

Mémento HTML

Première et terminale NSI

① SQUELETTE D'UNE PAGE

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <meta name="viewport"
      content="width=device-width">
    <title>Titre de l'onglet</title>
    <link rel="stylesheet" href="style.css">
    <script src="script.js" defer</script>
  </head>
  <body>
    <h1>Titre de la page</h1>
    <p>Premier paragraphe.</p>
  </body>
</html>
```

HTML décrit la **structure** et le **sens** du contenu ; CSS s'occupe de l'apparence, JavaScript du comportement. Une balise s'ouvre `<p>` et se ferme `</p>` ; certaines sont vides : `<br>`, `<img>`, `<meta>`, `<input>`. Les balises s'emboîtent sans se chevaucher.

② ÉLÉMENTS ET ATTRIBUTS

```
<a href="https://csilyon.fr" target="_blank">
  Lien
</a>

<p id="intro" class="important grand">
```

Attribut	Rôle
id="..."	identifiant unique dans la page
class="..."	une ou plusieurs classes, séparées par des espaces
href	cible d'un lien (a, link)
src	source d'une image ou d'un script
alt	texte de remplacement d'une image
title	info-bulle au survol
style="..."	CSS en ligne (à éviter)
lang="fr"	langue du contenu

Chemins : href="page2.html" (même dossier), href="img/a.png" (sous-dossier), href="../index.html" (dossier parent), href="#section" (ancrage vers un id).

③ STRUCTURE DU DOCUMENT

Balise	Rôle
<header>	en-tête de la page ou d'une section
<nav>	menu de navigation
<main>	contenu principal, unique
<section>	partie thématique
<article>	contenu autonome (billet, fiche)
<aside>	contenu annexe
<footer>	pied de page
<div>	bloc générique sans sens
	portion de texte générique

```
<body>
  <header><h1>Mon site</h1></header>
  <nav>
    <a href="index.html">Accueil</a>
    <a href="cours.html">Cours</a>
  </nav>
  <main>
    <section>
      <h2>Chapitre 1</h2>
      <p>Texte...</p>
    </section>
  </main>
  <footer>Contact : nsi@csilyon.fr</footer>
</body>
```

④ TEXTE

Balise	Rôle
<h1> à <h6>	titres, du plus important au moins important
<p>	paragraphe
 	retour à la ligne
<hr>	ligne de séparation
	importance forte (gras)
	mise en relief (italique)
<code>	code dans le texte
<pre>	texte préformaté (espaces conservés)
<blockquote>	citation
<sup>, <sub>	exposant, indice

Un seul `<h1>` par page, puis des `<h2>` et `<h3>` dans l'ordre, sans sauter de niveau.

⑤ LISTES ET TABLEAUX

7 MULTIMÉDIA ET INTÉGRATION

```
<ul>                                <!-- liste à puces -->
  <li>Python</li>
  <li>SQL</li>
</ul>
<ol>                                <!-- liste numérotée -->

  <li>Analyser</li>
  <li>Coder</li>
</ol>
```

```
<table>
  <thead>
    <tr><th>Nom</th><th>Note</th></tr>
  </thead>
  <tbody>
    <tr><td>Ada</td><td>18</td></tr>
    <tr><td>Alan</td><td>17</td></tr>
  </tbody>
</table>
```

tr est une ligne, th une cellule d'en-tête, td une cellule ordinaire. colspan="2" fusionne deux colonnes.

6 FORMULAIRES

```
<form action="/traitement" method="post">
  <label for="nom">Nom</label>
  <input type="text" id="nom" name="nom"
    required>
  <input type="number" name="age"
    min="0" max="120">
  <input type="password" name="mdp">
  <input type="checkbox" name="ok" value="1">
  <input type="radio" name="voie" value="G">
  <input type="radio" name="voie" value="T">
  <select name="classe">
    <option value="1">Première</option>
    <option value="T">Terminale</option>
  </select>
  <textarea name="msg" rows="3"></textarea>
  <button type="submit">Envoyer</button>
</form>
```

- name est le nom du champ transmis au serveur, value sa valeur.
- method="get" envoie les données dans l'URL (?nom=Ada&age=36) : visible, à réserver aux recherches. method="post" les envoie dans le corps de la requête : à utiliser pour les mots de passe et les modifications.
- label for relie une étiquette à un champ par son id.
- Autres types d'input : email, date, range, color, file, hidden.

```

<figure>
  

  <figcaption>Figure 1</figcaption>
</figure>
<audio controls src="son.mp3"></audio>
<video controls width="400" src="clip.mp4">
</video>
<iframe src="https://exemple.org" width="600"
  height="300"></iframe>
```

8 CARACTÈRES SPÉCIAUX ET COMMENTAIRES

Code	Affiche	Usage
<	<	chevron ouvrant
>	>	chevron fermant
&	&	esperluette
"	"	guillemet
 		espace insécable
é	é	accents (inutile en UTF-8)

```
<!-- Un commentaire, invisible pour le visiteur
-->
```

9 CLIENT, SERVEUR, NAVIGATEUR

- Le navigateur (client) envoie une **requête HTTP** (GET page.html) ; le serveur renvoie une **réponse** avec un code (200 OK, 404 introuvable, 500 erreur serveur) et le contenu.
- Une URL se lit https://site.fr/dossier/page.html?cle=valeur#ancree : protocole, domaine, chemin, paramètres, ancre.
- Le HTML, le CSS et le JavaScript s'exécutent côté client ; Python (Flask) ou PHP s'exécutent côté serveur et fabriquent la page.
- Outils de développement du navigateur (touche F12) : inspecter le code, la console, les requêtes réseau.

10 BONNES PRATIQUES

- Indenter le code d'un niveau à chaque emboîtement ; fermer chaque balise ouverte.
- Toujours renseigner alt sur les images et lang sur <html>.
- Utiliser les balises selon leur sens (nav, main, strong) plutôt que div et span partout.
- Séparer les fichiers : index.html, style.css, script.js.
- Valider la page sur validator.w3.org ; consulter la référence developer.mozilla.org (MDN).