

Technologies Web Avancées : Fiche

Sommaire

- Technologies Web Avancées : Fiche
 - Sommaire
 - 1. Présentation de Javascript
 - 1.1. Front-End et Back-End
 - 1.2. Avantages et inconvénients
 - 1.3. Variables
 - 1.4. Tableaux et listes
 - 1.5. Conditions booléennes
 - 1.6. Boucle **for** dans un tableau
 - 2. Présentation de HTML
 - 2.1. Présentation
 - 2.2. Listes
 - 2.3. Tableaux
 - 2.4. Inputs
 - 2.5. Images
 - 2.6. Liens
 - 2.7. Barres de progression et jauges
 - 3. Présentation de CSS
 - 3.1. Présentation
 - 3.2. Sélecteurs
 - 3.3. Unités
 - 3.3. Quelques propriétés pour le texte
 - 3.4. Quelques propriétés pour les blocs
 - 4. Le DOM
 - 4.1. Présentation
 - 4.2. Exemple
 - 4.3. Le DOM avec JavaScript
 - 4.3.1. Récupérer et modifier le contenu d'un élément
 - 4.3.2. Créer ou supprimer un élément
 - 5. Les événements avec JavaScript

1. Présentation de Javascript

Type de fichier	Rôle
HTML	La structure de la page
CSS	Sa mise en forme
Javascript	Le comportement de la page et ses interactions

1.1. Front-End et Back-End

Front-End :

Lorsque nous utilisons une **application** (site web, appli web ou appli mobile), nous interagissons avec l'**interface utilisateur**.

Cette interface utilisateur est désignée par le Front-End, c'est tout ce que nous **voyons** (animations, texte, boutons...).

Back-End :

Le Back-End désigne l'**arrière-plan**, la partie que l'utilisateur **ne peut pas voir**.

C'est dans le Back-End qu'est géré le comportement de l'application en réponse à l'utilisateur (connexion à une base de données, redirection...).

Javascript est un langage de programmation qui permet de **dynamiser** les pages web. Il est utilisé pour le **Front-End** et le **Back-End**.

1.2. Avantages et inconvénients

Avantages :

- **Facilité d'apprentissage** : Javascript est un langage très simple à apprendre, il est très proche du langage **anglais**.
- **Populaire** : Javascript est un langage très utilisé, il est donc très facile de trouver de l'aide sur internet et il y a beaucoup de bibliothèques et de frameworks.
- **Polyvalent** : Utilisable côté **Front-End** et **Back-End**.

Inconvénients :

- **Faible sécurité** : Javascript est un langage très simple à apprendre, il est donc très facile de faire des erreurs et de créer des failles de sécurité.
- **Langage de script** : Javascript est un langage de script interprété, exécuté par le navigateur, il est donc plus lent qu'un langage de programmation standard.
- **Dépendance** : Javascript est un langage dépendant du navigateur, il est donc difficile de faire des applications mobiles ou de bureau et le résultat varie en fonction du navigateur.

1.3. Variables

Types de variables	Exemples
Déclaration	Local : <code>let unEntier = 3;</code> Global : <code>var unEntier = 3;</code>
Affectation	<code>somme = 1.2 + 1.3;</code>
Chaînes	<code>let nom = "toto";</code> ou <code>let nom2 = 'tata';</code>
Booléens	<code>true</code> ou <code>false</code>
Opérateurs ou, et, non	<code> </code> <code>&&</code> <code>!</code>
Tableaux	<code>t = [1,2,3,4];</code> et <code>t[1]</code> ou <code>t.length</code>
Dictionnaires	<code>dico = {a:1, b:2, c:3};</code>

1.4. Tableaux et listes

```
// Tableau : on initialise un tableau vide
let t[4]
t = [1,2,3,4]

// Liste : pas de taille fixe
let l = [1,2,3,4]

// Un tableau ou une liste peut contenir différents types de données
let t2 = [1, "toto", true]
```

La différence entre un tableau et une liste est juste dans les règles de l'art, **en pratique c'est la même chose**.

1.5. Conditions booléennes

```
1 == "1"; // vrai
1 === "1"; // faux

var result = 8 % 2 > 0 || !(3 % 2 < 1); // vrai
```

1.6. Boucle **for** dans un tableau

```
// Pour tout élément i du tableau t, on affiche i
for(let var i of t) {
  console.log(i)
}
```

2. Présentation de HTML

2.1. Présentation

HTML (*HyperText Markup Language*) est un langage de balisage utilisé pour représenter les pages web. Il permet de structurer et de mettre en forme le contenu des pages web.

2.2. Listes

```
<!-- Liste ordonnée -->

<ol>
  <lh> Liste ordonnée </lh>
  <li>Item 1</li>
```

```
<li>Item 2</li>
<li>Item 3</li>
</ol>
```

Liste ordonnée

1. Item 1
2. Item 2
3. Item 3

```
<!-- Liste non ordonnée -->

<ul>
  <lh> Liste non ordonnée </lh>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

Liste non ordonnée

- Item 1
- Item 2
- Item 3

2.3. Tableaux

```
<table>
  <caption>
    Tableau simple
  </caption>
  <tr>
    <th>COLONNE 1</th>
    <th>COLONNE 2</th>
    <th>COLONNE 3</th>
  </tr>
  <tr>
    <td>Première ligne col1</td>
    <td>Première ligne col2</td>
    <td>Première ligne col3</td>
  </tr>
  <tr>
    <td>Deuxième ligne col1</td>
    <td>Deuxième ligne col2</td>
    <td>Deuxième ligne col3</td>
  </tr>
  <tr>
    <td>Troisième ligne col1</td>
    <td>Troisième ligne col2</td>
    <td>Troisième ligne col3</td>
  </tr>
```

```

<tr>
  <td>....</td>
  <td>....</td>
  <td>....</td>
</tr>
<tr>
  <td>Dernière ligne col1</td>
  <td>Dernière ligne col2</td>
  <td>Dernière ligne col3</td>
</tr>
</table>

```

Tableau simple

COLONNE 1	COLONNE 2	COLONNE 3
Première ligne col1	Première ligne col2	Première ligne col3
Deuxième ligne col1	Deuxième ligne col2	Deuxième ligne col3
Troisième ligne col1	Troisième ligne col2	Troisième ligne col3
....
Dernière ligne col1	Dernière ligne col2	Dernière ligne col3

2.4. Inputs

type	description
text	champ texte
password	champ mot de passe
email	champ email
number	champ nombre
date	champ date
time	champ heure
checkbox	case à cocher
radio	bouton radio
submit	bouton de validation
file	champ de fichier
range	curseur de valeur
code	champ de code

2.5. Images

```

```



- `src` : chemin de l'image
- `alt` : texte alternatif (affiché si l'image ne s'affiche pas)
- `width` : largeur de l'image

2.6. Liens

```
<a href="https://www.google.com" target="_blank">Google</a>
```

Google

- `href` : chemin du lien
- `target` : ouverture du lien
 - `_blank` : ouverture dans un nouvel onglet
 - `_self` : ouverture dans la même page

2.7. Barres de progression et jauges

```
<progress value="50" max="100"></progress>
```



```
<meter id="espace_dispo" value="0.5"></meter>
```



3. Présentation de CSS

3.1. Présentation

CSS signifie *Cascading Style Sheets*. Il s'agit d'un langage informatique utilisé pour décrire la présentation des documents **HTML** et **XML**. Il permet de contrôler le rendu visuel des pages web.

3.2. Sélecteurs

Sélecteur	Description
p.formatage	"Tous les p qui ont comme classe formatage"
p#formatage	"Tous les p qui ont comme id formatage"
p, .formatage	"Tous les p et toutes les classes formatage"
p .formatage	"Toutes les classes formatage qui sont dans un p"

3.3. Unités

Unité	Description	Exemple
px	Les pixels sont une unité de mesure relative à l'écran de l'utilisateur. Une valeur de 1 pixel correspond à un point sur l'écran.	font-size: 16px;
em	Les ems sont une unité de mesure relative à la taille de la police actuelle. Une valeur de 1em correspond à la taille de la police actuelle.	font-size: 1.2em;
rem	Les rems sont une unité de mesure relative à la taille de la police de la racine (root). Une valeur de 1rem correspond à la taille de la police de la racine.	font-size: 1.5rem;
%	Les pourcentages sont une unité de mesure relative à un parent ou à une taille de police. Une valeur de 100% correspond à la taille du parent ou à la taille de police actuelle.	width: 50%;
vw	Les vw (viewport width) sont une unité de mesure relative à la largeur de la fenêtre d'affichage de l'utilisateur. Une valeur de 1vw correspond à 1/100 de la largeur de la fenêtre d'affichage.	width: 50vw;
vh	Les vh (viewport height) sont une unité de mesure relative à la hauteur de la fenêtre d'affichage de l'utilisateur. Une valeur de 1vh correspond à 1/100 de la hauteur de la fenêtre d'affichage.	height: 50vh;
vmin	Les vmin sont une unité de mesure relative à la plus petite dimension de la fenêtre d'affichage (hauteur ou largeur). Une valeur de 1vmin correspond à 1/100 de la plus petite dimension de la fenêtre d'affichage.	font-size: 2vmin;
vmax	Les vmax sont une unité de mesure relative à la plus grande dimension de la fenêtre d'affichage (hauteur ou largeur). Une valeur de 1vmax correspond à 1/100 de la plus grande dimension de la fenêtre d'affichage.	width: 80vmax;

3.3. Quelques propriétés pour le texte

Propriété	Exemple de valeur	Description
font-family	Arial, Helvetica, sans-serif	Police de caractère
font-size	2em	Taille de la police
font-weight	bold	Gras

Propriété	Exemple de valeur	Description
color	#000	Couleur du texte
text-align	center	Alignement du texte
text-decoration	underline	Soulignement du texte
text-transform	uppercase	Transforme le texte en majuscules
text-shadow	2px 2px 2px #000	Ombre du texte
text-outline	2px 2px 2px #000	Contour du texte
opacity	0.5	Opacité du texte

3.4. Quelques propriétés pour les blocs

Propriété	Exemple de valeur	Description
width	100%	Largeur
height	100%	Hauteur
background	#000 url("image.jpg") no-repeat center	Couleur de fond / Image
background-color	#000	Couleur de fond
background-image	url("image.jpg")	Image de fond
background-repeat	no-repeat	Répétition de l'image de fond
background-position	center	Position de l'image de fond
background-size	cover	Taille de l'image de fond
background-attachment	fixed	Fixe l'image de fond
background-origin	border-box	Origine de l'image de fond
background-clip	padding-box	Zone de l'image de fond
background-blend-mode	multiply	Mode de fusion de l'image de fond
border	1px solid #000	Bordure
border-width	1px	Largeur de la bordure
border-style	solid	Style de la bordure
border-color	#000	Couleur de la bordure
border-radius	10px	Arrondi des coins
box-shadow	2px 2px 2px #000	Ombre du bloc
box-sizing	border-box	Taille du bloc
opacity	0.5	Opacité du bloc

Propriété	Exemple de valeur	Description
overflow	hidden	Dépassement du bloc
position	relative	Position du bloc
top	10px	Position du bloc par rapport au haut
right	10px	Position du bloc par rapport à droite
bottom	10px	Position du bloc par rapport au bas
left	10px	Position du bloc par rapport à gauche
z-index	1	Profondeur du bloc
display	inline	Affichage du bloc
float	left	Flottant du bloc
clear	both	Effacement du flottant
visibility	hidden	Visibilité du bloc

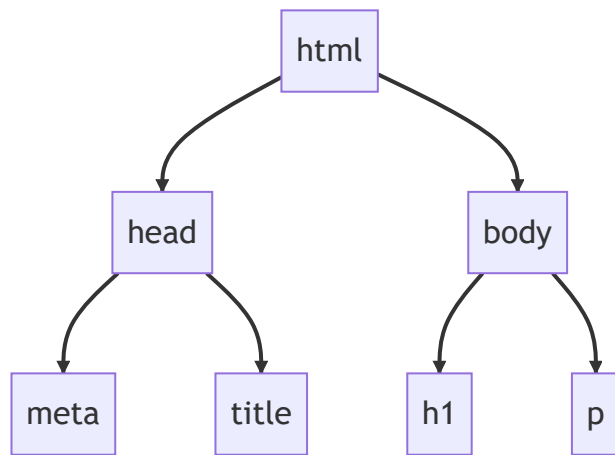
4. Le DOM

4.1. Présentation

Le **DOM** (*Document Object Model*) est une représentation sous forme d'arbre de l'ensemble des éléments d'une page web. Chaque élément est représenté par un objet qui possède des propriétés et des méthodes. Le DOM permet de manipuler les éléments d'une page web.

4.2. Exemple

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8" />
    <title>Exemple DOM</title>
  </head>
  <body>
    <h1>Titre</h1>
    <p>Paragraphe</p>
  </body>
</html>
```



4.3. Le DOM avec JavaScript

Cheatsheet : [codecademy](https://www.codecademy.com/cheatsheet/html-dom)

4.3.1. Récupérer et modifier le contenu d'un élément

- Page HTML de référence

```
<!DOCTYPE html>

<html lang="fr">
  <head>
    <meta charset="utf-8" />
    <title>Exemple DOM</title>
  </head>
  <body>
    <h1 id="titre">Titre</h1>
    <form>
      <input type="radio" name="choix" value="1" id="valeur1" checked="false" />
      <input type="radio" name="choix" value="2" id="valeur2" checked="false" />
    </form>
  </body>
</html>
```

Par identifiant (pareil pour les classes)

```
// Récupération de l'élément par son identifiant
let titre = document.getElementById("titre");
let valeur1 = document.getElementById("valeur1");
let valeur2 = document.getElementById("valeur2");

// Modification de l'élément
titre.innerHTML = "Nouveau titre";
valeur1.checked = true;
valeur2.value = 3;
```

```
// Modification de L'élément (autre méthode, plus stable)
```

```
valeur1.setAttribute("checked", true);  
valeur2.setAttribute("value", true);
```

Par nom

```
// Récupération de L'élément par son nom
```

```
let choix = document.getElementsByName("choix");
```

```
// Modification de L'élément
```

```
choix[0].checked = true;  
choix[1].value = 3;
```

Par balise

```
// Récupération de L'élément par sa balise
```

```
let inputs = document.getElementsByTagName("input");
```

```
// Modification de L'élément
```

```
inputs[0].checked = true;  
inputs[1].value = 3;
```

```
// Récupère tous les éléments
```

```
tout = document.getElementsByTagName("*");
```

Avec le body

```
// Récupération du body
```

```
let body = document.body;
```

```
// Modification de L'élément
```

```
body.firstChild.innerHTML = "Nouveau titre";  
body.lastChild.value = 3;
```

Avec le querySelector

Cheatsheet : [codepen](https://codepen.io)

```
// Récupération de L'élément par son sélecteur
```

```
let titre = document.querySelector("#titre");  
let premierChoix = document.querySelector("input[name='choix']");  
let choix = document.querySelectorAll("input[name='choix']");
```

```
// Modification de L'élément
```

```
titre.innerHTML = "Nouveau titre";
```

```
premierChoix.checked = true;  
choix[1].value = 3;
```

4.3.2. Créer ou supprimer un élément

- Page HTML de référence

```
<!DOCTYPE html>  
  
<html lang="fr">  
  <head>  
    <meta charset="utf-8" />  
    <title>Exemple DOM</title>  
  </head>  
  <body>  
    <div id="ancienDiv">Ancien div</div>  
  </body>  
</html>
```

```
// Création d'un élément  
let nouveauDiv = document.createElement("div");  
let nouveauLien = document.createElement("a");  
  
// Modification de l'élément  
nouveauDiv.innerHTML = "Nouveau div";  
  
nouveauLien.innerHTML = "Google";  
nouveauLien.setAttribute("href", "https://www.google.fr");  
  
// Ajout de l'élément  
document.body.appendChild(nouveauDiv);  
nouveauDiv.appendChild(nouveauLien);  
  
// Suppression de l'élément  
document.body.removeChild(document.getElementById("ancienDiv"));
```

5. Les événements avec JavaScript

Principe : répondre automatiquement à une action de l'utilisateur sur la page web.

Cheatsheet : [codecademy](https://www.codecademy.com/cheatsheet/javascript-events)