

I. Séparation du Fond et de la Forme

Objectifs :

- Comprendre le concept de séparation du fond (HTML) et de la forme (CSS).
- Apprendre à utiliser des feuilles de styles internes, externes et en ligne.
- Découvrir l'importance de l'utilisation des fichiers CSS externes pour une meilleure maintenabilité.

Création d'une Page HTML Basique :

- Créez une nouvelle page HTML nommée index.html.
- Ajoutez-y la structure HTML minimale :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>TP1 - Introduction à CSS</title>
</head>
<body>
  <h1>Bienvenue au TP CSS</h1>
  <p>Ceci est un paragraphe simple pour démontrer l'utilisation du CSS.</p>
  <p class="important">Ce paragraphe a une classe "important".</p>
  <div id="conteneur">Ceci est une div avec un ID.</div>
</body>
</html>
```

Application de Styles en Interne :

- Ajoutez une feuille de style interne dans la balise <head> :

```
<style>
  body {
    background-color: lightgray;
    color: black;
  }
  h1 {
    color: blue;
    text-align: center;
  }
  .important {
    color: red;
    font-weight: bold;
  }
  #conteneur {
    border: 2px solid green;
    padding: 10px;
    margin-top: 20px;
  }
</style>
```

Utilisation des Styles en Ligne

- Appliquez des styles en ligne aux éléments existants :

```
<h1 style="color: purple;">Bienvenue au TP CSS</h1>
<p style="font-size: 18px;">Paragraphe avec style en ligne.</p>
```

Création d'un Fichier CSS Externe :

- Créez un fichier `styles.css` et ajoutez-y le code suivant :

```
body {
    background-color: lightblue;
    font-family: Arial, sans-serif;
}

h1 {
    color: darkblue;
    text-transform: uppercase;
}

.important {
    color: orange;
    font-size: 20px;
}

#conteneur {
    background-color: lightgreen;
    padding: 20px;
    text-align: center;
}
```

- Liez ce fichier CSS à la page HTML en utilisant la balise `<link>` dans le `<head>` :

```
<link rel="stylesheet" href="styles.css">
```

Comparaison des Méthodes :


```
/* Sélecteur d'élément */
p {
    color: black;
    font-size: 16px;
}

/* Sélecteur de classe */
.highlight {
    color: white;
    background-color: blue;
    padding: 5px;
}

/* Sélecteur d'ID */
#container {
    border: 2px solid red;
    padding: 10px;
    margin: 20px;
}

/* Sélecteur d'attribut */
[title] {
    color: purple;
    font-style: italic;
}

/* Sélecteur universel */
* {
    font-family: Arial, sans-serif;
}

/* Sélecteur multiple */
h1, p {
    text-align: center;
}
```

Ajoutez un attribut title dans le fichier HTML :

```
<p title="texte spécial">Ce paragraphe a un attribut title.</p>
```

Expérimentations :

- Changez la couleur de fond des éléments <p>.
- Ajoutez une nouvelle classe et appliquez-la à plusieurs éléments.
- Testez différents types de sélecteurs (comme le sélecteur de pseudo-classe :hover).

III. Mise en Forme du Texte avec CSS

Objectifs :

- Découvrir les différentes propriétés CSS de mise en forme du texte.
- Apprendre à utiliser des unités de mesure relatives et absolues (px, em, %).
- Expérimenter avec des polices de caractères et des styles de texte.

Créez un fichier nommé `mise_forme.html` :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>TP3 - Mise en Forme du Texte</title>
  <link rel="stylesheet" href="text-styles.css">
</head>
<body>
  <h1>Expérimentation des Propriétés CSS de Texte</h1>
  <p>Ceci est un paragraphe standard pour tester les propriétés CSS de texte.</p>
  <p class="highlight">Ce texte est important.</p>
  <blockquote>Cette citation est mise en forme différemment.</blockquote>
  <p>Le texte suivant contient un <span class="important">mot important</span>.</p>
</body>
</html>
```

Créez un fichier `text-styles.css` :

```
/* Propriétés de base */
body {
    font-family: Arial, sans-serif;
    line-height: 1.5;
}

/* Taille et Style du Texte */
p {
    font-size: 16px;
    color: #333;
}

.highlight {
    font-size: 18px;
    font-weight: bold;
    color: #ff5733;
}

blockquote {
    font-style: italic;
    color: #555;
    margin-left: 20px;
    border-left: 5px solid #ccc;
    padding-left: 10px;
}

.important {
    text-transform: uppercase;
    color: #007bff;
}
```

Ajouter des propriétés CSS supplémentaires :

- Changer la couleur et la police (font-family) du texte.
- Ajouter des ombres (text-shadow) aux titres.
- Tester différentes valeurs pour text-align (left, right, center, justify).

La mise en forme du texte, l'une des parties importantes est l'utilisation des différentes **unités de mesure** en CSS. Il existe des unités **absolues** et **relatives**, et chacune a ses avantages et inconvénients. Voici quelques exemples pour montrer comment les utiliser et quand les choisir.

1. Exemple avec des Unités Absolues (px)

Les unités absolues comme px (pixels) définissent une taille fixe. Elles ne changent pas en fonction de la taille de l'écran ou des préférences de l'utilisateur.

Exemple :

```
<h1 style="font-size: 24px;">Titre avec taille fixe (24px)</h1>  
<p style="font-size: 16px;">Ce texte utilise des pixels pour la taille (16px).</p>
```

Explication :

- La taille du texte reste la même quel que soit l'appareil utilisé (ordinateur, tablette, téléphone).
- Idéal pour des éléments de design spécifiques qui nécessitent une taille constante.

2. Exemple avec des Unités Relatives (em)

Les unités relatives comme em sont basées sur la taille de la police de l'élément parent. Par exemple, 1em équivaut à la taille de la police parente, tandis que 2em est deux fois la taille de la police parente.

Exemple :

```
<div style="font-size: 20px;">  
  <p style="font-size: 1em;">Texte avec taille 1em (20px ici).</p>  
  <p style="font-size: 1.5em;">Texte avec taille 1.5em (30px ici).</p>  
</div>
```

Explication :

- 1em équivaut à la taille de la police parente (ici 20px).
- 1.5em est 1.5 fois la taille parente (30px).
- Utile pour rendre le texte évolutif et adaptable.

3. Exemple avec des Unités Relatives (%)

Les pourcentages (%) sont également relatifs, mais se réfèrent généralement à la taille de la police de l'élément parent ou à la largeur de l'écran pour les mises en page.

Exemple :

```
<div style="font-size: 16px;">  
  <p style="font-size: 100%;">Texte avec taille 100% (16px ici).</p>  
  <p style="font-size: 150%;">Texte avec taille 150% (24px ici).</p>  
</div>
```

Explication :

- 100% correspond à la taille parente (16px).
- 150% équivaut à 1.5 fois la taille parente (24px).
- Souvent utilisé pour les mises en page adaptatives et pour respecter les préférences de l'utilisateur.

Comparaison des Unités

Unité	Description	Avantages	Inconvénients
-------	-------------	-----------	---------------

px	Taille fixe	Précis, constant	Pas adaptable, peut être trop petit ou trop grand sur certains écrans
em	Taille relative à l'élément parent	Adaptable, évolutif	Peut devenir complexe avec des éléments imbriqués
%	Taille relative à l'élément parent ou à la largeur de l'écran	Adaptable, facile à comprendre	Peut varier de manière inattendue

- Modifier les tailles de texte en utilisant px, em, et %.
- Observer comment la taille change en fonction de l'écran et du conteneur parent.
- Tester l'effet de l'utilisation de différentes unités sur la lisibilité et l'adaptabilité du texte.

Ces exemples vous permettront de comprendre quand utiliser des unités absolues ou relatives, et comment cela affecte la mise en page et l'expérience utilisateur.

IV. Création de Mise en Page avec Flexbox

Objectifs :

- Découvrir la mise en page moderne avec Flexbox.
- Apprendre à utiliser les propriétés Flexbox pour créer des dispositions réactives.

Créez un fichier flexbox.html :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>TP4 - Flexbox Layout</title>
  <link rel="stylesheet" href="flexbox.css">
</head>
<body>
  <header>En-tête du Site</header>
  <nav>Navigation</nav>
  <section>
    <div class="content">Contenu Principal</div>
    <aside>Sidebar</aside>
  </section>
  <footer>Pied de Page</footer>
</body>
</html>
```

Créez un fichier flexbox.css :


```
* {
    box-sizing: border-box;
}

body {
    display: flex;
    flex-direction: column;
    font-family: Arial, sans-serif;
    margin: 0;
}

header, nav, footer {
    background-color: #333;
    color: #fff;
    text-align: center;
    padding: 10px;
}

section {
    display: flex;
    flex: 1;
    padding: 20px;
}

.content {
    flex: 3;
    background-color: #f0f0f0;
    margin-right: 10px;
    padding: 20px;
}

aside {
    flex: 1;
    background-color: #ccc;
    padding: 20px;
}
```

Modifiez les propriétés Flexbox :

- Changez flex-direction en row et observez le changement.
- Utilisez justify-content pour centrer les éléments (center, space-between, space-around).
- Ajoutez des marges et des espacements (gap).

V. Responsive Design avec Media Queries

Objectifs :

- Apprendre à créer des designs adaptatifs avec CSS.
- Utiliser les Media Queries pour ajuster le style selon la taille de l'écran.

Créez un fichier media.html :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>TP5 - Responsive Design</title>
  <link rel="stylesheet" href="responsive.css">
</head>
<body>
  <div class="container">
    <header>Responsive Header</header>
    <main>Contenu Principal</main>
    <footer>Responsive Footer</footer>
  </div>
</body>
</html>
```

Créez un fichier responsive.css :

```
.container {
  max-width: 1200px;
  margin: auto;
  padding: 20px;
}

@media (max-width: 768px) {
  body {
    background-color: lightyellow;
  }
}

@media (max-width: 480px) {
  body {
    background-color: lightblue;
  }
}
```

- Ajustez la largeur du navigateur et observez les changements.
- Ajoutez des styles supplémentaires pour d'autres tailles d'écran (min-width, orientation).