



# TP: HTML/CSS

## Créer votre page web

**Objectif :** Vous allez concevoir une page web de type “portfolio” pour présenter votre profil, vos compétences et vos projets. Ce projet vous permettra d'explorer de nombreuses balises HTML et d'appliquer des concepts de base en HTML et en structure de page.

**Contexte :** Vous êtes un développeur web junior, et vous souhaitez créer une page de présentation qui pourrait être utilisée dans le cadre d'un futur emploi ou stage. Cette page inclura une présentation personnelle, une liste de vos compétences, un exemple de projet, et une section de contact.

# Partie 1 : HTML

## 1. Structure de la page

- **<html> et <head>** : Créez la structure de base avec la déclaration `<!DOCTYPE html>`. Ajoutez des balises `<title>`, `<meta charset="UTF-8">`

## 2. En-tête de la page :

- **<header>** : Utilisez cette balise pour l'en-tête de votre page.
- **<h1>** : Ajoutez un titre principal avec votre nom ou votre pseudo.
- **<nav>** : Créez un menu de navigation simple avec des liens vers les différentes sections de la page (par exemple : Présentation, Compétences, Projets, Contact).
- **<img>** : Ajoutez une image de profil (vous pouvez utiliser une image fictive pour tester).

## 3. Section Présentation :

- **<section>** : Utilisez `<section>` pour séparer les différentes parties de la page.
- **<h2>** : Ajoutez un sous-titre pour cette section.
- **<p>** : Rédigez un paragraphe de présentation personnelle, expliquant qui vous êtes et pourquoi vous apprenez la programmation.

## 4. Section Compétences :

- **<h2> et <ul>/<ol>** : Créez une liste (ordonnée ou non) pour détailler vos compétences (par exemple : HTML, CSS, Python, etc.).
- **<progress>** : Si vous le souhaitez, ajoutez une barre de progression pour chaque compétence afin de représenter votre niveau.

## 5. Section Projets :

- **<h2>** : Titre pour la section.
- **<article>** : Utilisez cette balise pour présenter un projet spécifique.
- **<h3>** : Titre du projet.
- **<figure> et <figcaption>** : Insérez une image du projet avec une légende.
- **<a>** : Ajoutez un lien qui redirige vers le code source de votre projet (GitHub ou autre plateforme).
- **<iframe>** : Si possible, insérez une vidéo explicative du projet ou une démo en ligne.

## 6. Section Contact :

- **<h2>** : Titre de la section.
- **<form>** : Créez un formulaire simple avec différents types de champs :
  - **<label> et <input>** : Champs pour le nom, l'adresse e-mail et un sujet.
  - **<textarea>** : Un champ de texte pour ajouter un message.
  - **<button>** : Bouton pour envoyer le formulaire (même s'il ne fonctionnera pas en l'absence de backend).

## 7. Pied de page :

- **<footer>** : Utilisez cette balise pour le bas de page.
- **<p>** : Ajoutez une petite phrase pour indiquer que la page a été créée dans le cadre d'un TP en BTS SIO.
- **<a> et <small>** : Ajoutez des liens vers vos profils sociaux ou un autre site avec des icones.

## Consignes supplémentaires :

- **Utilisez autant de balises HTML que possible** pour découvrir leurs fonctions et leurs rendus.
- Personnalisez chaque section avec un peu de CSS de base pour rendre votre page plus attractive.
- Commentez votre code HTML pour indiquer le rôle de chaque section.

## Partie 2 : CSS

**Objectif :** Vous allez utiliser CSS pour rendre votre page de portfolio plus attrayante et professionnelle. Ce projet vous permettra d'explorer différentes propriétés CSS et de renforcer vos compétences en personnalisation de pages web.

### 1. Introduction au CSS

- **Fichier CSS externe :** Créez un fichier `style.css` et liez-le à votre fichier HTML en utilisant la balise `<link>` dans la section `<head>` de votre document HTML.
- **Sélecteurs :** Utilisez des sélecteurs simples et complexes pour cibler différentes parties de votre page. (`.className` ou `#idName`)

### 2. Mise en forme de la page :

- **Police et Couleurs :**
  - Appliquez des couleurs d'arrière-plan, des couleurs de texte et des couleurs de bordure pour rendre la page visuellement cohérente.
  - Utilisez Google Fonts ou des polices de style moderne pour donner du caractère à votre page (par exemple : `font-family`).
- **Styles de Texte :**
  - Appliquez des styles de texte (police, taille, couleur, espacement des lettres) aux titres, paragraphes, et liens.
  - Ajoutez des effets de survol (`hover`) pour les liens de navigation dans l'en-tête.

### 3. Mise en page avec Flexbox ou Grid :

- **Mise en page de l'en-tête :**
  - Utilisez `display: flex;` pour organiser le contenu de votre en-tête. Placez une image thématique et le menu de navigation côte à côte.
- **Disposition des sections :**
  - Utilisez Flexbox ou CSS Grid pour structurer la disposition de chaque section :
    - **Section Présentation :** Centrez le texte et l'image de profil.
    - **Section Compétences :** Organisez vos compétences en lignes ou en colonnes.
    - **Section Projets :** Disposez les projets en grille pour un affichage clair et uniforme.

### 4. Ajoutez des effets visuels :

- **Effets de bordure et d'ombre :**
  - Ajoutez des bordures avec `border` et des ombres avec `box-shadow` aux images et aux cartes de projet pour donner un effet de profondeur.
- **Transitions et Animations :**
  - Appliquez des transitions (`transition`) sur les liens et les boutons pour des effets de survol doux.
  - Ajoutez une animation simple (par exemple, agrandir légèrement une carte de projet au survol) en utilisant `transform: scale()` et `transition`.

## 5. Style du formulaire de Contact :

- **Disposition du formulaire :**
  - Utilisez des styles `display: grid;` ou `display: flex;` pour organiser les champs du formulaire.
- **Personnalisation des champs de formulaire :**
  - Appliquez des styles de couleur de fond, des bordures arrondies (`border-radius`) et des couleurs de texte.
  - Ajoutez des effets de focus (`focus`) pour améliorer l'expérience utilisateur lorsque les champs sont actifs.

## 6. Améliorez le Pied de Page :

- **Style du pied de page :**
  - Donnez un arrière-plan différent pour le pied de page et appliquez un style aux liens sociaux en ajoutant des icônes (vous pouvez utiliser des images d'icônes).
  - Aligned le contenu et ajoutez des marges et des paddings pour un rendu propre et bien espacé.

## Consignes supplémentaires :

- **Respectez la cohérence :** Assurez-vous que les styles choisis pour les différentes sections de la page sont cohérents.
- **Expérimentez avec les propriétés CSS :** Testez des propriétés comme `text-shadow`, `letter-spacing`, `margin`, et `padding` pour ajuster l'espacement et les effets de texte.
- **Commentez votre code CSS** pour expliquer les styles appliqués.

## Rendu attendu :

Une page HTML structurée et fonctionnelle qui met en valeur vos compétences en utilisant un maximum de balises HTML de base et en incluant le fichier css pour le style.

