

UAA3 – Site web

Partie 1 : Coder en HTML

1. Bases du HTML	2
2. Balises de structuration	6
3. Listes	10
4. Liens	13
5. Images	15
6. Tableaux	17
7. Formulaire	19
8. Sémantique HTML5.....	21
9. Notions avancées	23
10. Liste des balises HTML	25

1. BASES DU HTML

HTML (HyperText Markup Language) est le langage de base pour créer des pages web. Il permet de structurer le contenu d'une page avec des éléments tels que des titres, des paragraphes, des images et des liens.

Balises et attributs en HTML

En HTML, le contenu est structuré et mis en forme grâce à des **balises**. Imaginez les balises comme des instructions qui indiquent au navigateur web comment afficher le contenu d'une page.

1. Balises ouvrantes et fermantes

La plupart des balises se présentent sous la forme d'une paire :

- **Balise ouvrante** : `<nom_de_la_balise>` (ex: `<h1>`, `<p>`, ``)
- **Balise fermante** : `</nom_de_la_balise>` (ex: `</h1>`, `</p>`, ``)

Le contenu à afficher se place entre la balise ouvrante et la balise fermante.

Exemple à tester :

```
| <p>Ceci est un paragraphe de texte.</p>
```

Certaines balises, comme ``, sont **auto-fermantes**, c'est-à-dire qu'elles ne nécessitent pas de balise fermante.

2. Attributs

Les attributs permettent de fournir des informations supplémentaires sur une balise. Ils se placent à l'intérieur de la balise ouvrante et sont composés de :

- Un **nom d'attribut** (ex: `src`, `alt`, `class`, `id`)
- Une **valeur** entre guillemets (ex: `"image.jpg"`, `"description de l'image"`, `"menu"`, `"paragraphe1"`)

Exemple à tester :

```
| 
```

Dans cet exemple, la balise `` possède deux attributs :

- `src` : indique le chemin d'accès à l'image.
- `alt` : fournit une description textuelle de l'image, utile pour l'accessibilité et le référencement.

Exemples d'attributs courants :

- **src**: Source d'une image ou d'un fichier multimédia.
- **alt**: Texte alternatif pour les images.
- **href**: URL cible d'un lien hypertexte.
- **class**: Nom de classe pour appliquer des styles CSS.

- **id:** Identifiant unique d'un élément.
- **width et height:** Largeur et hauteur d'une image.

Tester :

Utiliser l'exemple précédent pour tester les différents attributs.

Les balises et les attributs sont les éléments fondamentaux du langage HTML. Ils permettent de structurer le contenu d'une page web et de lui donner du sens.

Structure d'une page HTML

Une page web est un document structuré grâce au langage HTML. Cette structure suit un schéma précis, essentiel pour que le navigateur puisse interpréter et afficher correctement le contenu.

Voici les éléments clés qui composent la structure d'une page HTML :

1. DOCTYPE

La première ligne d'un document HTML est la déclaration DOCTYPE. Elle indique au navigateur la version de HTML utilisée. Pour HTML5, la déclaration est simple :

```
| <!DOCTYPE html>
```

2. Élément <html>

L'élément <html> englobe tout le contenu de la page web. Il possède un attribut lang pour spécifier la langue du contenu (ex: lang="fr" pour le français).

```
| <!DOCTYPE html>  
| <html lang="fr">  
|  
| </html>
```

3. Élément <head>

L'élément <head> contient des informations sur la page web, appelées **métadonnées**. Ces informations ne sont pas affichées directement sur la page, mais elles sont importantes pour le navigateur et les moteurs de recherche.

- **<title>:** Définit le titre de la page, qui apparaît dans l'onglet du navigateur et dans les résultats de recherche.
- **<meta>:** Fournit diverses informations, comme l'encodage des caractères, la description de la page, les mots-clés, etc.
- **<meta charset="UTF-8"> :** Définit l'encodage des caractères pour un affichage correct.

Exemple à tester :

```
<head>
  <meta charset="UTF-8">
  <title>Ma première page web</title>
</head>
```

4. Élément <body>

L'élément <body> contient le contenu visible de la page web, c'est-à-dire ce qui sera affiché dans la fenêtre du navigateur. C'est ici que vous placez vos titres, paragraphes, images, liens, etc.

Exemple complet à reproduire :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Ma première page web</title>
</head>
<body>

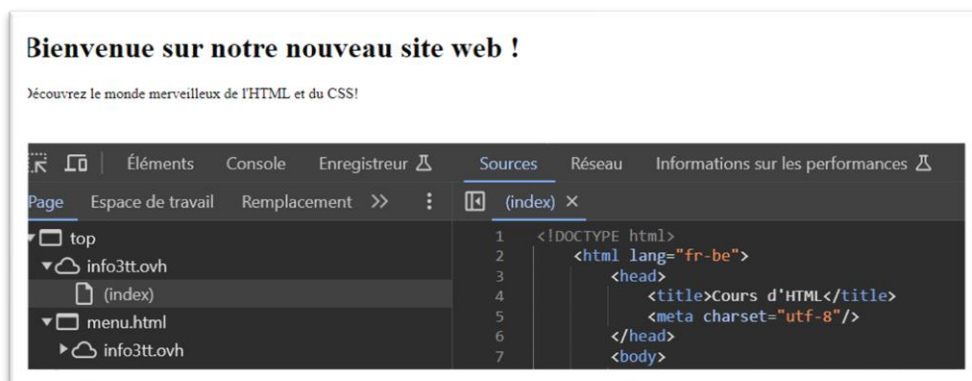
  <h1>Ma première page web</h1>
  <p>Bonjour le monde !</p>
  

</body>
</html>
```

Remarque :

Envie de visualiser le code source des pages web dans un navigateur ?

- clic droit -> inspecter -> sources



Exercice :

1. Créez un nouveau fichier texte et nommez-le "index.html".
2. Copiez le code HTML ci-dessus dans votre fichier.
3. Remplacez "logo.png" par le nom d'une image que vous avez sur votre ordinateur.
4. Enregistrez le fichier et ouvrez-le dans votre navigateur web.

Vous devriez voir votre première page web avec un titre, un paragraphe et une image !

Remarque :

Deux excellents logiciels gratuits à télécharger pour coder en HTML, ainsi qu'un compilateur en ligne :

1. Visual Studio Code (VS Code)

- **Téléchargement** : <https://code.visualstudio.com/>

2. Notepad++

- **Téléchargement** : <https://notepad-plus-plus.org/>

Compilateur en ligne :

- **CodePen** : <https://codepen.io/>

Conseils :

Commencez avec Notepad++ pour sa simplicité ou avec le compilateur en ligne. Puis explorez VS Code pour ses fonctionnalités avancées.

2. BALISES DE STRUCTURATION

Les balises de structuration permettent d'organiser le contenu d'une page web de manière logique et sémantique. Elles aident à hiérarchiser l'information, à la rendre plus lisible et accessible, et à améliorer le référencement.

1. Titres (h1 à h6)

Les balises de titre (<h1> à <h6>) définissent les titres et sous-titres d'une page. <h1> représente le titre principal, <h2> les sous-titres de niveau 1, <h3> les sous-titres de niveau 2, et ainsi de suite.

Importance pour le référencement et l'accessibilité :

- **Référencement** : Les moteurs de recherche utilisent les titres pour comprendre le sujet d'une page et la classer dans les résultats de recherche.
- **Accessibilité** : Les titres aident les personnes utilisant des lecteurs d'écran à naviguer et à comprendre la structure du contenu.

Testez-les !

2. Paragraphes (<p>)

La balise <p> définit un paragraphe de texte. Le navigateur ajoute automatiquement un saut de ligne avant et après chaque paragraphe.

Testez !

3. Divisions (<div>)

La balise <div> est un conteneur générique qui permet de regrouper des éléments HTML. Elle est souvent utilisée pour créer des sections distinctes dans une page, par exemple une zone d'en-tête, un menu de navigation, une zone de contenu principal, un pied de page, etc.

4. Spans ()

La balise permet de sélectionner une partie de texte à l'intérieur d'un paragraphe ou d'un autre élément. Elle est souvent utilisée pour appliquer des styles CSS spécifiques à un mot ou un groupe de mots, par exemple pour mettre en évidence des mots-clés.

5. Commentaires (<!-- -->)

Les commentaires peuvent être placés n'importe où dans un document HTML. Ils sont placés entre les chaînes de caractère <!-- et -->.

Exemple à tester :<!-- Ceci est un commentaire -->

6. Retour à la ligne (
)

Pour forcer un **retour à la ligne** on utilise **
**.

Testez-les !

7. Ligne séparatrice (<hr/>)

Pour placer une **ligne séparatrice horizontale** entre deux parties de la page on utilise **<hr/>** (Horizontal Rule).

Testez-les !

8. Mise en forme du texte

Il existe différentes balises pour mettre en forme le texte et lui donner un aspect visuel spécifique. Voici les principales balises pour le gras, l'italique, le soulignement, les exposants et les indices, ainsi que l'alignement du texte.

1. Gras, italique et souligné

- ** ou ** : Met le texte en gras. **** est préférable car il a une signification sémantique plus forte (importance).
- ** ou <i>** : Met le texte en italique. **** est préférable car il indique une mise en valeur.
- **<u>** : Souligne le texte. Attention, le soulignement est souvent utilisé pour les liens hypertextes, il est donc préférable de l'utiliser avec parcimonie.

Exemple à tester :

```
<p>Ceci est un texte <strong>important</strong>.</p>
```

```
<p>Ceci est un texte <em>mis en valeur</em>.</p>
```

```
<p>Ceci est un texte <u>souligné</u>.</p>
```

2. Exposant et indice

- **<sup>** : Crée un exposant (ex: x^2).
- **<sub>** : Crée un indice (ex: H_2O).

Exemple à tester:

```
<p>La formule de l'eau est H<sub>2</sub>O.</p>
```

```
<p>E=mc<sup>2</sup></p>
```

3. Alignement du texte

L'alignement du texte peut être contrôlé avec l'attribut `style` et la propriété `text-align` dans la balise de l'élément concerné (paragraphe, titre, etc.).

- **text-align: left;** : Alignement à gauche (par défaut).
- **text-align: center;** : Alignement au centre.
- **text-align: right;** : Alignement à droite.
- **text-align: justify;** : Justification du texte (les deux bords sont alignés).

Exemple à tester:

```
<p style="text-align: center;">Ce texte est centré.</p>  
<p style="text-align: right;">Ce texte est aligné à droite.</p>
```

Remarques :

- Il est possible de combiner plusieurs styles de mise en forme (ex: `texte en gras et italique`).
- Il est préférable d'utiliser les balises sémantiques (``, ``) plutôt que les balises purement visuelles (``, `<i>`).
- Pour une mise en forme plus complexe, il est recommandé d'utiliser les feuilles de style CSS.

Exercice :

Créez un fichier HTML et expérimentez les différentes balises de mise en forme du texte. Essayez de :

- Mettre du texte en gras, italique et souligné.
- Créer un exposant et un indice.
- Aligner du texte à gauche, à droite, au centre et justifier du texte.

N'hésitez pas à combiner les différentes balises et à observer les résultats !

Exercice : Structurer un article

Objectif : Mettre en pratique l'utilisation des balises de structuration pour organiser le contenu d'un article sur "Le chat sauvage".

Consignes :

1. Créer un nouveau fichier HTML.
2. Structurer l'article avec les éléments suivants :
 - Un titre principal (h1) : "Le chat sauvage"
 - Des sous-titres (h2) : "Habitat", "Alimentation", "Comportement"
 - Des paragraphes de texte pour chaque section (vous pouvez inventer le contenu).
3. Utiliser des <div> pour séparer les sections "Habitat", "Alimentation" et "Comportement".
4. Mettre en évidence certains mots-clés du texte avec des .

Exemple de code HTML:

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Le chat sauvage</title>
</head>
<body>

  <h1>Le chat sauvage</h1>
  <div>
    <h2>Habitat</h2>
    <p>Le chat sauvage vit principalement dans les forêts ... <span class="mot-cle">denses</span> et ...</p>
  </div>

  <div>
    <h2>Alimentation</h2>
    <p>Le chat sauvage est un <span class="mot-cle">prédateur</span> carnivore qui se nourrit de ...</p>
  </div>

  <div>
    <h2>Comportement</h2>
    <p>Le chat sauvage est un animal <span class="mot-cle">solitaire</span> et <span class="mot-cle">territorial</span> ...</p>
  </div>
</body>
</html>
```

3. LISTES

Les listes permettent d'organiser des éléments d'information de manière claire et structurée. HTML propose deux types de listes :

1. Listes non ordonnées ()

- Utilisées pour des éléments sans ordre particulier (ex: liste de courses, ingrédients).
- La balise définit la liste.
- Chaque élément de la liste est défini par la balise .
- **Types de puces :**
 - disc (puce pleine par défaut)
 - circle (puce ronde)
 - square (puce carrée)

Exemple à tester :

```
<ul>
  <li>Pommes</li>
  <li>Bananes</li>
  <li>Oranges</li>
</ul>
```

Testez les différents types de puces à ajoutant un attribut !

Exemple : <ul type="square"> ...

2. Listes ordonnées ()

- Utilisées pour des éléments avec un ordre précis (ex: étapes d'une recette, plan d'un document).
- La balise définit la liste.
- Chaque élément de la liste est défini par la balise .
- **Types de numérotation :**
 - 1 (chiffres arabes par défaut)
 - a (lettres minuscules)
 - A (lettres majuscules)
 - i (chiffres romains minuscules)
 - I (chiffres romains majuscules)

Exemple :

```
<ol>  
<li>Préchauffer le four.</li>  
<li>Mélanger les ingrédients.</li>  
<li>Enfourner pendant 30 minutes.</li>  
</ol>
```

Testez les types de puces !

3. Exercice :

Objectif : Mettre en pratique l'utilisation des listes pour présenter une recette.

Consignes :

1. Créer un nouveau fichier HTML.
2. Choisir une recette simple.
3. Créer une liste non ordonnée des ingrédients.
4. Créer une liste ordonnée des étapes de la recette.

Exemple de code HTML (recette de crêpes) :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Recette de crêpes</title>
</head>
<body>

  <h1>Recette de crêpes</h1>

  <h2>Ingrédients</h2>
  <ul>
    <li>250g de farine</li>
    <li>500ml de lait</li>
    <li>3 œufs</li>
    <li>50g de beurre fondu</li>
    <li>1 pincée de sel</li>
  </ul>

  <h2>Préparation</h2>
  <ol>
    <li>Dans un saladier, mélanger la farine et le sel.</li>
    <li>Ajouter les œufs un à un en mélangeant bien.</li>
    <li>Verser le lait progressivement en mélangeant pour éviter les grumeaux.</li>
    <li>Ajouter le beurre fondu.</li>
    <li>Laisser reposer la pâte pendant 30 minutes.</li>
    <li>Faire cuire les crêpes dans une poêle chaude légèrement huilée.</li>
  </ol>

</body>
</html>
```

4. LIENS

1. Les liens hypertextes en HTML

Les liens hypertextes sont la base de la navigation sur le Web. Ils permettent de relier des pages entre elles, d'accéder à des ressources externes, et de naviguer au sein d'une même page.

Balise `<a>` et attribut `href`

- La balise `<a>` (pour "anchor" en anglais) crée un lien hypertexte.
- L'attribut `href` (pour "hypertext reference") spécifie la destination du lien (URL ou ancre).
- Le texte ou l'image placé entre les balises `<a>` et `` sera cliquable.

Exemple à tester :

```
| <a href="https://www.wikipedia.org">Visitez Wikipédia</a>
```

2. Liens absolus

- Un lien absolu contient l'URL complète de la destination, incluant le protocole (`http://` ou `https://`), le nom de domaine et le chemin d'accès.
- Utilisés pour pointer vers des pages externes à votre site web.

Exemple à tester :

```
| <a href="https://www.google.be">Aller sur Google</a>
```

3. Liens relatifs

- Un lien relatif indique le chemin d'accès à la destination par rapport au fichier HTML courant.
- Utilisés pour lier des pages au sein de votre propre site web.

Exemples :

- `page2.html` : pointe vers un fichier "page2.html" situé dans le même dossier que le fichier HTML courant.
- `dossier/page3.html` : pointe vers un fichier "page3.html" situé dans un sous-dossier "dossier".
- `../page4.html` : pointe vers un fichier "page4.html" situé dans le dossier parent.

4. Ancres

- Une ancre est un point de repère dans une page web.
- Permet de créer des liens internes pour naviguer vers des sections spécifiques de la page.
- **Définition d'une ancre :** ``
- **Lien vers une ancre :** `Aller à la section`
- **Testez !**

Exercice :

Objectif : Mettre en pratique la création de liens hypertextes.

Consignes :

1. Créer un nouveau fichier HTML nommé "index.html".
2. Ajouter les éléments suivants :
 - Un lien vers Wikipédia (lien absolu).
 - Un lien vers un autre fichier HTML nommé "page2.html" (lien relatif, à créer également).
 - Une section "Contact" en bas de page avec un titre <h2>.
 - Un lien en haut de page pointant vers la section "Contact" (ancrage).

Exemple de code HTML (index.html) :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Liens hypertextes</title>
</head>
<body>

  <a href="https://www.wikipedia.org">Visitez Wikipédia</a>
  <br>
  <a href="page2.html">Aller à la page 2</a>
  <br>
  <a href="#contact">Contact</a>

  <p>Contenu de la page...</p>

  <h2 id="contact">Contact</h2>
  <p>Informations de contact...</p>

</body>
</html>
```

N'oubliez pas de créer le fichier "page2.html" !

5. IMAGES

Les images sont essentielles pour rendre une page web attractive et informative. En HTML, on utilise la balise `` pour insérer des images.

1. Balise `` et ses attributs

- **``** : Balise auto-fermante (pas de balise fermante).
- **src (source)** : Attribut obligatoire qui indique le chemin d'accès à l'image (URL absolue ou relative).
- **alt (texte alternatif)** : Attribut important pour l'accessibilité. Il fournit une description textuelle de l'image pour les personnes utilisant des lecteurs d'écran ou si l'image ne peut pas être affichée.
- **width et height** : Spécifient la largeur et la hauteur de l'image en pixels.

Exemple à tester :

```

```

2. Formats d'images

- **JPEG** : Format compressé, idéal pour les photos et les images avec beaucoup de couleurs.
- **PNG** : Format sans perte, idéal pour les images avec des zones de transparence ou des graphiques.
- **GIF** : Format limité en couleurs, idéal pour les animations simples.

Choisir le format adapté :

- **Photos** : JPEG
- **Logos et icônes** : PNG
- **Animations** : GIF

Exercice :

Objectif : Mettre en pratique l'insertion d'images avec la balise .

Consignes :

1. Créer un nouveau fichier HTML.
2. Choisir 3 images différentes (ex: une photo, un logo, un graphique).
3. Insérer les images dans la page web en utilisant la balise .
4. Utiliser des attributs alt pertinents pour chaque image.
5. Ajuster la taille d'une des images avec les attributs width et height.

Exemple de code HTML :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Les images</title>
</head>
<body>

  <h1>Images</h1>

</body>
</html>
```

Remarques :

- Assurez-vous que les images sont dans le même dossier que votre fichier HTML ou indiquez le chemin d'accès correct dans l'attribut src.
- L'attribut alt est essentiel pour l'accessibilité et le référencement.
- Vous pouvez ajuster la taille des images en modifiant les valeurs des attributs width et height.

6. TABLEAUX

Les tableaux permettent de présenter des données de manière structurée en lignes et en colonnes. Ils sont utiles pour afficher des informations tabulaires, comme des horaires, des résultats, des catalogues, etc.

1. Structure d'un tableau

- **<table>** : Définit le tableau.
- **<tr> (table row)** : Définit une ligne du tableau.
- **<td> (table data)** : Définit une cellule de données dans une ligne.
- **<th> (table header)** : Définit une cellule d'en-tête de colonne.

Exemple simple à tester:

```
<table>
<tr>
  <th>Nom</th>
  <th>Prénom</th>
</tr>
<tr>
  <td>Doe</td>
  <td>John</td>
</tr>
</table>
```

2. Attributs

- **border** : Définit la largeur de la bordure du tableau en pixels (ex: border="1").
- **cellspacing** : Définit l'espace entre les cellules en pixels.
- **cellpadding** : Définit l'espace entre le contenu d'une cellule et sa bordure en pixels.

Exemple avec attributs à tester :

```
<table border="1" cellspacing="5" cellpadding="10">
<tr>
  <th>Nom</th>
  <th>Prénom</th>
</tr>
<tr>
  <td>Doe</td>
  <td>John</td>
</tr>
</table>
```

Exercice :

Objectif : Mettre en pratique la création de tableaux en HTML.

Consignes :

1. Créer un nouveau fichier HTML.
2. Créer un tableau avec 3 colonnes : Nom, Prénom, Age.
3. Ajouter 3 lignes de données au tableau (vous pouvez inventer les données).
4. Utiliser les balises <th> pour les en-têtes de colonnes.
5. Ajouter une bordure au tableau en utilisant l'attribut border.

Exemple de code HTML :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Tableaux</title>
</head>
<body>

  <h1>Tableau des personnes</h1>

  <table border="1">
    <tr>
      <th>Nom</th>
      <th>Prénom</th>
      <th>Age</th>
    </tr>
    <tr>
      <td>Doe</td>
      <td>John</td>
      <td>30</td>
    </tr>
    <tr>
      <td>Dupont</td>
      <td>Marie</td>
      <td>25</td>
    </tr>
    <tr>
      <td>Martin</td>
      <td>Pierre</td>
      <td>40</td>
    </tr>
  </table>

</body>
</html>
```

Remarques :

- Vous pouvez ajouter autant de lignes et de colonnes que nécessaire à votre tableau.
- Les attributs cellspacing et cellpadding permettent d'ajuster l'espacement entre les cellules et le contenu.
- Il est possible de fusionner des cellules horizontalement avec l'attribut colspan et verticalement avec l'attribut rowspan (notions plus avancées).

7. FORMULAIRES

Les formulaires permettent aux utilisateurs d'interagir avec un site web en envoyant des données au serveur. Ils sont utilisés pour diverses actions, comme la création de compte, la connexion, l'envoi de messages, la recherche, etc.

1. Balise <form> et ses attributs

- **<form>** : Définit le formulaire.
- **action** : Spécifie l'URL du script qui traitera les données du formulaire.
- **method** : Spécifie la méthode d'envoi des données :
 - GET : les données sont ajoutées à l'URL (visible dans la barre d'adresse).
 - POST : les données sont envoyées dans le corps de la requête (plus sécurisé pour les données sensibles).

Exemple à tester:

```
<form action="traitement.php" method="post">
</form>
```

2. Types d'input

La balise <input> permet de créer différents types de champs de saisie :

- **text** : Champ de texte simple.
- **password** : Champ de mot de passe (les caractères sont masqués).
- **email** : Champ pour saisir une adresse email.
- **number** : Champ pour saisir un nombre.
- **radio** : Boutons radio pour choisir une seule option parmi plusieurs.
- **checkbox** : Cases à cocher pour choisir plusieurs options.
- **submit** : Bouton pour envoyer le formulaire.
- **reset** : Bouton pour réinitialiser les champs du formulaire.

3. Autres éléments de formulaire

- **<textarea>** : Zone de texte multiligne.
- **<select>** : Liste déroulante.
- **<option>** : Définit les options d'une liste déroulante.

Exercice :

Objectif : Mettre en pratique la création d'un formulaire de contact simple.

Consignes :

1. Créer un nouveau fichier HTML.
2. Créer un formulaire avec les champs suivants :
 - Nom (champ de texte)
 - Email (champ email)
 - Message (zone de texte multiligne)
3. Ajouter un bouton "Envoyer".

Exemple de code HTML :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Formulaire de contact</title>
</head>
<body>

  <h1>Formulaire de contact</h1>

  <form action="traitement.php" method="post">
    <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>
    </div>
    <div>
      <label for="message">Message :</label>
      <textarea id="message" name="message" required></textarea>
    </div>
    <button type="submit">Envoyer</button>
  </form>

</body>
</html>
```

Remarques :

- L'attribut name est utilisé pour identifier les champs du formulaire lors de l'envoi des données.
- L'attribut required indique que le champ est obligatoire.
- La balise <label> est utilisée pour associer un label à un champ de formulaire.
- Le fichier "traitement.php" (à créer séparément) contiendra le code pour traiter les données du formulaire.

8. SÉMANTIQUE HTML5

HTML5 a introduit de nouvelles balises qui permettent de structurer le contenu de manière plus sémantique. Ces balises apportent du sens à la structure de la page, ce qui améliore l'accessibilité, le référencement et la maintenance du code.

1. Nouvelles balises sémantiques

- **<header>** : En-tête de la page ou d'une section. Contient généralement le logo, le titre principal, le menu de navigation, etc.
- **<nav>** : Contient les liens de navigation.
- **<main>** : Contient le contenu principal de la page.
- **<article>** : Représente une section indépendante de contenu, comme un article de blog, un commentaire, un forum, etc.
- **<aside>** : Contient du contenu secondaire, comme une barre latérale, des publicités, des informations complémentaires, etc.
- **<footer>** : Pied de page de la page ou d'une section. Contient généralement des informations de copyright, des liens vers les mentions légales, etc.

2. Rôle et importance

- **Accessibilité** : Les balises sémantiques aident les personnes utilisant des lecteurs d'écran à mieux comprendre la structure et le contenu de la page.
- **Référencement** : Les moteurs de recherche utilisent les balises sémantiques pour mieux comprendre le contenu et l'indexer correctement.
- **Maintenance** : Un code HTML sémantique est plus facile à lire, à comprendre et à maintenir.

Exercice :

Objectif : Mettre en pratique l'utilisation des balises sémantiques HTML5.

Consignes :

1. Reprendre le code HTML de l'exercice 2 ("Le chat sauvage").
2. Structurer la page avec les balises sémantiques suivantes :
 - <header> pour l'en-tête contenant le titre principal.
 - <main> pour le contenu principal.
 - <article> pour chaque section ("Habitat", "Alimentation", "Comportement").
 - <footer> pour le pied de page (vous pouvez ajouter des informations de copyright).

Exemple de code HTML (Le chat sauvage Ex2 - version sémantique) :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Le chat sauvage</title>
</head>
<body>

  <header>
    <h1>Le chat sauvage</h1>
  </header>

  <main>
    <article>
      <h2>Habitat</h2>
      <p>Le chat sauvage vit principalement dans les forêts ... <span class="mot-cle">denses</span> et ...</p>
    </article>

    <article>
      <h2>Alimentation</h2>
      <p>Le chat sauvage est un <span class="mot-cle">prédateur</span> carnivore qui se nourrit de ...</p>
    </article>

    <article>
      <h2>Comportement</h2>
      <p>Le chat sauvage est un animal <span class="mot-cle">solitaire</span> et <span class="mot-cle">territorial</span> ...</p>
    </article>
  </main>

  <footer>
    <p>&copy; 2023 - Mon site web</p>
  </footer>

</body>
</html>
```

Remarques :

- L'utilisation des balises sémantiques rend le code plus clair et plus lisible.
- Les moteurs de recherche et les lecteurs d'écran comprennent mieux la structure du contenu.
- N'hésitez pas à utiliser les balises sémantiques dans tous vos projets web.

9. NOTIONS AVANCÉES

Ce chapitre aborde des éléments HTML plus avancés qui permettent d'intégrer du contenu multimédia et d'autres pages web dans vos propres pages.

1. Iframes

- Une iframe (inline frame) permet d'intégrer une page web externe directement dans votre page.
- C'est comme une fenêtre qui affiche le contenu d'une autre page.
- **Balise <iframe> :**
 - src : URL de la page à afficher dans l'iframe.
 - width et height : dimensions de l'iframe.

Exemple :

```
<iframe src="https://www.example.com" width="800" height="600"></iframe>
```

2. Audio et vidéo

- HTML5 permet d'intégrer facilement des fichiers audio et vidéo sans avoir recours à des plugins externes.
- **Balise <audio> :**
 - src : URL du fichier audio.
 - controls : Affiche les contrôles de lecture (play, pause, volume, etc.).
- **Balise <video> :**
 - src : URL du fichier vidéo.
 - controls : Affiche les contrôles de lecture.
 - width et height : dimensions du lecteur vidéo.

Exemples :

```
<audio src="musique.mp3" controls></audio>
```

```
<video src="video.mp4" width="640" height="360" controls></video>
```

Exercice :

Objectif : Intégrer une vidéo YouTube dans une page web.

Consignes :

1. Aller sur YouTube et choisir une vidéo.
2. Cliquer sur "Partager" puis sur "Intégrer".
3. Copier le code iframe fourni.
4. Créer un nouveau fichier HTML et coller le code iframe dans le <body>.

Exemple de code iframe YouTube :

```
<iframe width="560" height="315" src="https://www.youtube.com/embed/VIDEO_ID" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share" allowfullscreen></iframe>
```

Remarques :

- Remplacez VIDEO_ID par l'identifiant de la vidéo YouTube.
- L'attribut allowfullscreen permet d'afficher la vidéo en plein écran.
- Vous pouvez ajuster la taille de l'iframe en modifiant les valeurs des attributs width et height.

Attention :

- L'utilisation d'iframes peut poser des problèmes de sécurité et de performance.
- Il est important de bien comprendre les implications avant de les utiliser.

10. LISTE DES BALISES HTML

La liste complète des balises HTML est disponible ici : <https://www.w3schools.blog/html-tags-list>

Balise	Utilisation
<!-- -->	Placer un commentaire dans le document HTML
<!DOCTYPE>	Spécifier la version d'HTML
A	
<a>	Créer un lien ou un hyperlien
<audio>	Intégrer un contenu sonore
B	
	Rendre un texte en gras
<blockquote>	Intégrer une citation
<body>	Spécifier le corps (contenu principal) du document HTML
 	Insérer un passage à la ligne
<button>	Insérer un bouton cliquable
C	
<caption>	Insérer une légende pour un tableau
D	
<div>	Insérer une division ou une section
E	
	Mettre en évidence un élément
F	
<fieldset>	Pour regrouper des éléments/étiquettes dans un formulaire.
<footer>	Spécifier le pied de page du document
<form>	Ajouter un formulaire
H	
<h1>...<h6>	Spécifier les titres d'un document HTML du niveau 1 au niveau 6
<head>	Contient les métadonnées du document
<header>	Spécifier la section d'en-tête d'un document HTML

<html>	La racine du document HTML (toute première balise)
I	
	Insérer une image
<input>	Insérer un champ de saisie dans un formulaire HTML
L	
<label>	Insérer une étiquette de texte pour un champ de saisie du formulaire
<legend>	Insérer une légende pour le contenu de <fieldset>
	Insérer un élément dans une liste
<link>	Créer une relation entre le document actuel et une ressource externe
M	
<main>	Insérer le contenu principal du document
<mark>	Mettre du texte en évidence
<meta>	Spécifier une métadonnée du document
N	
<nav>	Insérer une barre de navigation
O	
	Insérer une liste ordonnée
<optgroup>	Regrouper les options d'une liste déroulante
<option>	Spécifier des options ou des éléments d'une liste déroulante
P	
<p>	Insérer un paragraphe
S	
<section>	Insérer une section générique
<select>	Insérer une liste déroulante
<small>	Pour rendre un texte plus petit
<source>	Pour spécifier plusieurs ressources média pour différents éléments média tels que les éléments <video> et <audio>.
	Encapsuler une portion de texte afin de le styliser
	Marquer un texte comme important

<code><style></code>	Spécifier les informations de style du document
T	
<code><table></code>	Insérer un tableau
<code><tbody></code>	Contient le corps (contenu principal) d'un tableau
<code><td></code>	Représente une cellule d'un tableau
<code><textarea></code>	Insère un champ de texte libre
<code><tfoot></code>	Contient le pied de page d'un tableau
<code><th></code>	Marquer une ligne de cellules d'un tableau comme ligne d'en-tête
<code><thead></code>	Contient l'en-tête d'un tableau
<code><title></code>	Spécifier le titre d'un document
<code><tr></code>	Insérer une ligne de cellules dans un tableau
U	
<code></code>	Insérer une liste non-ordonnée
V	
<code><video></code>	Insérer une vidéo