

HTML

HTML

Qu'est-ce que HTML ?

- HTML signifie HyperText Markup Language.
- C'est le langage de balisage standard pour créer des pages web.
- HTML décrit la structure d'une page web à l'aide de balises.

Structure de base d'un document HTML

Un document HTML se compose des parties suivantes :

1. Déclaration `<!DOCTYPE html>`
2. Élément `<html>` (élément racine)
3. Élément `<head>` (contient les métadonnées)
4. Élément `<title>` (définit le titre de la page)
5. Élément `<body>` (contient le contenu visible)

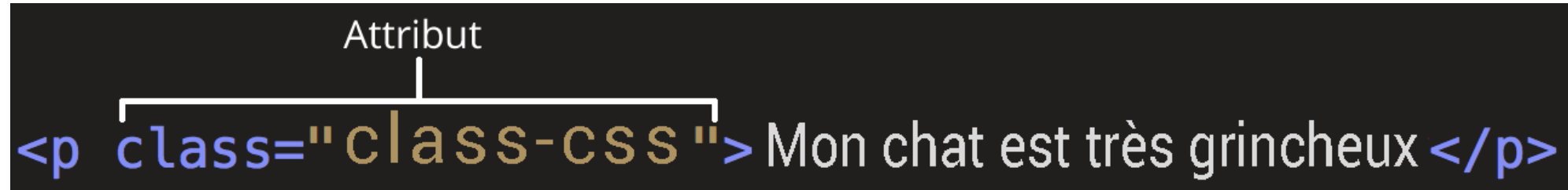
Anatomie d'une balise HTML

La balise ouvrante, suivie par le contenu puis par la balise fermante, composent ensemble un **élément HTML**.



Attributs

Les attributs contiennent des informations supplémentaires sur l'élément sans qu'elles n'apparaissent dans le contenu réel.



```
<p class="class-css"> Mon chat est très grincheux </p>
```

l'attribut class permet de donner à l'élément un nom d'identification qui peut ensuite être utilisé pour cibler l'élément afin de lui attribuer un style CSS

Balises HTML courantes

- Les balises sont utilisées pour définir les éléments dans un document HTML.
- `<h1>` à `<h6>` (titres)
- `<p>` (paragraphe)
- `<a>` (lien)
- `` (image)
- `` et `` (liste non ordonnée)
- `` et `` (liste ordonnée)

Page HTML simple

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ma première page HTML</title>
  </head>
  <body>
    <h1>Titre principal</h1>
    <p>Ceci est un paragraphe.</p>
    <a href="https://www.example.com">Lien vers Example.com</a>
    
  </body>
</html>
```


<!DOCTYPE html>

- C'est le premier élément présent dans toute page HTML
- Cela peut ressembler à un élément HTML, mais ce n'est pas le cas. Il s'agit d'un type spécial de nœud appelé "doctype"
- Le doctype indique au navigateur d'utiliser le mode standard
- Sans cette base, le navigateur utilise le mode [quirks](#)

<html>

- L'élément `<html>` est l'élément racine d'un document HTML.
- Il s'agit du parent de `<head>` et de `<body>`.
- Il contient tout le contenu du document HTML à l'exception du doctype.
- Si cet attribut est omis, il sera implicite, mais il est important de l'inclure, car il s'agit de l'élément sur lequel la langue du contenu du document est déclarée.

L'attribut lang

- L'attribut de langue lang ajouté à la balise `<html>` définit la langue principale du document.
- La valeur de l'attribut lang correspond à un code de langue ISO à deux ou trois lettres suivi de la région.
- La région est facultative, mais recommandée, car la langue peut varier considérablement d'une région à l'autre.

<head>

- L'en-tête `<head>`, ou l'en-tête des métadonnées du document, contient toutes les métadonnées d'un site ou d'une application.
- Le corps `<body>` comporte le contenu visible.
- Le reste de cette section se concentre sur les composants imbriqués dans les `<head></head>` d'ouverture et de fermeture.

Les composants requis de <head>

- Encodage des caractères

```
<meta charset="utf-8" />
```

- Titre du document

```
<title>Utopios Solutions</title>
```

- Rendre le site responsive, en commençant par faire de la largeur du contenu la largeur de l'écran

```
<meta name="viewport" content="width=device-width" />
```

Le CSS

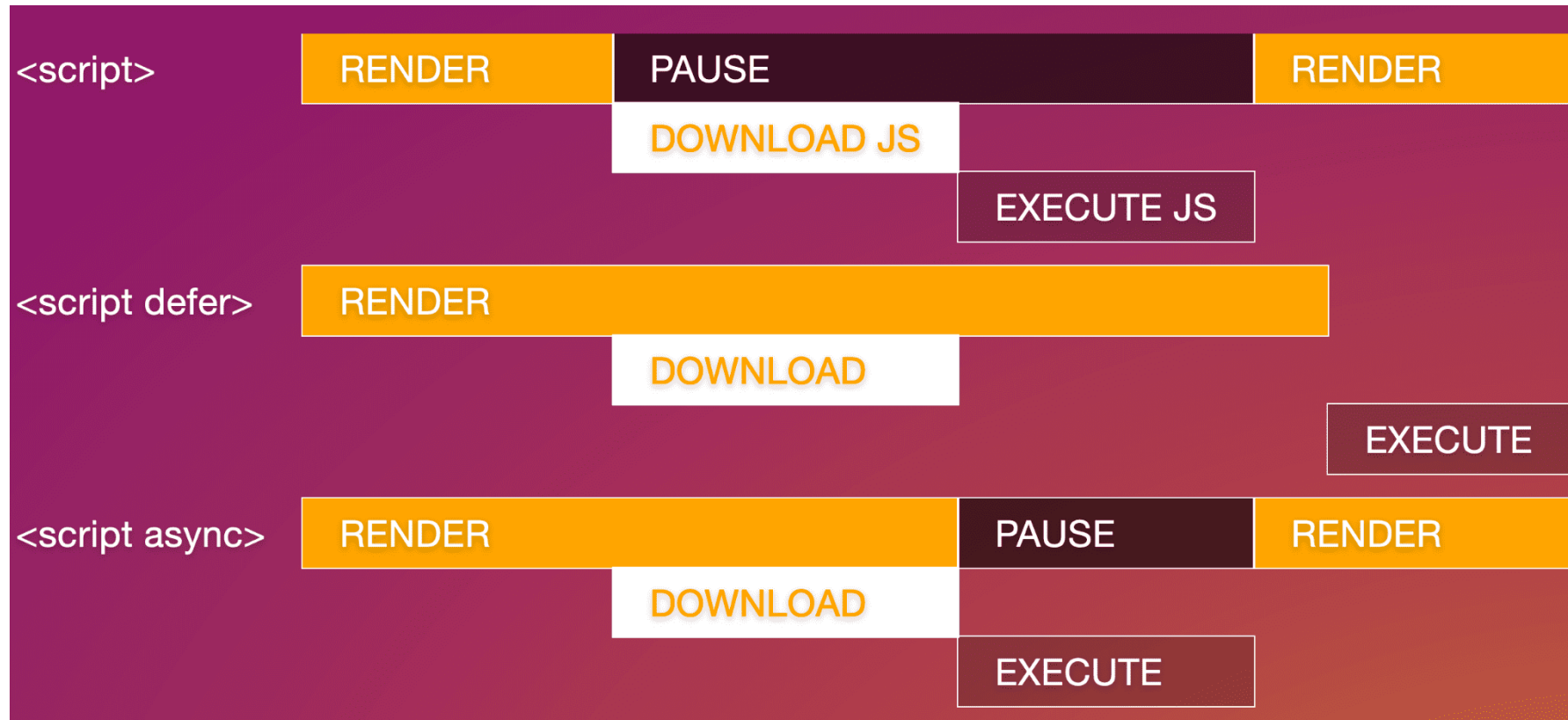
- On inclut généralement le CSS à l'aide de la balise `<link>`
- L'association d'une seule feuille de style externe ou de plusieurs feuilles de style externes est bénéfique à la fois pour l'expérience des développeurs et les performances du site
 - Le style est centralisé
 - Les navigateurs mettent en cache les fichiers

```
<link rel="stylesheet" href="styles.css" />
```

<script>

- La balise `<script>` permet d'inclure des scripts
- Le type `type="module"` permet d'ajouter un module JavaScript
- Deux attributs peuvent réduire le caractère bloquant du téléchargement et de l'exécution de JavaScript: `defer` et `async`

Sync, async et defer



Commentaires HTML

- Les commentaires HTML peuvent être placés n'importe où sur la page, y compris dans l'en-tête ou le corps, à l'exception des scripts et des blocs de style, où vous devez utiliser respectivement les commentaires JavaScript et CSS.
- Aucun élément compris entre `<!--` et `-->` n'est visible ni analysé.
- Les commentaires HTML sont visibles sur la page lorsque l'on inspecte son code

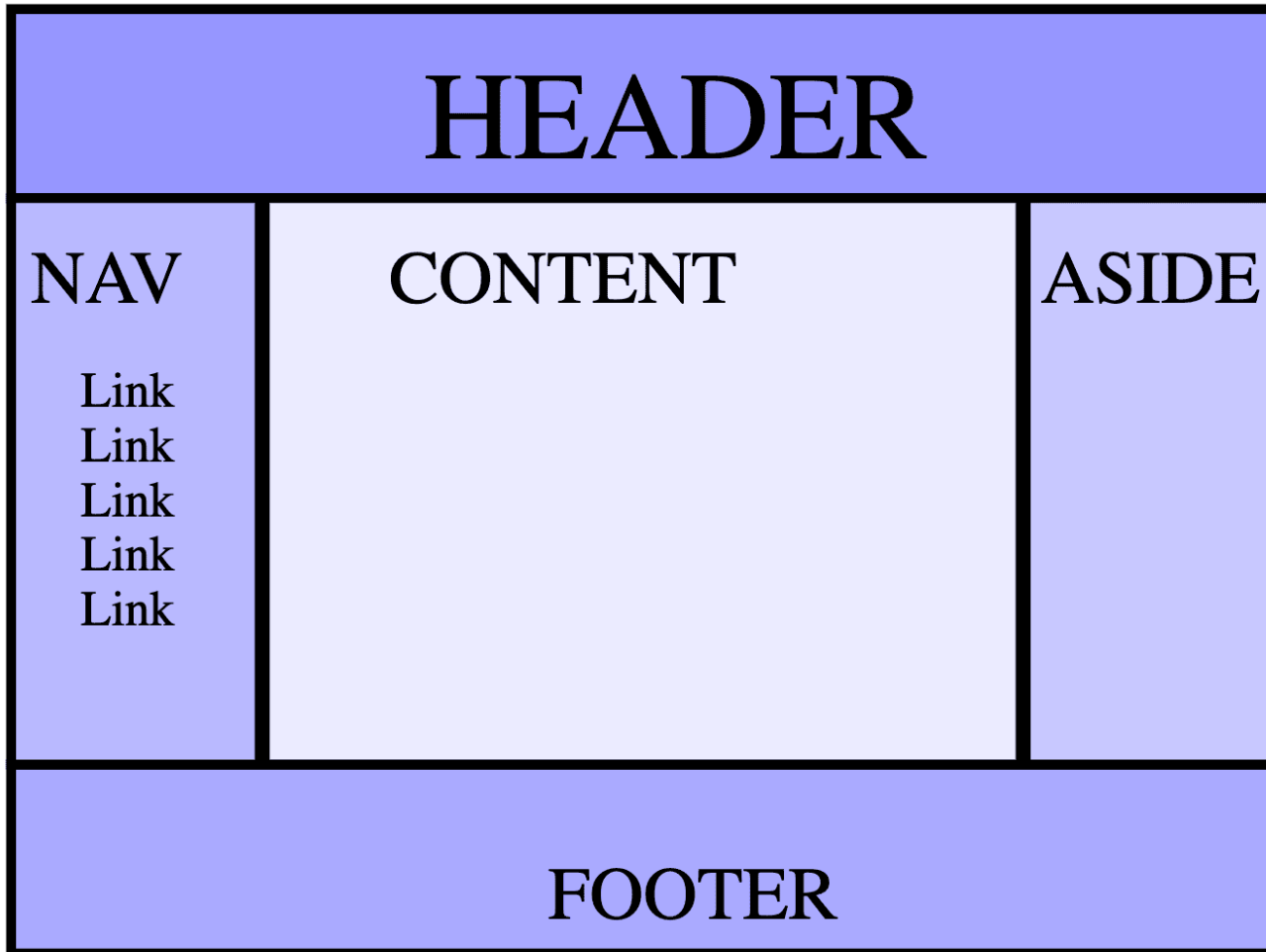
Qu'est-ce qu'une balise sémantique ?

- Sémantique signifie "**lié au sens**". Écrire du code HTML sémantique signifie utiliser des éléments HTML pour structurer votre contenu en fonction de la signification de chaque élément, et non de son apparence.
- Elles aident les moteurs de recherche et les lecteurs d'écran à comprendre le contenu.

Principales balises sémantiques

1. `<header>` : En-tête de la page (logo, titre, menu)
2. `<nav>` : Menu de navigation
3. `<main>` : Contenu principal de la page
4. `<article>` : Contenu indépendant (article, blog, commentaire)
5. `<section>` : Section thématique
6. `<aside>` : Contenu complémentaire (barre latérale, publicité)
7. `<footer>` : Pied de page

Mise en page du Saint Graal



Les attributs

Attributs booléens

- Si un attribut booléen est présent, la valeur est toujours "true"
- Les attributs booléens incluent autofocus, inert, checked, disabled, required, reversed, allowfullscreen, default, loop, autoplay, controls, muted, readonly, multiple, et selected.

```
<input required />
```

Attributs énumérés

- Ce sont des attributs HTML qui ont un ensemble limité de valeurs valides prédéfinies.
- Dans la plupart des cas avec des attributs énumérés, les valeurs manquantes et non valides sont les mêmes.
- Par exemple, si l'attribut type d'un élément `<input>` est manquant, présent mais sans valeur, ou s'il a une valeur non valide, il est défini par défaut sur `text`.

Attributs globaux

Les attributs globaux peuvent être définis pour n'importe quel élément HTML.

- **id**: L'attribut global `id` permet de définir un identifiant unique pour un élément
- **class**: L'attribut `class` offre un moyen supplémentaire de cibler des éléments avec CSS (et JavaScript), mais ne remplit aucune autre fonction en HTML. La valeur de l'attribut de classe est une liste des classes sensibles à la casse de l'élément, séparées par des espaces.

Attributs globaux

- **style**: L'attribut `style` permet d'appliquer des styles à l'élément unique sur lequel l'attribut est défini. Son utilisation n'est pas recommandée.
- **tabindex**: L'attribut `tabindex` peut être ajouté à n'importe quel élément pour qu'il soit sélectionné.
- **role**: L'attribut `role` peut être utilisé pour donner une signification sémantique au contenu, ce qui permet aux lecteurs d'écran d'informer les utilisateurs du site de l'interaction utilisateur attendue pour un objet.

Attributs personnalisés

- Vous pouvez créer les attributs personnalisés de votre choix en ajoutant le préfixe `data-`
- Les attributs personnalisés en HTML permettent aux développeurs de stocker des informations supplémentaires sur des éléments HTML

```
<div id="user-profile" data-user-id="12345">John Doe</div>
```

Balisage du texte

Titres

- Il y a six éléments de titre de section: `<h1>` ... `<h6>`
- Chacun représente l'un des six niveaux de titre de section, `<h1>` étant le niveau de section le plus élevé ou le plus important, et `<h6>` le niveau le plus bas.

Paragraphes

- En dehors des titres, la plupart des textes structurés sont composés d'une série de paragraphes.
- En HTML, les paragraphes sont balisés avec la balise `<p>`.
- La balise de fermeture est facultative, mais elle est toujours recommandée.

Citations

Lorsque vous balisez un article ou un article de blog, vous pouvez inclure une citation ou une citation, avec ou sans citation visible:

- `<blockquote>` : indiquer une citation longue
- `<q>` : indiquer une courte citation en ligne
- `<cite>` : indiquer le titre d'une œuvre créative

Exemple de code

- Pour documenter du code il est possible d'utiliser la balise `<code>`
- Par défaut, le contenu textuel est affiché en police à chasse fixe
- Pour inclure plusieurs lignes de codes, on englobe la base code de la balise `<pre>`

```
<code>push()</code>
```

Mise en valeur du texte

- `` : mettre en avant une partie du contenu.
- `<mark>` : identifier ou mettre en surbrillance le texte qui est pertinent.
- `` : identifie le texte comme ayant une grande importance.
- `
` : permet de faire un retour à la ligne.
- `<hr>` : ligne horizontale, utilisé comme séparateur de contenu.

Liens

Présentation

- La balise d'ancrage `<a>` et l'attribut `href` créent un lien hypertexte.
- Les liens sont la colonne vertébrale d'Internet.
- Ils représentent une connexion entre deux ressources, dont l'une est le document actuel.
- Les liens peuvent être créés par `<a>`, `<area>`, `<form>` et `<link>`.

Attribut href

- l'attribut href figure dans presque toutes les balises `<a>`
- L'attribut href permet de créer des liens hypertextes vers des emplacements de la page active, d'autres pages d'un site ou d'autres sites
- L'attribut href peut commencer par `mailto:` ou `tel:` pour envoyer des e-mails ou passer des appels.

Contexte de navigation

- L'attribut `target` permet de définir le contexte de navigation pour la navigation par lien
- Il peut prendre 4 valeurs différentes:
 1. `_self`: ouvre la fenêtre actuelle
 2. `_blank`: ouvre dans un nouvel onglet
 3. `_parent`: ouvre dans le parent si le lien est imbriqué
 4. `_top`: ouvre dans l'ancêtre actuel

Listes

Les types de listes

HTML propose plusieurs façons de baliser les listes :

- Il existe des listes numérotées (``), des listes à puces (``) et des listes de descriptions (`<dl>`).
- Les éléments de liste (``) sont imbriqués dans des listes numérotées et des listes à puces.
- Dans une liste de descriptions, vous trouverez les termes descriptifs (`<dt>`) et les détails de la description. `<dd>`.

Listes à puces

- L'élément `` est l'élément parent des listes d'éléments non ordonnées.
- Les seuls enfants d'un élément `` sont un ou plusieurs éléments d'élément de liste ``.

```
<ul>  
  <li>Blender</li>  
  <li>Toaster</li>  
  <li>Vacuum</li>  
</ul>
```

Listes ordonnées

- L'élément `` est l'élément parent des listes d'éléments ordonnées.
- Les seuls enfants d'un élément `` sont un ou plusieurs éléments `` ou des éléments de liste
- L'attribut `type` définit le type de numérotation

```
<ol type="A">  
  <li>Blender</li>  
  <li>Toaster</li>  
  <li>Vacuum</li>  
</ol>
```


Listes de description

Une liste de descriptions (`<dl>`) contient une série de termes de description (`<dt>`) et leurs informations de description (`<dd>`)

```
<dl>
  <dt>Blendan Smooth</dt>
  <dd>Originally a margarita maker, they are now an aspiring load balancer.</dd>
  <dt>Toasty McToastface</dt>
  <dd>
    Formerly partially to fully baked, they are now an aspiring nuclear codes
    handler.
  </dd>
</dl>
```

Les tables

Description

- Les tableaux HTML sont utilisés pour afficher des données tabulaires avec des lignes et des colonnes.
- La décision d'utiliser un `<table>` doit être basée sur le contenu que vous présentez et sur les besoins de vos utilisateurs par rapport à ce contenu.
- Si des données sont présentées, comparées, triées, calculées ou croisées, `<table>` est probablement le bon choix.

Éléments d'un tableau

1. La balise `<table>` encapsule le contenu du tableau
2. On trouve ensuite les en-têtes de tableau (`<thead>`), les corps de tableau (`<tbody>`) et, éventuellement, les pieds de page (`<tfoot>`)
3. Chacun de ces éléments est composé de lignes du tableau (`<tr>`). Les lignes contiennent des en-têtes de tableau (`<th>`) et des cellules de données (`<td>`)

Elements complémentaires

- Dans le DOM, avant tout cela, vous pouvez trouver deux fonctionnalités supplémentaires: la légende de la table (`<caption>`) et les groupes de colonnes (`<colgroup>`).
- Selon que `<colgroup>` comporte ou non un attribut `span`, il peut contenir des éléments de colonne de tableau (`<col>`) imbriqués.

Exemple de tableau

```
<table>
  <tbody>
    <tr>
      <th>Hal Gibrah</th>
      <td>Calculator</td>
      <td>2020</td>
    </tr>
    <tr>
      <th>Cathy Terr</th>
      <td>Waste disposal</td>
      <td>2018</td>
    </tr>
  </tbody>
</table>
```

Formulaires

Description

- L'élément `<form>` HTML identifie un point de repère de document contenant des commandes interactives pour l'envoi d'informations.
- Imbriqués dans un `<form>`, vous trouverez toutes les commandes de formulaire interactives (et non interactives) qui le composent.

Envoie de formulaire

- Les formulaires sont envoyés lorsque l'utilisateur active un bouton d'envoi imbriqué dans le formulaire.
- Il existe deux façons d'écrire un bouton d'envoi:

```
<input type="submit" value="Submit Form" />  
<button type="submit">Submit Form</button>
```

Attributs de formulaire

- Les valeurs des attributs `action` et `method` (**GET**, **POST**) de l'élément `<form>` définissent respectivement l'URL qui traite les données du formulaire et la méthode HTTP utilisée pour les envoyer.
- Par défaut, les données du formulaire sont envoyées à la page active.
- Les données envoyées sont constituées des paires nom/valeur des différentes commandes de formulaire du formulaire.

Attribut method

- Avec `POST`, les données sont ajoutées **au corps de la requête HTTP**. Lorsque vous envoyez des données sécurisées, telles que des mots de passe et des informations de carte de crédit, utilisez toujours `POST`.
- Avec `GET` les données sont transmises dans l'URL précisé par l'attribut `action`. On les appelle les **query parameters**. Ex:
`?name=toto&id=123`

Valeurs d'un formulaire

- Lors de l'envoi d'un formulaire, les informations des champs sont transmises sous formes de clé/valeur
- Les clés correspondent à l'attribut `name` dans le champs
- Les valeurs correspondent à l'attribut `value` ou la valeur saisie par l'utilisateur

<input>

- L'élément HTML `<input>` est utilisé pour créer un contrôle interactif dans un formulaire web qui pour la saisie des données
- La saisie et le comportement dépend de l'attribut `type` ([22 formes](#)):
 - `checkbox`: case à cocher
 - `email`: comme texte avec validation
 - `file`: sélection de fichiers
 - `password`: champ masqué
 - `radio`: sélection de valeur

Libellés

- Chaque champ de formulaire doit être associée à un libellé.
- Pour associer explicitement un champ de formulaire à un `<label>`, ajoutez l'attribut `for` au `<label>`: la valeur correspond à l'id du champ de formulaire auquel il est associé.

```
<label for="full_name">Your name</label>  
<input type="text" id="full_name" name="name" />
```

Ensemble de champs

L'élément HTML `<fieldset>` permet de regrouper une partie d'un formulaire HTML et d'associer une légende (`<legend>`) décrivant ce groupe

```
<form action="/subscribe" method="post">
  <fieldset>
    <legend>Subscribe to the Newsletter</legend>
    <input type="email" name="email" />
    <button>Ok</button>
  </fieldset>
</form>
```

Exemple de formulaire

```
<form method="GET">
  <label for="student">Pick a student:</label>
  <select name="student" id="student">
    <option value="hoover">Hoover Sukhdeep</option>
    <option>Blendan Smooth</option>
    <option value="toasty">Toasty McToastface</option>
  </select>
  <input type="submit" value="Submit Form" />
</form>
```


Les images

Description

- Les images décoratives, telles que les dégradés d'arrière-plan sur des boutons ou les images de fond sur des sections de contenu ou sur la page entière, sont des images de présentation et doivent être **appliquées en CSS**.
- Lorsqu'une image ajoute du contexte à un document, **il s'agit d'un contenu** qui doit être intégré au code HTML.

L'élément `img`

La principale méthode pour inclure des images consiste à utiliser la balise `` avec l'attribut `src` faisant référence à une ressource image et l'attribut `alt` décrivant l'image.

```

```

Attribut srcset

L'attribut `srcset` sur `` et l'élément `<picture>` permettent d'inclure plusieurs sources d'images avec les requêtes média associées, chacune avec une source d'image de remplacement.

```

```

Picture et source

- L'élément `<picture>` est un élément conteneur pour plusieurs options d'image répertoriées dans un nombre illimité d'éléments `<source>` et un seul élément `` obligatoire.
- Le navigateur examinera chaque élément `<source>` enfant et choisira la meilleure correspondance.

```
<picture>  
  <source src="images/eve.png" media="(max-width: 800px)" />  
  <source src="images/eve-xl.jpg" />  
    
</picture>
```

Figure et figcaption

- L'élément `<figure>` et son élément `<figcaption>` imbriqués permettent d'inclure une image avec une description associée.
- Lorsque vous incluez un `<figcaption>`, assurez-vous qu'il s'agit du premier ou du dernier enfant imbriqué dans `<figure>`.

```
<figure>
  
  <figcaption>An elephant at sunset</figcaption>
</figure>
```

Optimisation d'images

- La balise `` peut inclure des attributs `height` et `width` pour définir le format de l'image et réduire le décalage de la mise en page du contenu
- La valeur `lazy` sur l'attribut `loading` active le chargement différé de l'image

```

```

Détails, résumés et boîte de dialogue

Définition de détails

- L'élément `<summary>` combiné avec `<details>` permet de masquer et d'afficher du contenu
- Souvent utilisée pour créer des accordéons ou des FAQ, il n'est pris en compte par les navigateurs depuis janvier 2020
- Il s'agit d'un moyen intégré de gérer l'expansion et le repli du contenu

Exemple de détail

```
<details>
  <summary>
    I have keys but no doors. I have space but no room. You can enter but can't
    leave. What am I?
  </summary>
  <p>A keyboard.</p>
</details>
```

Boîte de dialogue

- Une boîte de dialogue modale est un type particulier de fenêtre pop-up sur une page Web: un pop-up qui interrompt l'utilisateur pour qu'il se concentre sur cet élément.
- Les boîtes de dialogue peuvent être **modales** (seul le contenu de la boîte de dialogue peut être interagi) ou **non modales** (il est toujours possible d'interagir avec le contenu en dehors de la boîte de dialogue)
- L'élément HTML `<dialog>` permet de créer une boîte de dialogue

Exemple de boîte de dialogue

```
<dialog open>  
  <p>Greetings, one and all!</p>  
  <form method="dialog">  
    <button>OK</button>  
  </form>  
</dialog>
```

Template, Slot, shadow DOM

Template

L'élément HTML `<template>` est un mécanisme utilisé pour stocker du contenu HTML qui ne doit pas être affiché lors du chargement de la page mais qui peut être instancié et affiché par la suite grâce à un script JavaScript.

```
<template id="productrow">
  <tr>
    <td class="record"></td>
    <td></td>
  </tr>
</template>
```

Slot

- L'élément HTML `<slot>` représente un emplacement d'un composant web qu'on peut remplir avec son propre balisage.
- Les `<slot>` sont identifiés par leur attribut `name` et permettent de définir des emplacements dans un élément `<template>` qui peut être alimenté avec n'importe quelle structure HTML.
- Cet élément fait partie des outils relatifs aux composants web (Framework JS: React, Angular, Vue)

[En savoir plus](#)

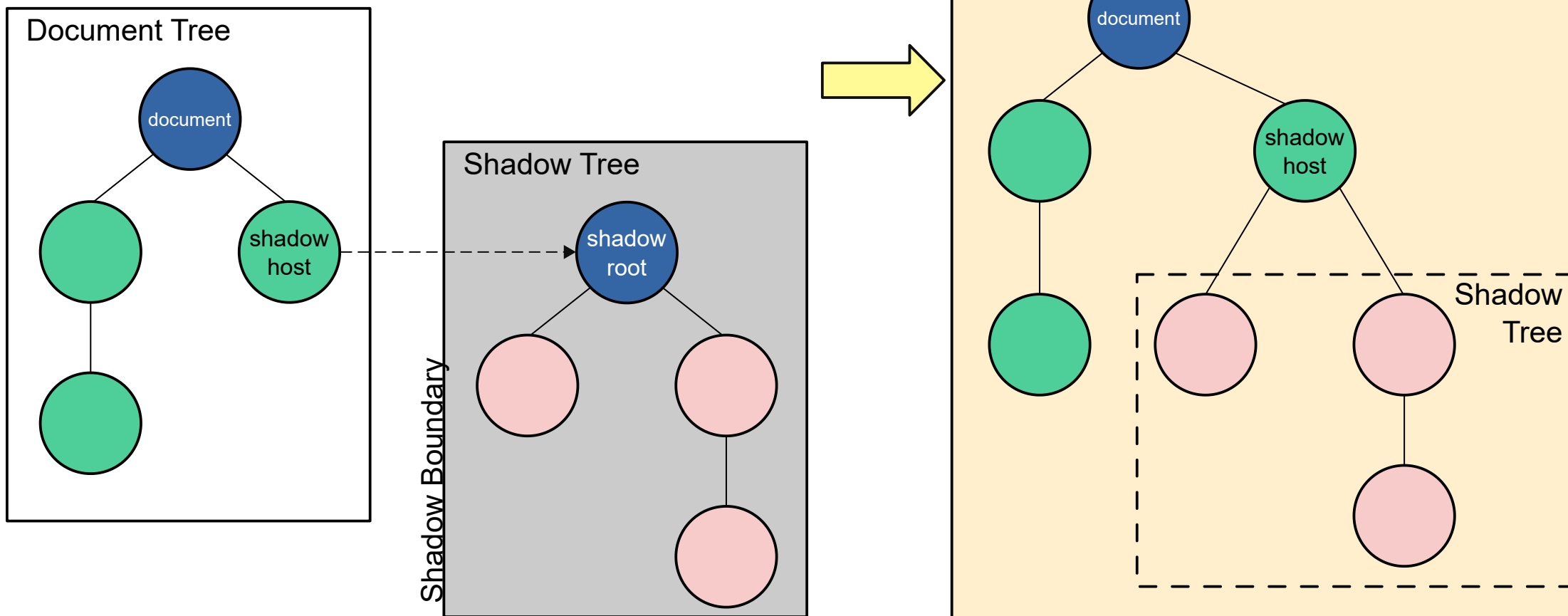
Shadow DOM

Le Shadow DOM est une technique qui permet d'encapsuler la **structure**, le **style** et le **comportement** d'un élément HTML, le séparant du reste de la page.

Terminologie:

- **Shadow Host** : L'élément auquel le Shadow DOM est associé.
- **Shadow Tree** : L'arbre DOM caché à l'intérieur du Shadow DOM.
- **Shadow Boundary** : La limite entre le Shadow DOM et le DOM principal.
- **Shadow Root** : Le nœud racine de l'arbre fantôme.

Mécanisme du shadow DOM



Exemple de shadow DOM

Avez-vous déjà pensé à la façon dont les contrôles complexes des navigateurs sont créés et stylisés ? (Vidéo, slider ...)

Par exemple, l'élément `<video>`, on l'utilise comme un balise mais mais il contient plusieurs boutons et autres contrôles au sein de son Shadow DOM.



Merci pour votre attention

Des questions ?

