px;
}

.layout-nav h3 {
    color: white;
}

.layout-nav ul {
    list-style-type: none;
}

.layout-nav a {
    color: white;
    text-decoration: none;
    display: block;
    padding: 5px 0;
}

.layout-nav a:hover {
    color: #1abc9c;
}

.layout-main {
    background-color: white;
    padding: 20px;
    margin-bottom: 20px;
    float: left;
    width: 70%;
}

.layout-sidebar {
    background-color: #ecf0f1;
    padding: 20px;
    margin-bottom: 20px;
    float: right;
    width: 28%;
}
```

```
.layout-footer {
  background-color: #2c3e50;
  color: white;
  padding: 20px;
  clear: both; /* Pour s'assurer qu'il apparaît sous les
éléments flottants */
  text-align: center;
}

.layout-footer h3 {
  color: white;
}

/* Flexbox */
.flex-demo {
  margin-bottom: 2rem;
}

.flex-container {
  display: flex;
  margin-bottom: 1rem;
  background-color: #f0f0f0;
  padding: 10px;
}

.flex-item {
  background-color: #3498db;
  color: white;
  padding: 20px;
  margin: 5px;
  text-align: center;
  flex: 1;
}

.direction-row {
  flex-direction: row;
}

.direction-column {
  flex-direction: column;
  height: 300px;
}

.justify-between {
  justify-content: space-between;
}

.align-center {
  align-items: center;
  height: 150px;
}
```

```
.tall {
  height: 100px;
}

.code {
  font-family: 'Courier New', Courier, monospace;
  background-color: #f8f8f8;
  padding: 5px;
  border-left: 3px solid #3498db;
  margin-bottom: 20px;
}

/* Galerie d'images avec Flexbox */
.flex-gallery {
  display: flex;
  flex-wrap: wrap;
  gap: 15px;
  justify-content: center;
}

.gallery-item {
  flex: 0 0 calc(33.333% - 15px);
  background-color: white;
  border-radius: 5px;
  overflow: hidden;
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
  transition: transform 0.3s;
}

.gallery-item:hover {
  transform: translateY(-5px);
}

.gallery-item img {
  width: 100%;
  height: auto;
  display: block;
}

.gallery-item p {
  padding: 10px;
  text-align: center;
  margin: 0;
}

/* Bouton retour en haut */
.back-to-top {
  position: fixed;
  bottom: 20px;
  right: 20px;
  background-color: #3498db;
  color: white;
}
```

```

width: 40px;
height: 40px;
border-radius: 50%;
display: flex;
justify-content: center;
align-items: center;
text-decoration: none;
font-size: 20px;
box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);
transition: background-color 0.3s;
}

.back-to-top:hover {
background-color: #2980b9;
}

/* Pied de page */
.page-footer {
background-color: #2c3e50;
color: white;
text-align: center;
padding: 1rem;
margin-top: 2rem;
}

/* Media queries pour la responsivité */
@media (max-width: 768px) {
.layout-main, .layout-sidebar {
float: none;
width: 100%;
}

.gallery-item {
flex: 0 0 calc(50% - 15px);
}

.menu-fixe ul {
flex-direction: column;
align-items: center;
}

.menu-fixe li {
margin: 5px 0;
}

body {
padding-top: 150px;
}
}

@media (max-width: 480px) {
.gallery-item {

```

```
    flex: 0 0 100%;  
  }  
}
```

## Points importants à retenir

La mise en page CSS est un aspect fondamental du développement web moderne. Voici les concepts clés à comprendre :

Le modèle de boîte CSS est le principe selon lequel chaque élément HTML est représenté comme une boîte rectangulaire composée du contenu, du padding, de la bordure et de la marge. La propriété `box-sizing` permet de contrôler comment la largeur et la hauteur d'un élément sont calculées : - `content-box` (par défaut) : la largeur et la hauteur spécifiées s'appliquent uniquement au contenu - `border-box` : la largeur et la hauteur spécifiées incluent le contenu, le padding et la bordure

Le positionnement CSS permet de contrôler l'emplacement des éléments dans la page. Les valeurs principales de la propriété `position` sont : - `static` : positionnement normal dans le flux du document (par défaut) - `relative` : positionné par rapport à sa position normale - `absolute` : positionné par rapport à l'ancêtre positionné le plus proche - `fixed` : positionné par rapport à la fenêtre du navigateur - `sticky` : se comporte comme `relative` jusqu'à ce qu'un seuil soit atteint, puis comme `fixed`

Les techniques de mise en page traditionnelles utilisent souvent les propriétés `float` et `clear` pour positionner les éléments côte à côte. Bien que cette approche soit encore utilisée, elle présente des limitations et est progressivement remplacée par des méthodes plus modernes comme Flexbox et Grid.

Flexbox est un modèle de mise en page unidimensionnel conçu pour organiser les éléments en lignes ou en colonnes. Il offre un contrôle puissant sur l'alignement, la distribution de l'espace et l'ordre des éléments. Les principales propriétés du conteneur flex sont : - `display: flex` : active le modèle Flexbox - `flex-direction` : définit l'axe principal (row, column, row-reverse, column-reverse) - `justify-content` : aligne les éléments le long de l'axe principal - `align-items` : aligne les éléments le long de l'axe secondaire - `flex-wrap` : contrôle si les éléments peuvent passer à la ligne

Les propriétés des éléments enfants flex incluent : - `flex-grow` : facteur de croissance - `flex-shrink` : facteur de rétrécissement - `flex-basis` : taille initiale - `flex` : raccourci pour les trois propriétés ci-dessus - `align-self` : remplace l'alignement défini par `align-items`

La mise en page responsive est essentielle pour s'adapter aux différentes tailles d'écran. Les media queries permettent d'appliquer des styles spécifiques en fonction des caractéristiques de l'appareil, comme la largeur de l'écran. Cette approche est fondamentale pour créer des sites web qui fonctionnent bien sur tous les appareils, des téléphones mobiles aux grands écrans.

## TP 7 : Formulaires HTML

### Objectif

Ce travail pratique vous permettra de maîtriser la création et la stylisation de formulaires en HTML. Vous apprendrez à utiliser différents types de champs, à structurer un formulaire de manière logique, à implémenter une validation de base et à améliorer l'apparence des formulaires avec CSS.

### Énoncé

#### Partie 1 : Création d'un formulaire de contact

1. Créez un nouveau fichier HTML nommé `formulaires.html`
2. Ajoutez la structure de base d'une page HTML5
3. Donnez comme titre à votre page "Formulaires HTML"
4. Créez un fichier CSS externe nommé `formulaires.css` et liez-le à votre document HTML
5. Dans le corps de la page, créez un titre principal "Création de formulaires en HTML"
6. Créez un formulaire de contact avec les éléments suivants :
7. Champ pour le nom (texte)
8. Champ pour le prénom (texte)
9. Champ pour l'email (email)
10. Champ pour le numéro de téléphone (tel)
11. Zone de texte pour le message
12. Bouton d'envoi

#### Partie 2 : Différents types de champs

1. Créez une section "Types de champs" avec un titre de niveau 2
2. Créez un formulaire présentant les types de champs suivants :
3. Champ de texte simple ( `text` )

4. Champ de mot de passe ( password )
5. Case à cocher ( checkbox )
6. Boutons radio ( radio )
7. Liste déroulante ( select )
8. Champ de date ( date )
9. Champ numérique ( number )
10. Curseur ( range )
11. Sélecteur de couleur ( color )
12. Champ de recherche ( search )
13. Champ d'URL ( url )
14. Téléchargement de fichier ( file )
15. Ajoutez des attributs appropriés à chaque champ (placeholder, min/max, required, etc.)

### Partie 3 : Validation de formulaire

1. Créez une section "Validation de formulaire" avec un titre de niveau 2
2. Créez un formulaire d'inscription avec les champs suivants :
3. Nom d'utilisateur (obligatoire, minimum 3 caractères)
4. Email (obligatoire, format email)
5. Mot de passe (obligatoire, minimum 8 caractères)
6. Confirmation de mot de passe
7. Acceptation des conditions d'utilisation (obligatoire)
8. Utilisez les attributs de validation HTML5 ( required , minlength , pattern , etc.)
9. Ajoutez des messages d'erreur personnalisés avec l'attribut title

### Partie 4 : Stylisation des formulaires avec CSS

1. Dans votre fichier CSS, créez des styles pour améliorer l'apparence de vos formulaires :
2. Ajoutez des marges et des rembourrages appropriés
3. Stylisez les étiquettes ( label )
4. Personnalisez l'apparence des champs de saisie
5. Créez un style distinct pour les champs en focus
6. Stylisez les boutons d'envoi
7. Ajoutez des effets de transition
8. Créez un style pour les messages d'erreur

# Conseils

- Utilisez toujours l'élément `<label>` associé à chaque champ pour améliorer l'accessibilité
- Regroupez les champs connexes avec `<fieldset>` et `<legend>`
- Testez vos formulaires dans différents navigateurs
- Pensez à l'expérience utilisateur : les formulaires doivent être clairs et faciles à remplir
- N'oubliez pas d'ajouter l'attribut `name` à tous les champs qui doivent envoyer des données

# Solution

Fichier HTML ( `formulaires.html` ) :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Formulaires HTML</title>
  <link rel="stylesheet" href="formulaires.css">
</head>
<body>
  <header>
    <h1>Création de formulaires en HTML</h1>
    <p>Découvrez comment créer et styliser différents types
de formulaires</p>
  </header>

  <main>
    <section id="contact">
      <h2>Formulaire de contact</h2>
      <form action="#" method="post" class="form-contact">
        <div class="form-group">
          <label for="nom">Nom :</label>
          <input type="text" id="nom" name="nom"
placeholder="Votre nom" required>
        </div>

        <div class="form-group">
          <label for="prenom">Prénom :</label>
          <input type="text" id="prenom" name="prenom"
placeholder="Votre prénom" required>
        </div>
      </form>
    </section>
  </main>
</body>
</html>
```

```

        <div class="form-group">
            <label for="email">Email :</label>
            <input type="email" id="email" name="email"
placeholder="votre.email@exemple.com" required>
        </div>

        <div class="form-group">
            <label for="telephone">Téléphone :</label>
            <input type="tel" id="telephone"
name="telephone" placeholder="01 23 45 67 89" pattern="[0-9]{2}
[\s][0-9]{2}[\s][0-9]{2}[\s][0-9]{2}[\s][0-9]{2}">
            <small>Format: 01 23 45 67 89</small>
        </div>

        <div class="form-group">
            <label for="message">Message :</label>
            <textarea id="message" name="message"
rows="5" placeholder="Votre message..." required></textarea>
        </div>

        <div class="form-group">
            <button type="submit" class="btn-
submit">Envoyer</button>
        </div>
    </form>
</section>

<section id="types-champs">
    <h2>Types de champs</h2>
    <form action="#" method="post" class="form-demo">
        <fieldset>
            <legend>Informations personnelles</legend>

            <div class="form-group">
                <label for="texte">Texte simple :</
label>
                <input type="text" id="texte"
name="texte" placeholder="Entrez du texte">
            </div>

            <div class="form-group">
                <label for="password">Mot de passe :</
label>
                <input type="password" id="password"
name="password" placeholder="Votre mot de passe">
            </div>

            <div class="form-group checkbox-group">
                <label>Centres d'intérêt :</label>
                <div class="checkbox-container">
                    <input type="checkbox" id="sport"
name="interets" value="sport">

```

```

        <label for="sport">Sport</label>
    </div>
    <div class="checkbox-container">
        <input type="checkbox" id="musique"
name="interets" value="musique">
        <label for="musique">Musique</label>
    </div>
    <div class="checkbox-container">
        <input type="checkbox" id="lecture"
name="interets" value="lecture">
        <label for="lecture">Lecture</label>
    </div>
</div>

<div class="form-group radio-group">
    <label>Sexe :</label>
    <div class="radio-container">
        <input type="radio" id="homme"
name="sexe" value="homme">
        <label for="homme">Homme</label>
    </div>
    <div class="radio-container">
        <input type="radio" id="femme"
name="sexe" value="femme">
        <label for="femme">Femme</label>
    </div>
    <div class="radio-container">
        <input type="radio" id="autre"
name="sexe" value="autre">
        <label for="autre">Autre</label>
    </div>
</div>

<div class="form-group">
    <label for="pays">Pays :</label>
    <select id="pays" name="pays">
        <option value="">Sélectionnez un
pays</option>
        <option value="fr">France</option>
        <option value="be">Belgique</option>
        <option value="ch">Suisse</option>
        <option value="ca">Canada</option>
        <option value="autre">Autre</option>
    </select>
</div>
</fieldset>

<fieldset>
    <legend>Champs spéciaux</legend>

    <div class="form-group">
        <label for="date">Date :</label>

```

```

        <input type="date" id="date"
name="date">
    </div>

    <div class="form-group">
        <label for="nombre">Nombre :</label>
        <input type="number" id="nombre"
name="nombre" min="0" max="100" step="1" value="50">
    </div>

    <div class="form-group">
        <label for="curseur">Curseur :</label>
        <input type="range" id="curseur"
name="curseur" min="0" max="100" step="1" value="50">
        <output for="curseur">50</output>
    </div>

    <div class="form-group">
        <label for="couleur">Couleur :</label>
        <input type="color" id="couleur"
name="couleur" value="#3498db">
    </div>

    <div class="form-group">
        <label for="recherche">Recherche :</
label>
        <input type="search" id="recherche"
name="recherche" placeholder="Rechercher...">
    </div>

    <div class="form-group">
        <label for="url">URL :</label>
        <input type="url" id="url" name="url"
placeholder="https://exemple.com">
    </div>

    <div class="form-group">
        <label for="fichier">Fichier :</label>
        <input type="file" id="fichier"
name="fichier">
    </div>
</fieldset>

    <div class="form-group">
        <button type="submit" class="btn-
submit">Envoyer</button>
        <button type="reset" class="btn-
reset">Réinitialiser</button>
    </div>
</form>
</section>

```

```

<section id="validation">
  <h2>Validation de formulaire</h2>
  <form action="#" method="post" class="form-
validation">
    <div class="form-group">
      <label for="username">Nom d'utilisateur :</
label>
      <input type="text" id="username"
name="username" required minlength="3" maxlength="20"
pattern="[a-zA-Z0-9_]+"
title="Le nom d'utilisateur ne doit contenir que des lettres,
chiffres et underscores">
      <small>3 à 20 caractères, lettres, chiffres
et underscores uniquement</small>
    </div>

    <div class="form-group">
      <label for="email-inscription">Email :</
label>
      <input type="email" id="email-inscription"
name="email" required>
    </div>

    <div class="form-group">
      <label for="mdp">Mot de passe :</label>
      <input type="password" id="mdp" name="mdp"
required minlength="8"
pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-
Z]).{8,}"
title="Le mot de passe doit contenir
au moins 8 caractères, incluant une majuscule, une minuscule et
un chiffre">
      <small>Minimum 8 caractères, avec au moins
une majuscule, une minuscule et un chiffre</small>
    </div>

    <div class="form-group">
      <label for="mdp-confirm">Confirmation du mot
de passe :</label>
      <input type="password" id="mdp-confirm"
name="mdp-confirm" required>
    </div>

    <div class="form-group checkbox-group">
      <div class="checkbox-container">
        <input type="checkbox" id="conditions"
name="conditions" required>
        <label for="conditions">J'accepte les <a
href="#">conditions d'utilisation</a></label>
      </div>
    </div>
  </form>

```

```

        <div class="form-group">
            <button type="submit" class="btn-
submit">S'inscrire</button>
        </div>
    </form>
</section>
</main>

<footer>
    <p>© 2025 - Cours d'introduction au HTML et CSS</p>
</footer>

<script>
    // Script pour mettre à jour la valeur du curseur

document.getElementById('curseur').addEventListener('input',
function() {

document.querySelector('output[for="curseur"]').value =
this.value;
    });

    // Script pour vérifier que les mots de passe
correspondent
    document.querySelector('.form-
validation').addEventListener('submit', function(e) {
        const mdp = document.getElementById('mdp').value;
        const mdpConfirm = document.getElementById('mdp-
confirm').value;

        if (mdp !== mdpConfirm) {
            e.preventDefault();
            alert('Les mots de passe ne correspondent
pas. ');
        }
    });
</script>
</body>
</html>

```

Fichier CSS ( formulaires.css ):

```

/* Styles généraux */
* {
    box-sizing: border-box;
    margin: 0;
    padding: 0;
}

body {

```

```
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    line-height: 1.6;
    color: #333;
    background-color: #f8f8f8;
    padding: 20px;
}

header {
    text-align: center;
    margin-bottom: 2rem;
    padding: 2rem;
    background-color: #3498db;
    color: white;
    border-radius: 8px;
}

section {
    margin-bottom: 3rem;
    background-color: white;
    padding: 2rem;
    border-radius: 8px;
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
}

h1, h2 {
    margin-bottom: 1rem;
}

h2 {
    color: #2c3e50;
    border-bottom: 2px solid #3498db;
    padding-bottom: 0.5rem;
}

/* Styles des formulaires */
form {
    max-width: 800px;
    margin: 0 auto;
}

.form-group {
    margin-bottom: 1.5rem;
}

label {
    display: block;
    margin-bottom: 0.5rem;
    font-weight: bold;
    color: #2c3e50;
}
```

```
input, select, textarea {
  width: 100%;
  padding: 10px;
  border: 1px solid #ddd;
  border-radius: 4px;
  font-size: 1rem;
  transition: border-color 0.3s, box-shadow 0.3s;
}

input:focus, select:focus, textarea:focus {
  outline: none;
  border-color: #3498db;
  box-shadow: 0 0 0 3px rgba(52, 152, 219, 0.2);
}

textarea {
  resize: vertical;
  min-height: 100px;
}

small {
  display: block;
  margin-top: 0.25rem;
  color: #7f8c8d;
  font-size: 0.85rem;
}

/* Styles pour les boutons */
button {
  padding: 10px 20px;
  border: none;
  border-radius: 4px;
  font-size: 1rem;
  cursor: pointer;
  transition: background-color 0.3s;
}

.btn-submit {
  background-color: #2ecc71;
  color: white;
}

.btn-submit:hover {
  background-color: #27ae60;
}

.btn-reset {
  background-color: #e74c3c;
  color: white;
  margin-left: 10px;
}
```

```
.btn-reset:hover {
    background-color: #c0392b;
}

/* Styles pour les fieldsets */
fieldset {
    border: 1px solid #ddd;
    border-radius: 4px;
    padding: 1.5rem;
    margin-bottom: 1.5rem;
}

legend {
    padding: 0 10px;
    font-weight: bold;
    color: #2c3e50;
}

/* Styles pour les cases à cocher et boutons radio */
.checkbox-group, .radio-group {
    margin-bottom: 1rem;
}

.checkbox-container, .radio-container {
    display: flex;
    align-items: center;
    margin-bottom: 0.5rem;
}

.checkbox-container input, .radio-container input {
    width: auto;
    margin-right: 10px;
}

.checkbox-container label, .radio-container label {
    margin-bottom: 0;
    font-weight: normal;
}

/* Styles pour les champs spéciaux */
input[type="color"] {
    height: 40px;
    padding: 2px;
}

input[type="range"] {
    padding: 0;
}

output {
    margin-left: 10px;
}
```

```
/* Styles pour la validation */
input:invalid {
  border-color: #e74c3c;
}

input:invalid:focus {
  box-shadow: 0 0 0 3px rgba(231, 76, 60, 0.2);
}

/* Styles pour les liens */
a {
  color: #3498db;
  text-decoration: none;
}

a:hover {
  text-decoration: underline;
}

/* Styles pour le pied de page */
footer {
  text-align: center;
  margin-top: 2rem;
  padding: 1rem;
  color: #7f8c8d;
}

/* Media queries pour la responsivité */
@media (max-width: 768px) {
  section {
    padding: 1.5rem;
  }

  .form-group {
    margin-bottom: 1rem;
  }

  input, select, textarea {
    padding: 8px;
  }
}

@media (max-width: 480px) {
  button {
    width: 100%;
    margin-bottom: 0.5rem;
  }

  .btn-reset {
    margin-left: 0;
  }
}
```

```
}  
}
```

## Points importants à retenir

Les formulaires HTML sont essentiels pour collecter des informations auprès des utilisateurs. Voici les concepts clés à comprendre :

La structure de base d'un formulaire comprend l'élément `<form>` qui encapsule tous les champs, avec les attributs `action` (URL où les données seront envoyées) et `method` (méthode HTTP à utiliser, généralement GET ou POST). À l'intérieur du formulaire, on trouve des éléments `<input>`, `<select>`, `<textarea>` et `<button>` qui permettent aux utilisateurs d'interagir et de saisir des données.

L'élément `<input>` est polyvalent et peut créer différents types de champs selon l'attribut `type` : - `text` : champ de texte simple - `password` : champ pour les mots de passe (caractères masqués) - `email` : champ pour les adresses email - `tel` : champ pour les numéros de téléphone - `checkbox` : case à cocher - `radio` : bouton radio - `date` : sélecteur de date - `number` : champ numérique - `range` : curseur - `color` : sélecteur de couleur - `file` : téléchargement de fichier - `submit` : bouton d'envoi du formulaire - et bien d'autres...

L'accessibilité est cruciale pour les formulaires. L'élément `<label>` associé à chaque champ (via l'attribut `for` correspondant à l' `id` du champ) permet aux technologies d'assistance de comprendre la relation entre l'étiquette et le champ. Les éléments `<fieldset>` et `<legend>` permettent de regrouper logiquement les champs connexes.

HTML5 a introduit des fonctionnalités de validation native qui permettent de vérifier les données saisies avant l'envoi du formulaire : - L'attribut `required` indique qu'un champ doit être rempli - Les attributs `minlength` et `maxlength` définissent les longueurs minimale et maximale - L'attribut `pattern` permet de spécifier une expression régulière que la valeur doit respecter - Les types de champs spécifiques (`email`, `url`, etc.) incluent une validation automatique - L'attribut `title` permet de fournir un message d'erreur personnalisé

La stylisation des formulaires avec CSS est essentielle pour améliorer l'expérience utilisateur. Les aspects importants incluent : - Des étiquettes claires et bien positionnées - Des tailles de champs appropriées - Des états visuels distincts (normal, focus, erreur) - Des marges et des espacements cohérents - Des boutons clairement identifiables - Une mise en page responsive qui s'adapte aux différentes tailles d'écran

JavaScript peut être utilisé pour ajouter des validations plus complexes, comme la vérification que deux champs correspondent (par exemple, pour la confirmation de mot de passe) ou pour créer des interactions dynamiques comme la mise à jour en temps réel d'une valeur affichée.

En résumé, la création de formulaires efficaces nécessite une attention particulière à la structure HTML, à la validation des données et à l'expérience utilisateur. Un bon formulaire doit être facile à comprendre, à remplir et à soumettre, tout en garantissant que les données collectées sont valides et utiles.

# TP 8 : Projet final - Site web personnel

## Objectif

Ce travail pratique final vous permettra de mettre en pratique toutes les compétences HTML et CSS acquises au cours des TP précédents. Vous allez créer un site web personnel complet, avec plusieurs pages, une navigation cohérente, et une mise en page responsive.

## Énoncé

### Partie 1 : Structure du site

1. Créez un nouveau dossier nommé `site_personnel`
2. À l'intérieur de ce dossier, créez les fichiers suivants :
3. `index.html` (page d'accueil)
4. `a-propos.html` (page à propos)
5. `competences.html` (page de compétences)
6. `projets.html` (page de projets)
7. `contact.html` (page de contact)
8. `styles.css` (feuille de style commune)
9. Créez un sous-dossier `images` pour stocker les images du site

### Partie 2 : Création de la page d'accueil

1. Dans `index.html`, créez une structure de base avec :
2. Un en-tête (`<header>`) contenant le logo/nom et un menu de navigation

3. Une section principale ( `<main>` ) avec une bannière d'accueil et une brève présentation
4. Un pied de page ( `<footer>` ) avec des informations de copyright et des liens vers les réseaux sociaux
5. Assurez-vous que le menu de navigation contient des liens vers toutes les pages du site
6. Ajoutez une image de profil ou un avatar dans la section de présentation

## Partie 3 : Développement des autres pages

1. Pour chaque page, conservez la même structure d'en-tête et de pied de page
2. Page "À propos" :
3. Ajoutez une biographie détaillée
4. Incluez des informations sur votre formation et vos expériences
5. Ajoutez une section "Centres d'intérêt"
6. Page "Compétences" :
7. Créez une liste de compétences organisées par catégories
8. Utilisez des barres de progression ou des indicateurs visuels pour montrer votre niveau
9. Page "Projets" :
10. Créez une galerie de projets avec des images et des descriptions
11. Ajoutez des liens vers les projets (réels ou fictifs)
12. Page "Contact" :
13. Créez un formulaire de contact complet
14. Ajoutez vos coordonnées et liens vers les réseaux sociaux

## Partie 4 : Stylistation et mise en page responsive

1. Dans `styles.css` , créez des styles cohérents pour l'ensemble du site
2. Utilisez Flexbox ou Grid pour la mise en page
3. Ajoutez des media queries pour rendre le site responsive :
4. Adaptation pour les écrans larges (ordinateurs)
5. Adaptation pour les tablettes
6. Adaptation pour les smartphones
7. Créez un menu hamburger pour la version mobile
8. Assurez-vous que tous les éléments sont lisibles et accessibles sur tous les appareils

## Conseils

- Planifiez votre site avant de commencer à coder

- Utilisez une palette de couleurs cohérente (3-4 couleurs maximum)
- Privilégiez la lisibilité et la simplicité
- Testez régulièrement votre site sur différentes tailles d'écran
- Vérifiez que tous les liens fonctionnent correctement
- Optimisez vos images pour le web

## Solution

Voici une solution complète pour le projet final. Pour des raisons de clarté, nous allons présenter les fichiers principaux et expliquer la structure globale.

### Structure des fichiers

```
site_personnel/  
├── index.html  
├── a-propos.html  
├── competences.html  
├── projets.html  
├── contact.html  
├── styles.css  
└── images/  
    ├── profile.jpg  
    ├── banner.jpg  
    ├── projet1.jpg  
    ├── projet2.jpg  
    └── projet3.jpg
```

### Fichier index.html

```
<!DOCTYPE html>  
<html lang="fr">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-  
scale=1.0">  
  <title>Marie Dupont - Développeuse Web</title>  
  <link rel="stylesheet" href="styles.css">  
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/  
ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">  
</head>  
<body>  
  <header class="site-header">  
    <div class="logo">  
      <h1>Marie Dupont</h1>  
      <p class="tagline">Développeuse Web</p>
```

```

</div>

<nav class="main-nav">
  <button class="menu-toggle" aria-label="Menu
principal">
    <span class="hamburger"></span>
  </button>
  <ul class="nav-links">
    <li><a href="index.html"
class="active">Accueil</a></li>
    <li><a href="a-propos.html">À propos</a></li>
    <li><a href="competences.html">Compétences</a></
li>
    <li><a href="projets.html">Projets</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</nav>
</header>

<main>
  <section class="hero">
    <div class="hero-content">
      <h2>Bienvenue sur mon portfolio</h2>
      <p>Je crée des sites web modernes, accessibles
et performants</p>
      <a href="contact.html" class="btn btn-primary">Me
contacter</a>
    </div>
  </section>

  <section class="intro">
    <div class="profile-image">
      
    </div>
    <div class="intro-text">
      <h2>Qui suis-je ?</h2>
      <p>Bonjour ! Je suis Marie Dupont, développeuse
web passionnée par la création d'expériences numériques
innovantes et accessibles. Spécialisée en HTML, CSS et
JavaScript, je conçois des sites web qui allient esthétique et
fonctionnalité.</p>
      <p>Avec 3 ans d'expérience dans le domaine, j'ai
travaillé sur divers projets allant des sites vitrines aux
applications web complexes. Je suis constamment à la recherche
de nouveaux défis et opportunités pour développer mes
compétences.</p>
      <a href="a-propos.html" class="btn btn-
secondary">En savoir plus</a>
    </div>
  </section>

```

```

<section class="featured-projects">
  <h2>Projets récents</h2>
  <div class="project-cards">
    <div class="project-card">
      
      <h3>Site E-commerce</h3>
      <p>Conception et développement d'une
boutique en ligne responsive avec panier d'achat et système de
paiement.</p>
      <a href="projets.html#projet1" class="btn
btn-small">Voir le projet</a>
    </div>
    <div class="project-card">
      
      <h3>Application de recettes</h3>
      <p>Développement d'une application web
progressive permettant de rechercher et sauvegarder des recettes
de cuisine.</p>
      <a href="projets.html#projet2" class="btn
btn-small">Voir le projet</a>
    </div>
    <div class="project-card">
      
      <h3>Portfolio d'artiste</h3>
      <p>Création d'un portfolio élégant pour un
photographe, avec galerie d'images et système de réservation.</
p>
      <a href="projets.html#projet3" class="btn
btn-small">Voir le projet</a>
    </div>
  </div>
  <div class="center">
    <a href="projets.html" class="btn btn-
secondary">Tous les projets</a>
  </div>
</section>
</main>

<footer class="site-footer">
  <div class="footer-content">
    <div class="footer-info">
      <h3>Marie Dupont</h3>
      <p>Développeuse Web Freelance</p>
      <p>Paris, France</p>
    </div>
    <div class="footer-links">
      <h3>Liens rapides</h3>
      <ul>
        <li><a href="index.html">Accueil</a></li>

```

```

        <li><a href="a-propos.html">À propos</a></li>
    </li>
    <li><a href="competences.html">Compétences</a></li>
    <li><a href="projets.html">Projets</a></li>
    <li><a href="contact.html">Contact</a></li>
</ul>
</div>
<div class="footer-social">
    <h3>Suivez-moi</h3>
    <div class="social-icons">
        <a href="#" aria-label="LinkedIn"><i
class="fab fa-linkedin"></i></a>
        <a href="#" aria-label="GitHub"><i
class="fab fa-github"></i></a>
        <a href="#" aria-label="Twitter"><i
class="fab fa-twitter"></i></a>
        <a href="#" aria-label="Instagram"><i
class="fab fa-instagram"></i></a>
    </div>
</div>
</div>
<div class="copyright">
    <p>&copy; 2025 Marie Dupont. Tous droits réservés.</p>
</div>
</footer>

<script>
    // Script pour le menu hamburger
    document.querySelector('.menu-
toggle').addEventListener('click', function() {
        document.querySelector('.nav-
links').classList.toggle('active');
        this.classList.toggle('active');
    });
</script>
</body>
</html>

```

## Fichier styles.css

```

/* Styles généraux */
:root {
    --primary-color: #3498db;
    --secondary-color: #2c3e50;
    --accent-color: #e74c3c;
    --light-color: #ecf0f1;
    --dark-color: #2c3e50;
    --text-color: #333;
}

```

```
--background-color: #fff;
--shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
}

* {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
}

body {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  line-height: 1.6;
  color: var(--text-color);
  background-color: var(--background-color);
}

a {
  text-decoration: none;
  color: var(--primary-color);
  transition: color 0.3s;
}

a:hover {
  color: var(--accent-color);
}

h1, h2, h3 {
  margin-bottom: 1rem;
  color: var(--dark-color);
}

img {
  max-width: 100%;
  height: auto;
  display: block;
}

.btn {
  display: inline-block;
  padding: 10px 20px;
  border-radius: 4px;
  font-weight: bold;
  text-align: center;
  transition: background-color 0.3s, transform 0.3s;
}

.btn:hover {
  transform: translateY(-2px);
}
```

```
.btn-primary {
  background-color: var(--primary-color);
  color: white;
}

.btn-primary:hover {
  background-color: #2980b9;
  color: white;
}

.btn-secondary {
  background-color: var(--secondary-color);
  color: white;
}

.btn-secondary:hover {
  background-color: #1a252f;
  color: white;
}

.btn-small {
  padding: 5px 10px;
  font-size: 0.9rem;
}

.center {
  text-align: center;
  margin: 2rem 0;
}

/* En-tête */
.site-header {
  background-color: white;
  box-shadow: var(--shadow);
  padding: 1rem;
  position: sticky;
  top: 0;
  z-index: 100;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.logo h1 {
  font-size: 1.8rem;
  margin-bottom: 0;
}

.tagline {
  font-size: 1rem;
  color: var(--primary-color);
}
```

```
.main-nav .nav-links {
  display: flex;
  list-style: none;
}

.main-nav li {
  margin-left: 1.5rem;
}

.main-nav a {
  color: var(--dark-color);
  font-weight: 500;
  padding: 0.5rem;
  position: relative;
}

.main-nav a::after {
  content: '';
  position: absolute;
  bottom: 0;
  left: 0;
  width: 0;
  height: 2px;
  background-color: var(--primary-color);
  transition: width 0.3s;
}

.main-nav a:hover::after,
.main-nav a.active::after {
  width: 100%;
}

.menu-toggle {
  display: none;
  background: none;
  border: none;
  cursor: pointer;
}

.hamburger {
  display: block;
  width: 25px;
  height: 3px;
  background-color: var(--dark-color);
  position: relative;
  transition: background-color 0.3s;
}

.hamburger::before,
.hamburger::after {
  content: '';
```

```
    position: absolute;
    width: 25px;
    height: 3px;
    background-color: var(--dark-color);
    transition: transform 0.3s;
}

.hamburger::before {
    top: -8px;
}

.hamburger::after {
    bottom: -8px;
}

.menu-toggle.active .hamburger {
    background-color: transparent;
}

.menu-toggle.active .hamburger::before {
    transform: rotate(45deg) translate(5px, 5px);
}

.menu-toggle.active .hamburger::after {
    transform: rotate(-45deg) translate(5px, -5px);
}

/* Section héro */
.hero {
    background-image: linear-gradient(rgba(0, 0, 0, 0.5),
    rgba(0, 0, 0, 0.5)), url('images/banner.jpg');
    background-size: cover;
    background-position: center;
    color: white;
    text-align: center;
    padding: 6rem 2rem;
}

.hero-content {
    max-width: 800px;
    margin: 0 auto;
}

.hero h2 {
    font-size: 2.5rem;
    margin-bottom: 1rem;
    color: white;
}

.hero p {
    font-size: 1.2rem;
    margin-bottom: 2rem;
}
```

```
}

/* Section intro */
.intro {
  padding: 4rem 2rem;
  display: flex;
  align-items: center;
  max-width: 1200px;
  margin: 0 auto;
  gap: 3rem;
}

.profile-image {
  flex: 1;
  text-align: center;
}

.profile-image img {
  border-radius: 50%;
  max-width: 300px;
  margin: 0 auto;
  box-shadow: var(--shadow);
}

.intro-text {
  flex: 2;
}

.intro-text p {
  margin-bottom: 1.5rem;
}

/* Section projets en vedette */
.featured-projects {
  background-color: var(--light-color);
  padding: 4rem 2rem;
}

.featured-projects h2 {
  text-align: center;
  margin-bottom: 2rem;
}

.project-cards {
  display: flex;
  flex-wrap: wrap;
  gap: 2rem;
  justify-content: center;
  max-width: 1200px;
  margin: 0 auto;
}
```

```
.project-card {
  background-color: white;
  border-radius: 8px;
  overflow: hidden;
  box-shadow: var(--shadow);
  flex: 1 1 300px;
  max-width: 350px;
  transition: transform 0.3s;
}

.project-card:hover {
  transform: translateY(-5px);
}

.project-card img {
  width: 100%;
  height: 200px;
  object-fit: cover;
}

.project-card h3, .project-card p {
  padding: 0 1rem;
}

.project-card h3 {
  margin-top: 1rem;
}

.project-card .btn {
  margin: 1rem;
}

/* Pied de page */
.site-footer {
  background-color: var(--dark-color);
  color: white;
  padding: 3rem 2rem 1rem;
}

.footer-content {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-between;
  max-width: 1200px;
  margin: 0 auto;
  gap: 2rem;
}

.footer-info, .footer-links, .footer-social {
  flex: 1;
  min-width: 200px;
}
```

```
.footer-links ul {
  list-style: none;
}

.footer-links a {
  color: var(--light-color);
  display: block;
  padding: 0.3rem 0;
}

.footer-links a:hover {
  color: var(--primary-color);
}

.social-icons {
  display: flex;
  gap: 1rem;
  margin-top: 1rem;
}

.social-icons a {
  color: white;
  font-size: 1.5rem;
  transition: color 0.3s;
}

.social-icons a:hover {
  color: var(--primary-color);
}

.copyright {
  text-align: center;
  margin-top: 2rem;
  padding-top: 1rem;
  border-top: 1px solid rgba(255, 255, 255, 0.1);
}

/* Media queries pour la responsivité */
@media (max-width: 992px) {
  .intro {
    flex-direction: column;
    text-align: center;
  }

  .profile-image {
    margin-bottom: 2rem;
  }
}

@media (max-width: 768px) {
  .menu-toggle {
```

```
    display: block;
}

.nav-links {
  position: absolute;
  top: 100%;
  left: 0;
  width: 100%;
  background-color: white;
  flex-direction: column;
  align-items: center;
  padding: 1rem;
  box-shadow: var(--shadow);
  transform: translateY(-150%);
  opacity: 0;
  transition: transform 0.3s, opacity 0.3s;
  z-index: 99;
}

.nav-links.active {
  transform: translateY(0);
  opacity: 1;
}

.main-nav li {
  margin: 0.5rem 0;
}

.hero {
  padding: 4rem 1rem;
}

.hero h2 {
  font-size: 2rem;
}

.project-card {
  flex: 0 0 100%;
}
}

@media (max-width: 480px) {
  .site-header {
    padding: 0.8rem;
  }

  .logo h1 {
    font-size: 1.5rem;
  }

  .hero h2 {
    font-size: 1.8rem;
  }
}
```

```
    }  
    .footer-content {  
        flex-direction: column;  
    }  
}
```

## Autres pages

Les autres pages ( `a-propos.html` , `competences.html` , `projets.html` , `contact.html` ) suivent la même structure d'en-tête et de pied de page que la page d'accueil, avec un contenu spécifique à chaque page. Voici un aperçu du contenu de chaque page :

### `a-propos.html`

- Biographie détaillée
- Parcours académique et professionnel
- Centres d'intérêt et passions
- Témoignages ou recommandations

### `competences.html`

- Compétences techniques (HTML, CSS, JavaScript, etc.) avec barres de progression
- Compétences en design (UI/UX, Responsive Design, etc.)
- Outils maîtrisés (Git, Photoshop, etc.)
- Langues parlées

### `projets.html`

- Galerie complète de projets
- Filtres par catégorie (Web, Mobile, Design)
- Description détaillée de chaque projet
- Technologies utilisées et défis relevés

### `contact.html`

- Formulaire de contact complet
- Coordonnées professionnelles
- Carte de localisation
- Disponibilité et délais de réponse

# Points importants à retenir

La création d'un site web personnel complet nécessite une planification minutieuse et l'application de nombreux concepts HTML et CSS. Voici les points clés à retenir :

L'architecture du site est fondamentale. Une structure claire avec des fichiers bien organisés facilite le développement et la maintenance. La séparation du contenu (HTML) et de la présentation (CSS) est une bonne pratique qui permet de modifier l'apparence sans toucher à la structure.

La cohérence visuelle est essentielle pour une expérience utilisateur de qualité. L'utilisation de variables CSS (comme dans `:root` ) permet de définir une palette de couleurs et des styles réutilisables dans tout le site. Cette approche facilite également les modifications ultérieures.

La navigation doit être intuitive et accessible. Un menu de navigation clair, présent sur toutes les pages, aide les visiteurs à se repérer facilement. L'indication de la page active (avec une classe `.active` ) améliore l'orientation de l'utilisateur.

Le design responsive est incontournable. L'utilisation de media queries permet d'adapter la mise en page aux différentes tailles d'écran. Les techniques comme Flexbox facilitent la création de layouts qui s'adaptent naturellement à différentes résolutions.

L'accessibilité ne doit pas être négligée. L'utilisation d'attributs comme `aria-label` , de textes alternatifs pour les images, et d'une structure sémantique améliore l'expérience pour tous les utilisateurs, y compris ceux utilisant des technologies d'assistance.

Les interactions et animations subtiles améliorent l'engagement. Des effets simples comme les transitions sur les liens et les boutons rendent le site plus dynamique sans nuire à la performance ou à l'accessibilité.

La performance est un aspect crucial. L'optimisation des images, la minification des fichiers CSS et JavaScript, et l'utilisation judicieuse des ressources externes contribuent à un chargement rapide du site.

En résumé, un site web personnel réussi combine une structure solide, un design cohérent et responsive, une navigation intuitive, et une attention aux détails comme l'accessibilité et la performance. Ce projet final vous a permis d'appliquer toutes les compétences acquises au cours des TP précédents dans un contexte réel et complet.