# Guide de base HTML & CSS

**1.** Balises ouvrantes et fermantes

- Ouverture : <nomDeBalise>
- Fermeture : </nomDeBalise>
- Le contenu se place entre les deux.

Exemple avec la balise <a> (lien) :

<a href="https://exemple.com">Clique ici</a>

- <a href="..."> : balise ouvrante
- </a> : balise fermante

### 2. Balises auto-fermantes (auto-close)

- Certaines balises ne contiennent pas de contenu.
- Elles s'ouvrent et se « ferment » en une seule fois, sans balise fermante séparée.
- Syntaxe XHTML : <nomDeBalise />

En HTML5, on peut simplement écrire <br>, <img>, etc., mais <br /> reste valide.

- Balise
   Usage
- <br> Saut de ligne
- <hr>
   <h<hr>
   <hr>
   <hr>
- <img>
   Insertion d'une image
- <input> Champ de formulaire
- <meta>
   Métadonnée dans le <head>
- k> Lien vers CSS ou favicon
- <area>
   Zone cliquable dans une image map
- <col>

   Définition de colonne dans un
- <wbr> Indique un point de coupure possible
- ٠

### Exemples :

html

### <br>> <!-- saut de ligne -->

<img src="photo.jpg" alt="image" /> <!-- image -->

<input type="text" /> <!-- champ de formulaire -->

### Attention

<a/> n'existe pas : la balise <a> nécessite un contenu (texte ou élément cliquable), donc elle ne peut pas être auto-fermante.

1. Déclaration et <head>

html

<!DOCTYPE html>

<html lang="en">

<head>

<!-- Métadonnées et liens externes -->

<meta charset="UTF-8">

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

```
<title>mon 1er exemple</title>
```

```
k rel="stylesheet" href="ex2.css">
```

</head>

- <!DOCTYPE html> : indique que c'est un document HTML5.
- <html lang="en"> : le contenu est en anglais (à ajuster si français).
- <meta charset> : encodage UTF-8 pour supporter les accents.
- <meta viewport> : rend la mise en page responsive sur mobiles.
- <title> : titre de l'onglet navigateur.
- k rel="stylesheet"> : lie le fichier CSS externe ex2.css.

2. Le corps de la page (<body>)

html

<body>

<!-- LE CONTENU à afficher -->

<h1 id="haut">titre niveau 1</h1>

<br>text1 </br> text 2

ceci est un paragraphe

Lorem ipsum dolor sit amet...

- <h1> : titre principal, ici avec un id="haut" pour y revenir via un lien d'ancre.
- <br> <br> : saut de ligne ; attention <br> est une balise auto-fermante, on écrit simplement <br> (pas </br>).
- : paragraphe de texte.

3. Hiérarchie des titres II n'y a qu'un H1 html <h2>titre niveau 2</h2> <h2>titre niveau 2</h2> <h2>titre niveau 2</h2> <h2 class="cl1">titre niveau 2</h2> ... <h3>titre niveau 3</h3> <h4>titre niveau 4</h4> <h5>titre niveau 5</h5> <h6>titre niveau 6</h6> • <h2> à <h6> : sous-titres de plus en plus secondaires.

• La classe cl1 permet d'appliquer un style CSS particulier à l'un des <h2>.

4. Listes, images et tableaux html 1 2 3 4

<img src="(l'image avec l'endroit où elle se trouve)" alt="description">

```
12
```

```
34
```

#### 

- / : liste à puces non ordonnée.
- <img> : image, avec alt descriptif pour l'accessibilité.
- : tableau HTML simple, 2 lignes de 2 cellules.

5. Lien vers le haut et navigation

html

<a href="#haut">haut</a>

<br>

<a href="ex2.html">page 2</a>

- <a href="#haut">: ancre interne vers l'élément id="haut".
- <a href="ex2.html">: lien vers un autre fichier HTML.

### 6. Formulaires

Formulaire 1

html

<form action="destination.php" method="get">

```
<label for="nom">nom</label>
```

```
<input type="text" id="nom" name="nom"/>
```

<label for="prenom">prenom</label>input type="text" id="prenom" name="prenom"/>

<!-- Boutons radio (même id à corriger !) -->

<label><input type="radio" name="rad" id="civ" value="mme" /> Mme</label>

<label><input type="radio" name="rad" id="civ" value="mme" /> M.</label>

<label><input type="checkbox" name="newsletter" value="oui" /> Je souhaite recevoir la newsletter</label>

### </form>

- action : URL où envoyer les données.
- method="get" : transmet les données dans l'URL.
- <label for="..."> et <input id="..."> : relient label et champ.
- Corriger : chaque radio doit avoir un id unique, et la valeur de l'un (value) change pour "mme" / "mr".

Formulaire 2

html

<form>

```
Vos centres d'intérêt :
```

```
<label><input type="checkbox" name="interets" value="musique"/> Musique</label><br><label><input type="checkbox" name="interets" value="cinema"/> Cinéma</label><br><label><input type="checkbox" name="interets" value="voyage"/> Voyage</label>
```

<select name="ville">

```
<option value="Paris">Paris</option>
```

<option value="Marseille">Marseille</option>

<option value="autre">autre</option>

</select>

<input type="date" name="date"> <textarea name="comm" cols="60" rows="10"></textarea>

<button type="submit">envoyer</button>

<button type="reset">annuler</button>

</form>

- <input type="date"> : sélecteur de date natif.
- <textarea> : zone de texte multilignes.
- <select> / <option> : liste déroulante.

## 7. À retenir

- Utilise seulement <br> pour un saut de ligne (pas de balise fermante).
- Chaque id doit être unique sur la page.
- Sépare idéalement ton HTML (structure), ton CSS (style) et ton JavaScript (comportement) dans des fichiers différents pour maintenir un code propre et facile à gérer.

```
La deuxième page
</html>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport"...>
<title> :
<link rel="stylesheet" href="ex2.css">
<body>
Contient le contenu visible :
<h1>...</h1> : titre principal de la page.
... : paragraphe de texte.
```

## CSS

```
/* selecteur { propriété : valeur;
```

```
propriété2 : valeur;}
```

Le sélecteur peut être le nom d'une balise, une classe ou un id

\*/

```
/* 1. Sélecteur universel */
* {
 margin: 30px;
}
/* → Cible tous les éléments HTML et leur applique une marge extérieure de 30px. */
/* 2. Classe cl1 */
/* La deuxième déclaration écrase la première sur .cl1 */
.cl1 {
 color: rgb(188, 217, 92);
}
/* \rightarrow Tous les éléments avec class="cl1" auront la couleur vert clair. */
/* 3. Sélecteur de balise  */
/* La deuxième déclaration écrase la première sur p */
p{
 color: violet;
 background-color: brown;
 padding: 20px;
}
/* \rightarrow Tous les  sont en texte violet, fond marron, et ont 20px de padding interne. */
/* 4. Sélecteur de balise <h2> */
```

h2 {

color: blue;

background-color: blue;

### }

/\*  $\rightarrow$  Tous les <h2> sont en texte bleu sur fond bleu (le texte devient invisible). \*/

/\* 5. Classe cl2 \*/ .cl2 {

```
color: red;
}
/* → Tous les éléments avec class="cl2" sont en rouge. */
/* 6. Sélecteur d'ID #id1 */
#id1 {
 color: gray;
}
/* \rightarrow L'unique élément avec id="id1" est en gris. */
/* 7. Sélecteur de balise <h1> */
h1 {
 color: red;
 background: bisque;
}
/* \rightarrow Tous les <h1> ont texte rouge et fond beige clair (bisque). */
/* 8. Sélecteur de balise <h3> */
/* Exemple de bordure commentée puis appliquée */
h3 {
 /* border-color: red;
   border-style: dashed;
```

border-width: 2px; \*/

border: 2px solid green;

# }

/\*  $\rightarrow$  Tous les <h3> auront une bordure verte solide de 2px. \*/

# /\*

Spécificité et priorité :

- ID (#id1) a une priorité plus forte qu'une classe (.cl1, .cl2).

- Les classes ont priorité plus forte qu'une balise (p, h1...).
- Si deux règles ont même spécificité, c'est la dernière dans le fichier qui s'applique.
- Enfin, la mise en forme par défaut du navigateur est la moins prioritaire.

\*/

Points clés :

Ordre des règles et spécificité déterminent quelle règle l'emporte.

Le sélecteur universel (\*) affecte tout le monde, mais ses propriétés peuvent être écrasées par des sélecteurs plus spécifiques.

Les classes (ex. .cl1) s'écrivent avec un point, les IDs (ex. #id1) avec un dièse.

Les sélecteurs de balise (ex. p, h2) s'appliquent à tous les éléments de ce type.

Les commentaires (/\* ... \*/) permettent de désactiver ou documenter des règles.

Bonnes pratiques à garder en tête

Sépare toujours ton HTML (structure) de ton CSS (styles) et de ton JS (comportement).

Assure-toi que le fichier ex2.css existe et contient bien tes règles de style.

Choisis un titre <title> explicite pour décrire le contenu de la page.

## 7. Conseils pratiques

- Sépare toujours ton HTML et ton CSS.
- Choisis des noms de classes clairs et descriptifs.
- Valide ton code avec :
  - W3C HTML Validator
  - o W3C CSS Validator
- Pratique en modifiant des exemples et en testant différentes propriétés CSS.
- N'hésite pas à copier-coller, comprendre puis adapter le code.

### 8. Balises sémantiques HTML5

1. <header>

En-tête de la page ou d'une section (logo, titre, navigation).

html

<header>

<h1>Mon site</h1>

<nav>...</nav>

</header>

2. <nav> Menu de navigation.

html

<nav>

...

</nav>

 <main> Contenu principal (un seul par page).

html

```
<main>...</main>
```

4. <section> Regroupe un contenu thématique.

html

<section>

<h2>Nos services</h2>

...

```
</section>
```

5. <article> Contenu autonome (article de blog, actualité).

html

```
<article>
```

<h3>Titre article</h3>

...

```
</article>
```

6. <footer> Pied de page (informations légales, copyrights).

html

<footer>

© 2025 Mon Site

</footer>

9. Exemple complet minimal html <!DOCTYPE html> <html lang="fr"> <head> <meta charset="UTF-8" /> <title>Exemple sémantique</title> </head> <body> <header> <h1>Mon Site</h1> <nav> <a href="#accueil">Accueil</a> <a href="#services">Services</a> <a href="#contact">Contact</a> </nav> </header> <main> <section id="accueil"> <h2>Bienvenue</h2> Introduction au site. </section>

<section id="services">

<h2>Nos services</h2>

<article>

<h3>Service 1</h3>

Description du service 1.

</article>

<article>

<h3>Service 2</h3>

Description du service 2.

</article>

</section>

</main>

<footer>

© 2025 Mon Site

</footer>

</body>

</html>

**10.** Flexbox — Disposition flexible

HTML

html

<section class="flex-example">

<h2>Exemple Flexbox</h2>

<div class="flex-container">

<div class="flex-item">Élément 1</div>

<div class="flex-item">Élément 2</div>

<div class="flex-item">Élément 3</div>

<div class="flex-item">Élément 4</div>

</div>

</section>

CSS

CSS

.flex-container { /\* active Flexbox \*/ display: flex; justify-content: space-around;/\* espace uniforme entre les items \*/ /\* aligne verticalement \*/ align-items: center; flex-wrap: wrap; /\* retour à la ligne si besoin \*/ gap: 1rem; /\* espace entre chaque item \*/ padding: 1rem; background-color: #f0f0f0; } .flex-item { background-color: #4CAF50; color: white; padding: 1rem;

border-radius: 4px; flex: 1 1 120px; /\* base de 120px, peut grandir ou rétrécir \*/

text-align: center;

}

11. CSS Grid — Mise en page en grille

### HTML

html

<section class="grid-example">

<h2>Exemple CSS Grid</h2>

<div class="grid-container">

<div class="grid-item">1</div>

<div class="grid-item">2</div>

<div class="grid-item">3</div>

<div class="grid-item">4</div>

<div class="grid-item">5</div>

```
<div class="grid-item">6</div>
</div>
</section>
CSS
CSS
.grid-container {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(150px, 1fr));
  grid-gap: 1rem; /* espace entre les cellules */
  padding: 1rem;
  background-color: #f9f9f9;
}
```

```
.grid-item {
```

background-color: #2196F3;

color: white;

padding: 1rem;

border-radius: 4px;

text-align: center;

```
}
```

# 12. Responsive Design avec Media Queries

# HTML (ajoute dans ton guide après les sections Flexbox et Grid)

```
html
```

<section class="responsive-example">

```
<h2>Exemple Responsive</h2>
```

```
<div class="responsive-container">
```

```
<div class="box">Box 1</div>
```

```
<div class="box">Box 2</div>
```

```
<div class="box">Box 3</div>
```

```
<div class="box">Box 4</div>
```

```
</div>
```

### </section>

## CSS (ajoute à la fin de ton fichier styles.css)

CSS

/\* Styles par défaut : 4 colonnes \*/

.responsive-container {

display: grid;

grid-template-columns: repeat(4, 1fr);

gap: 1rem;

padding: 1rem;

```
}
```

## .box {

background-color: #ff9800; color: white; padding: 1rem; border-radius: 4px; text-align: center;

```
}
```

```
/* Media Query pour écrans moyens (tablettes) */
@media (max-width: 992px) {
   .responsive-container {
    grid-template-columns: repeat(2, 1fr); /* 2 colonnes */
   }
}
```

```
/* Media Query pour petits écrans (smartphones) */
@media (max-width: 576px) {
   .responsive-container {
    grid-template-columns: 1fr; /* 1 seule colonne */
}
```

### **Explications**

- grid-template-columns :
  - $\circ$  repeat(4, 1fr) → 4 colonnes égales par défaut.
  - o Dans la première media query (≤ 992px) on passe à 2 colonnes.
  - Dans la seconde (≤ 576px) on passe à 1 colonne pour un affichage optimal sur mobile.
- @media :
  - S'applique **seulement** si la condition est vraie (largeur d'écran ≤ valeur).
  - Permet d'adapter la mise en page pour différentes tailles d'écran (desktop, tablette, smartphone).

0

### Deux exemple en plus

1.

html

<!DOCTYPE html>

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

<head>

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

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

<title>Blog Nature</title>

k rel="stylesheet" href="ex3.css">

</head>

- <!DOCTYPE html> : indique un document HTML5.
- <html lang="fr"> : la langue principale est le français.
- <meta charset="UTF-8"> : encodage UTF-8 (prises en charge des accents).
- <meta name="viewport"...> : rend la page responsive sur mobiles.
- <title> : titre qui apparaît dans l'onglet du navigateur.

### 2. Barre de navigation

html

<header>

<h1>Blog Nature</h1>

<nav>

<a href="#">Accueil</a>

<a href="#">Articles</a>

<a href="#">Contact</a>

</nav>

</header>

- <header> : en-tête de la page, contient le titre et le menu.
- <nav> : zone de navigation, regroupe les liens principaux.
- Les / / <a> constituent une liste de liens cliquables.

### 3. Contenu principal

html

<main>

<section>

<article>

<h2>Randonnée dans les Alpes</h2>

<time datetime="2023-06-15">15 juin 2023</time>

...

<img src="..." alt="Montagnes des Alpes">

<a href="#">Lire la suite</a>

</article>

<!-- deux autres <article> similaires -->

</section>

</main>

- **<main>** : contenu principal, unique par page.
- <section> : regroupe plusieurs articles thématiques.
- **<article>** : contient un billet de blog autonome.
- **<h2>** : titre de l'article.
- <time datetime=...> : date de publication, format machine-readable (datetime) + affichage humain.
- **:** résumé du contenu.
- <img> : illustration, avec alt pour l'accessibilité.
- <a> : lien "Lire la suite" vers l'article complet.

# 4. Formulaire d'exemple html <form action="/traitement" method="post"> <fieldset> <legend>Informations personnelles</legend> <div> <label for="nom">Nom :</label> <input type="text" id="nom" name="nom" required> </div> <div> <label for="email">Email :</label> <input type="email" id="email" name="email" required> <!-- Création de divers types d'input pour montrer la variété HTML5 --> <input type="url"> <input type="number" min="0" max="100" step="5"> <input type="range" min="0" max="100"> <input type="date"> <input type="time"> <input type="datetime-local">

<input type="color">

```
<input type="search">
```

<input type="text" placeholder="Votre nom" autocomplete="on" autofocus>

<input type="file" multiple>

</div>

<button type="submit">Envoyer</button>

## </fieldset>

</form>

- <form action=... method=...> : définit où envoyer les données (action) et comment (method).
- <fieldset> / <legend> : regroupe logiquement les champs du formulaire.
- <label for=...> : assigne un nom lisible à chaque champ et améliore l'accessibilité.
- <input> : nombreux types pour valider automatiquement différentes données :
  - text, email, url, number, range, date, time, datetime-local, color, search, file.
  - Attributs required, min, max, step, placeholder, autocomplete, autofocus, multiple.

### 5. Sélecteurs et listes déroulantes

html

```
<label for="pays">Pays :</label>
```

```
<select id="pays" name="pays">
```

```
<option value="">Sélectionnez</option>
```

<option value="FR">France</option>

```
<option value="BE">Belgique</option>
```

```
<option value="CH">Suisse</option>
```

</select>

<label for="continent">Continent :</label>

<select id="continent" name="continent">

```
<optgroup label="Europe">
```

<option value="FR">France</option>

<option value="DE">Allemagne</option>

</optgroup>

<optgroup label="Amérique">

```
<option value="US">États-Unis</option>
```

<option value="CA">Canada</option>

</optgroup>

</select>

- <select> / <option> : menu déroulant pour choisir une seule option.
- **<optgroup>** : groupe d'options, utile pour organiser les choix.

## 6. Autres champs de formulaire

html

<label for="message">Message :</label>

<textarea id="message" name="message" rows="5" cols="30"></textarea>

### <fieldset>

<legend>Centres d'intérêt :</legend>

<input type="checkbox" id="sport" name="interets" value="sport">

<label for="sport">Sport</label>

<input type="checkbox" id="musique" name="interets" value="musique">

<label for="musique">Musique</label>

</fieldset>

### <fieldset>

<legend>Sexe :</legend>

<input type="radio" id="homme" name="sexe" value="homme">

<label for="homme">Homme</label>

<input type="radio" id="femme" name="sexe" value="femme">

<label for="femme">Femme</label>

</fieldset>

<input list="navigateurs" id="navigateur" name="navigateur">

<datalist id="navigateurs">

<option value="Chrome">

<option value="Firefox">

```
<option value="Safari">
<option value="Edge">
</datalist>
```

```
<label for="satisfaction">Satisfaction :</label><input type="range" id="satisfaction" min="0" max="10" value="5"
oninput="result.value=this.value"><output id="result">5</output></output>
```

```
<label for="password">Mot de passe :</label><input type="password" id="password" required
aria-describedby="password-requirements"><div id="password-requirements" class="sr-only">8 caractères minimum, une majuscule et un chiffre.
```

```
</div>
```

- <textarea> : zone de texte multilignes.
- <input type="checkbox"> : cases à cocher, plusieurs choix possibles.
- <input type="radio"> : boutons radio, un seul choix possible.
- <datalist> : suggestions automatiques.
- <input type="range"> + <output> : curseur + affichage de la valeur en temps réel.
- <input type="password"> : champ masqué, instructions accessibles via aria-describedby.
- .sr-only (CSS non montré ici) : masque visuellement tout en restant accessible aux lecteurs d'écran.

# 7. Pied de page

html

<footer>

```
© 2023 - Tous droits réservés
```

</footer>

• **<footer>** : section en bas de page, informations légales ou de contact.

### En résumé

- Structure sémantique : <header>, <nav>, <main>, <section>, <article>, <footer>.
- Variété de champs : HTML5 propose de nombreux types d'<input> pour une validation native.
- Accessibilité : utilisation de <label>, aria-, et .sr-only.
- **Responsive** : le meta viewport dans <head> prépare au responsive (Flexbox et Grid en CSS gèrent la mise en page adaptative).

## 1. Réinitialisation de base

CSS

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

- Sélecteur universel \* : cible tous les éléments.
- margin: 0; padding: 0; supprime les marges et paddings par défaut du navigateur.
- **box-sizing: border-box;** inclut la bordure et le padding dans la largeur/hauteur totales des éléments, simplifiant le calcul des dimensions.

# 2. Style général du <body>

CSS

CopierModifier

body {

font-family: Arial, sans-serif;

background-color: #f5f5f5;

color: #333;

}

• Déclare la police par défaut, la couleur d'arrière-plan et la couleur du texte.

# 3. En-tête (<header>) et navigation

CSS

```
header {
 background-color: #1D6154;
 color: white;
 padding: 30px 20px;
 text-align: center;
}
header h1 {
 font-size: 2.5rem;
}
nav ul {
 display: flex;
 justify-content: center;
 list-style: none;
 margin-top: 15px;
}
nav ul li {
 margin: 0 15px;
}
nav ul li a {
 color: white;
 text-decoration: none;
 font-weight: bold;
}
nav ul li a:hover {
 text-decoration: underline;
}
```

- header : fond vert foncé, texte blanc, centrage.
- **nav ul** : transforme la liste en barre horizontale grâce à Flexbox, supprime les puces.
- **a:hover** ajoute un soulignement au passage de la souris.

4. Section des articles (<main>, <section>, <article>)

```
CSS
main {
 max-width: 1200px;
 margin: 40px auto;
 padding: 0 20px;
}
section {
 display: grid;
 grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
 gap: 30px;
}
article {
 background-color: white;
 padding: 20px;
 border-radius: 10px;
 box-shadow: 0 4px 12px rgba(0,0,0,0.1);
 transition: transform 0.2s ease-in-out;
}
article:hover {
 transform: translateY(-5px);
}
/* Titres, date, paragraphe, image et lien dans l'article */
article h2 { color: #1D6154; margin-bottom: 10px; }
article time {
 font-size: 0.9rem; color: #777; display: block; margin-bottom: 10px;
}
article p { margin: 10px 0 15px; }
article img {
 width: 100%; height: auto; border-radius: 5px; margin-bottom: 10px;
}
article a {
```

display: inline-block; color: #1D6154; font-weight: bold; text-decoration: none;

}

article a:hover { text-decoration: underline; }

- main centré et limité en largeur.
- section utilise CSS Grid pour un affichage réactif des articles.
- article a un fond blanc, coins arrondis, ombre et effet "lift" au hover.

## 5. Formulaire (<form>, <fieldset>, <input>, <button>)

```
CSS
form {
 max-width: 500px;
 margin: 20px auto;
 padding: 15px 20px;
 background-color: #f9f9f9;
 border-radius: 8px;
 box-shadow: 0 0 10px rgba(0,0,0,0.1);
}
/* Regroupe */
fieldset {
 border: 1.5px solid #ddd;
 padding: 15px 20px; margin-bottom: 20px;
 border-radius: 8px; background-color: #fafafa;
}
/* Légende */
legend {
 font-weight: bold; color: #1d6154; padding: 0 8px;
}
/* Labels */
label {
 display: block; margin-bottom: 6px; font-weight: bold; color: #333;
}
```

```
/* Tous les inputs / select / textarea */
select, textarea,
input[type="text"], input[type="list"],
input[type="range"], input[list] {
 width: 100%; padding: 8px 10px; margin-bottom: 15px;
 border: 1.5px solid #ccc; border-radius: 5px;
 font-size: 1rem; transition: border-color 0.3s ease;
}
select:focus, textarea:focus, input:focus {
 border-color: #1d6154; outline: none; background-color: #fff;
}
/* Checkbox & radio agrandis */
input[type="checkbox"], input[type="radio"] {
 margin-right: 8px; transform: scale(1.2); cursor: pointer;
}
/* Bouton submit */
button[type="submit"] {
 background-color: #1D6154; color: white;
 padding: 10px 25px; border: none; border-radius: 6px;
 cursor: pointer; font-weight: bold; font-size: 1rem;
 transition: background-color 0.3s ease;
}
button[type="submit"]:hover {
 background-color: #14503e;
}
/* Afficheur pour <input type="range"> */
output {
 display: inline-block; margin-left: 10px;
 font-weight: bold; color: #1d6154; min-width: 25px; text-align: center;
}
```

```
/* Responsive mobile */
```

@media (max-width: 480px) {

form { padding: 10px 15px; }

}

- Formulaire centré, fond clair, ombre.
- **fieldset** et **legend** pour la structure.
- Tous les champs ont le même style et un focus visible.
- Media query pour réduire le padding sur petits écrans.

### 6. Pied de page

CSS	
ooter {	
background-color: #1D6154;	
color: white;	
text-align: center;	
padding: 20px;	
margin-top: 40px;	

}

• Pied de page cohérent avec l'en-tête : fond vert, texte blanc, centré.

# Conclusion

Ce CSS propose un **design cohérent** (même palette de couleur), utilise **Flexbox** et **Grid** pour la mise en page, intègre des **effets au hover**, assure une bonne **accessibilité visuelle** (focus, contrastes) et est **responsive** grâce à une media query.

## Structure globale du fichier

html

<!DOCTYPE html>

<html lang="fr">

<head>

<!-- Métadonnées et styles -->

<meta charset="UTF-8">

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

<title>Blog Nature</title>

k rel="stylesheet" href="ex3.css">

<!-- (Si on intégrait du CSS inline, on le mettrait ici entre <style>...</style>) -->

</head>

<body>

<!-- Corps de la page : HTML, puis JS en bas si besoin -->

...

<!-- (Si on intégrait du JavaScript inline, on le mettrait ici entre <script>...</script>) -->

</body>

</html>

### **Contenu HTML (sémantique)**

- 1. <header>
  - <h1>Blog Nature</h1> : titre principal.
  - <nav> : menu de navigation avec une liste de liens (----<a>).

### 2. **<main>**

- <section> contenant plusieurs <article>.
- Chaque <article> :
  - <h2>: titre de l'article.
  - <time datetime="..."> : date format machine + affichage humain.
  - >: résumé.
  - <img> : image illustrant l'article, avec alt pour l'accessibilité.
  - <a>: lien "Lire la suite".

### 3. Formulaire

- <form action="..."> : cible du traitement serveur.
- <fieldset> / <legend> : regroupe les champs de manière sémantique.
- o <label for="..."> + <input> : champ texte (nom, email).
- Nombreux <input type="..."> démonstratifs des différents types HTML5 (url, number, range, date, time, datetime-local, color, search, file).
- <button type="submit"> : bouton d'envoi.

### 4. Choix avancés

- <select> / <option> / <optgroup> : listes déroulantes et groupes d'options.
- <textarea> : zone de texte multi-lignes.
- Cases à cocher (checkbox) et boutons radio (radio) regroupés dans des <fieldset>.
- <datalist> : suggestions automatiques pour un <input list="...">.
- <input type="range"> + <output> : curseur + affichage dynamique de la valeur.
- <input type="password">: champ mot de passe, avec instructions cachées .sr-only pour les lecteurs d'écran.

### 5. **<footer>**

Mentions légales ou copyright.

### Où placer le CSS et le JavaScript ?

• CSS

Idéalement dans **ex3.css**, relié via <link> dans le <head>. On évite les <style> inline pour garder la maintenance plus simple.

• JavaScript Si on avait du code JS pour valider ou enrichir le formulaire, on le mettrait :

html

```
<script src="script.js"></script>
```

</body>

</html>

(Juste avant la fermeture de </body>) ou dans un bloc <script> en bas de page.

### Pourquoi séparer ?

- 1. **Lisibilité** : le HTML garde la structure, le CSS le style, le JS le comportement.
- 2. Maintenance : modifier le style sans toucher au markup HTML, et vice-versa.
- 3. **Performance** : les fichiers CSS/JS externes sont mis en cache par le navigateur.