

Chapitre 1 : HTML - Approfondissement

1. Définition détaillée du HTML

HTML (HyperText Markup Language) est le langage fondamental du web. Plus qu'un simple langage de balisage, il permet de :

- Structurer sémantiquement le contenu (titres, paragraphes, listes...)
- Créer des liens hypertextes entre documents
- Intégrer des médias (images, vidéos)
- Préparer l'interactivité via des formulaires

Exemple vivant :

Toute page web que vous consultez (articles, réseaux sociaux, boutiques en ligne) utilise HTML comme squelette de base.

2. Structure d'une page HTML - Approfondi

Anatomie complète :

```
<!DOCTYPE html> <!-- Déclaration du type de document -->
<html lang="fr"> <!-- Racine du document + langue -->
<head> <!-- Métadonnées (non visibles) -->
  <meta charset="UTF-8"> <!-- Encodage caractères -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Titre visible dans l'onglet</title> <!-- Important pour le SEO -->
  <link rel="stylesheet" href="styles.css"> <!-- Lien vers CSS -->
</head>
<body> <!-- Contenu visible -->
  <!-- Contenu ici -->
  <script src="script.js"></script> <!-- JavaScript à la fin -->
</body>
</html>
```

Bonnes pratiques :

- Toujours inclure `<!DOCTYPE html>`
 - Spécifier la langue (`lang="fr"`)
 - Utiliser `meta charset="UTF-8"` pour les caractères spéciaux
 - Placer les scripts JavaScript avant la fermeture de `</body>`
-

3. Les balises HTML - Guide complet

Balises essentielles :

Structure :

```
<header>En-tête (logo, navigation)</header>
<nav>Liens de navigation</nav>
<main>Contenu principal</main>
<section>Section thématique</section>
<article>Contenu autonome (article de blog)</article>
<aside>Contenu complémentaire</aside>
<footer>Pied de page</footer>
```

Contenu :

```
<h1>Titre principal</h1> <!-- Un seul par page -->
<h2>Sous-titre</h2> <!-- Jusqu'à h6 -->
<p>Paragraphe avec <strong>mise en valeur</strong> et <em>emphase</em>. </p>
<a href="https://exemple.com" target="_blank">Lien (ouvre nouvel onglet)</a>

```

Listes :

```
<!-- Liste non ordonnée -->
<ul>
  <li>Élément 1</li>
  <li>Élément 2</li>
</ul>

<!-- Liste ordonnée -->
<ol>
  <li>Premier</li>
  <li>Second</li>
</ol>
```

4. Création de page web - Pratique guidée

Projet pas à pas :

1. Initialisation :

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="UTF-8">
    <title>Ma première page complète</title>
  </head>
  <body>
    <header>
      <h1>Bienvenue sur mon site</h1>
    </header>
```

```
</body>
</html>
```

2. Ajout de contenu :

```
<main>
  <section>
    <h2>À propos</h2>
    <p>Je m'appelle <strong>Jean</strong> et j'apprends le HTML. </p>
    
  </section>
</main>
```

3. Ajout de navigation :

```
<nav>
  <ul>
    <li><a href="#apropos">À propos</a></li>
    <li><a href="#contact">Contact</a></li>
  </ul>
</nav>
```

5. Création de tableaux - Techniques avancées

Tableau complexe avec fusion de cellules :

```
<table border="1">
  <caption>Emploi du temps</caption>
  <thead>
    <tr>
      <th rowspan="2">Heure</th>
      <th colspan="5">Jours</th>
    </tr>
    <tr>
      <th>Lundi </th>
      <th>Mardi </th>
      <th>Mercredi </th>
      <th>Jeudi </th>
      <th>Vendredi </th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>9h-11h</td>
      <td>HTML</td>
      <td>CSS</td>
      <td rowspan="2">Projet</td>
      <td>JavaScript</td>
      <td>Python</td>
    </tr>
    <tr>
      <td>11h-13h</td>
      <td>Exercices</td>
      <td>Révision</td>
    </tr>
  </tbody>
</table>
```

```

        <td>Base de données</td>
        <td>Algorithme</td>
    </tr>
</tbody>
</table>

```

Accessibilité :

- Toujours utiliser `<thead>`, `<tbody>`, `<tfoot>`
- Ajouter `<caption>` pour décrire le tableau
- Utiliser `scope="col"` ou `scope="row"` pour les en-têtes

6. Création de formulaires - Guide expert

Formulaire complet avec validation :

```

<form action="/traitement" method="post" novalidate>
    <fieldset>
        <legend>Informations personnelles</legend>

        <div>
            <label for="nom">Nom complet* :</label>
            <input type="text" id="nom" name="nom" required
                minlength="3" maxlength="50"
                pattern="[A-Za-zÀ-ÿ\s-]+">
        </div>

        <div>
            <label for="email">Email* :</label>
            <input type="email" id="email" name="email" required>
        </div>
    </fieldset>

    <fieldset>
        <legend>Préférences</legend>

        <div>
            <label for="newsletter">
                <input type="checkbox" id="newsletter" name="newsletter">
                S'abonner à la newsletter
            </label>
        </div>

        <div>
            <label>Langue préférée :</label>
            <select name="langue">
                <option value="fr">Français</option>
                <option value="en">Anglais</option>
                <option value="es">Espagnol</option>
            </select>
        </div>
    </fieldset>
</form>

```

```
<button type="submit">Envoyer</button>
<button type="reset">Réinitialiser</button>
</form>
```

Bonnes pratiques :

- Regrouper les champs avec `<fieldset>`
- Ajouter des légendes avec `<legend>`
- Utiliser `required`, `minlength`, `pattern` pour la validation
- Associer chaque label à son input avec `for` et `id`

Exercices approfondis avec corrections

Exercice 1 : Page complète

Consigne : Créez une page "CV" contenant :

1. Un en-tête avec votre nom et titre professionnel
2. Une section "Compétences" avec liste à puces
3. Une section "Expérience" avec tableau chronologique
4. Un formulaire de contact avec validation

Correction :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Mon CV Professionnel</title>
  <style>
    body { font-family: Arial, sans-serif; line-height: 1.6; max-width: 800px;
margin: 0 auto; padding: 20px; }
    header { text-align: center; margin-bottom: 30px; }
    h1 { color: #2c3e50; }
    section { margin-bottom: 30px; }
    table { width: 100%; border-collapse: collapse; }
    th, td { padding: 10px; border: 1px solid #ddd; text-align: left; }
    form div { margin-bottom: 15px; }
    label { display: inline-block; width: 150px; }
    input[type="text"], input[type="email"], textarea { width: 300px; padding:
8px; }
  </style>
</head>
<body>
  <header>
    <h1>Jean Dupont</h1>
    <p>Développeur Web Junior</p>
  </header>
```

```

<section id="competences">
  <h2>Compétences</h2>
  <ul>
    <li>HTML5 et CSS3 avancé</li>
    <li>JavaScript (ES6+)</li>
    <li>Python avec Flask</li>
    <li>Git et GitHub</li>
    <li>Responsive Design</li>
  </ul>
</section>

<section id="experience">
  <h2>Expérience Professionnelle</h2>
  <table>
    <thead>
      <tr>
        <th>Période</th>
        <th>Poste</th>
        <th>Entreprise</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>2022-2023</td>
        <td>Stagiaire Développeur</td>
        <td>WebTech Solutions</td>
      </tr>
      <tr>
        <td>2021-2022</td>
        <td>Assistant Informatique</td>
        <td>Services Généraux</td>
      </tr>
    </tbody>
  </table>
</section>

<section id="contact">
  <h2>Contactez-moi </h2>
  <form>
    <div>
      <label for="nom_contact">Nom :</label>
      <input type="text" id="nom_contact" name="nom" required
mi nlength="2">
    </div>
    <div>
      <label for="email_contact">Email :</label>
      <input type="email" id="email_contact" name="email" required>
    </div>
    <div>
      <label for="message">Message :</label>
      <textarea id="message" name="message" rows="5" required></textarea>
    </div>
    <button type="submit">Envoyer</button>
  </form>

```

```
</section>
</body>
</html>
```

Exercice 2 : Tableau complexe

Consigne : Créez un tableau de résultats scolaires avec :

1. Fusion de colonnes pour les matières
2. Fusion de lignes pour les semestres
3. Note moyenne calculée

Correction :

```
<table border="1">
  <caption>Résultats Scolaires - Année 2023/2024</caption>
  <thead>
    <tr>
      <th rowspan="2">Semestre</th>
      <th colspan="3">Matières</th>
      <th rowspan="2">Moyenne</th>
    </tr>
    <tr>
      <th>Mathématiques</th>
      <th>Français</th>
      <th>Informatique</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td rowspan="2">Semestre 1</td>
      <td>15</td>
      <td>12</td>
      <td>14</td>
      <td>13.67</td>
    </tr>
    <tr>
      <td>16</td>
      <td>13</td>
      <td>15</td>
      <td>14.67</td>
    </tr>
    <tr>
      <td>Semestre 2</td>
      <td>17</td>
      <td>14</td>
      <td>16</td>
      <td>15.67</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <th colspan="4">Moyenne Générale</th>
      <td>14.67</td>
    </tr>
  </tfoot>
</table>
```

```

        </tr>
    </tfoot>
</table>

```

Exercice 3 : Formulaire interactif

Consigne : Créez un formulaire d'inscription avec :

1. Validation en temps réel avec JavaScript
2. Messages d'erreur personnalisés
3. Prévisualisation de l'avatar

Correction :

```

<!DOCTYPE html >
<html >
<head>
    <title>Inscription</title>
    <style>
        .error { color: red; font-size: 0.8em; }
        .success { color: green; }
        #avatarPreview { width: 100px; height: 100px; border: 1px dashed #ccc;
margin: 10px 0; }
    </style>
</head>
<body>
    <h1>Inscription</h1>
    <form id="inscriptionForm">
        <div>
            <label for="pseudo">Pseudo* :</label>
            <input type="text" id="pseudo" name="pseudo" required minlength="4">
            <span id="pseudoError" class="error"></span>
        </div>

        <div>
            <label for="mdp">Mot de passe* :</label>
            <input type="password" id="mdp" name="mdp" required
pattern="(?!.*\d)(?!.*[a-z])(?!.*[A-Z]).{8,}">
            <span id="mdpError" class="error"></span>
            <small>8 caractères min, avec majuscule, minuscule et chiffre</small>
        </div>

        <div>
            <label for="avatar">Avatar :</label>
            <input type="file" id="avatar" name="avatar" accept="image/*">
            <div id="avatarPreview"></div>
        </div>

        <button type="submit">S'inscrire</button>
    </form>

    <script>
        document.getElementById('pseudo').addEventListener('input', function() {
            const pseudo = this.value;

```



```

        const errorSpan = document.getElementById('pseudoError');

        if (pseudo.length < 4) {
            errorSpan.textContent = "Le pseudo doit faire au moins 4 caractères";
        } else {
            errorSpan.textContent = "";
        }
    });

    document.getElementById('mdp').addEventListener('input', function() {
        const mdp = this.value;
        const errorSpan = document.getElementById('mdpError');
        const regex = /^(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}$/;

        if (!regex.test(mdp)) {
            errorSpan.textContent = "Le mot de passe ne respecte pas les
critères";
        } else {
            errorSpan.textContent = "";
        }
    });

    document.getElementById('avatar').addEventListener('change', function(e) {
        const file = e.target.files[0];
        if (file) {
            const reader = new FileReader();
            reader.onload = function(event) {
                const preview = document.getElementById('avatarPreview');
                preview.innerHTML = ``;
            }
            reader.readAsDataURL(file);
        }
    });

    document.getElementById('inscriptionForm').addEventListener('submit',
function(e) {
    e.preventDefault();
    alert('Formulaire validé ! (Envoi simulé)');
});
</script>
</body>
</html>

```

Bonnes pratiques HTML professionnelles

1. Sémantique :

- Utilisez les balises selon leur sens (<article>, <section>)
- Évitez les <div> quand une balise sémantique existe

2. Accessibilité :

- Ajoutez toujours des `alt` aux images
 - Utilisez `aria-label` pour les éléments non textuels
 - Structurez vos titres logiquement (`h1` > `h2` > `h3`)
3. Validation :
- Vérifiez votre code avec le validateur W3C
 - Testez sur plusieurs navigateurs
4. Performance :
- Minimisez la nesting (imbrication) des balises
 - Utilisez les balises natives plutôt que des solutions JavaScript complexes
5. SEO :
- Structurez bien vos titres
 - Utilisez des balises `<meta>` descriptives
 - Privilégiez le HTML sémantique
-

Ressources pour aller plus loin

6. Références :
- [Documentation MDN HTML](#)
 - [HTML Living Standard](#)
7. Outils :
- Valideur HTML : validator.w3.org
 - Éditeur en ligne : [CodePen](#) ou [JSFiddle](#)
8. Tutoriels avancés :
- HTML5 Canvas et SVG
 - Web Components
 - Microdonnées et Rich Snippets

Ce guide approfondi couvre l'essentiel du HTML moderne. La clé pour maîtriser le HTML est la pratique régulière et l'expérimentation avec différents types de contenus et structures.