Concevoir une page WEB dynamique

Sommaire

1. Objectif de l'activité

- Apprendre à utiliser JavaScript pour agir sur le contenu et sur le format d'une page web

2. JavaScript

JavaScript est un langage interprété. Embarqué dans une page web ou lié à une page web, il permet d'agir sur son contenu, sa structure et sa mise en page. C'est un des trois langages web que doit connaitre tout développeur web :

- HTML pour définir le contenu des pages web
- CSS pour spécifier la mise en page des pages web
- JavaScript pour programmer le comportement des pages web

Le saviez-vous ?

JavaScript et **Java** sont des langages totalement différents.

JavaScript a été inventé par **Brendan Eich** en 1995, et est devenu une norme ECMA en 1997.

ECMA-262 est le nom officiel de la norme. **ECMAScript** est le nom officiel du langage.

2.1. Travail à faire : Premier script

- Tester le code suivant :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>Mon premier script JavaScript</h1>
    <button type="button"
      onclick="document.getElementById('demo').innerHTML = 'Bonjour le
monde !'">
      Clic moi....</button>
    <p id="demo">Le code Javascript, via l'évenement 'onclick' du bouton,
modifie le contenu présent ici</p>
  </body>
</html>
```

- Identifier le code Javascript et modifier le texte à afficher
- Comment le script indentifie-t-il l'élément html à modifier ?
- Modifier le script pour changer le titre.

3. La syntaxe

JavaScript emprunte la plupart des éléments de sa syntaxe à Java, C et C++ mais sa syntaxe est également influencée par Python.

- Les **instructions sont séparées par des points virgules**.
- Un **bloc d'instructions est délimité par une paire d'accolades** :

```
{
   instruction_1;
   instruction_2;
   .
   .
   .
   instruction_n;
}
```

- La déclaration d'une variable est précédée de :
  - `var` : La variable à une portée globale (connue dans tout le script)
  - `let` : La variable à une portée locale (connue seulement dans le bloc courant).
  - `const` : Une constante n'est connue que dans le bloc courant et n'est pas modifiable.
- Comme Python, JavaScript est un langage à typage dynamique. Cela signifie qu'il n'est pas nécessaire de spécifier le type de données d'une variable lors de sa déclaration. C'est lors de l'affectation que JavaScript type la donnée :

```
   var uneVariable = 42; // uneVariable est un entier (Int)
   uneVariable = "Merci pour le dîner..."; // uneVariable est une chaine
de caratères (String)
```

- Comme avec Python, les chiffres sont écrits avec ou sans décimal :

```
   37 // Entier
```

```
37.7 // Flottant
```

- Comme avec Python, les chaînes sont du texte, écrit entre guillemets doubles ou simples :

```
"Voici une chaine de caractère"
'Voici également une chaine de caractères'
```

- Une déclaration de fonction en Javascript est construite avec le mot-clé `function`, suivi par :
  - Le nom de la fonction.
  - Une liste d'arguments à passer à la fonction, entre parenthèses et séparés par des virgules.
  - Les instructions JavaScript définissant la fonction, entre accolades, { }.

```
function carre(nombre) {
  return nombre*nombre;
}
```

3.1. Travail à faire : Console JavaScript

- Dans le navigateur (Mozilla Firefox), ouvrir la **Console web** (F12 → Console)
- Saisir et évaluer les expressions suivantes :

```
var uneVariable = 60;
console.log(uneVariable);
uneVariable = "Maintenant c'est du texte...";
alert("Aïe aïe aïe!")
function carre(nombre) {
   return nombre*nombre;
}
carre(5);
```

3.2. Travail à faire : Agir sur une page

- Saisir le code suivant dans un fichier html et observer le résultat dans un navigateur (Firefox).

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style>
      h1 {
        font-size: 40px;
      }

      h2 {
        font-size: 30px;
      }

      p {
        font-size: 14px;
      }
```

```
    </style>
  </head>
  <body>
    <h1 id="h1">This is heading 1</h1>
    <h2 id="h2">This is heading 2</h2>
    <p id="p1">This is a paragraph.</p>
    <p id="p2">This is another paragraph.</p>
  </body>
</html>
```

- Ouvrir la **Console web** dans cette page
- Evaluer les expressions suivantes :

```
document.getElementById("h1").innerHTML = "Bonjour le monde !"
document.getElementById("h1").style.color="red";
```

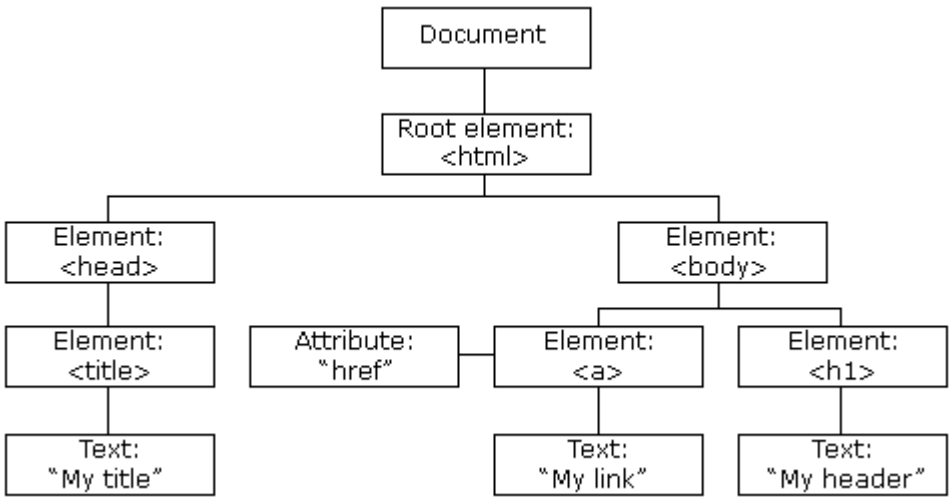
- Modifier le contenu du texte de niveau 2
- Modifier la couleur du texte du second paragraphe
- Modifier la couleur de l'arrière plan du premier paragraphe
- Modifier la couleur de l'arrière-plan de la page

3.3. JavaScript DOM

Lorsqu'une page Web est chargée, le navigateur crée le **Document Objet Modèle** (DOM) de cette page Web.

3.3.1. Qu'est-ce que le DOM ?

Le **DOM** est construit comme un arbre d'objets :



C'est une interface de programmation HTML/CSS. Elle définit :

- Les éléments HTML comme des objets
- Les propriétés de tous les éléments HTML
- Les méthodes pour accéder à tous les éléments HTML

- Les événements pour tous les éléments HTML

En d'autres termes : Le DOM permet **d'obtenir, de modifier, d'ajouter ou de supprimer** des éléments HTML/CSS.

Avec ce système, JavaScript permet de :

- changer tous les éléments HTML dans la page
- changer tous les attributs HTML dans la page
- changer tous les styles CSS dans la page
- supprimer des éléments existants et attributs HTML
- ajouter de nouveaux éléments et attributs HTML
- réagir à tous les événements HTML existants dans la page

3.3.2. L'interface de programmation DOM

Dans le DOM, tous les éléments HTML sont définis comme des objets .

Chaque objet est accesible via des **propriétés** et des **méthodes** :

- **Une propriété** est une valeur que vous pouvez obtenir ou définir (comme changer le contenu d'un élément HTML).
- **Une méthode** est une action que vous pouvez faire (comme ajout ou la suppression d'un élément HTML).

L'exemple suivant modifie le contenu (innerHTML) de l'élément <p> avec id="demo":

```
<html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <p id="demo">Le script modifie le contenu présent ici</p>
    <script>
      document.getElementById("demo").innerHTML = "Hello World!";
    </script>
  </body>
</html>
```

Dans cet exemple :

- `document` : est un objet qui contient tous les éléments de la page web
- `getElementById()` : est une méthode qui permet d'accéder à l'élément indiqué en paramètre (`"demo"` dans l'exemple)
- `innerHTML` : est une propriété qui contient le code HTML de l'élément accédé.

Trouver des éléments HTML

Voici quelques exemples d'utilisation de l'objet `document` pour accéder et manipuler le code HTML :

Methode	Description
`document.getElementById(id)`	Trouver un élément par son identifiant `id`
`document.getElementsByTagName(name)`	Trouver un élément par son type de balise (tag name)
`document.getElementsByClassName(name)`	Trouver un élément par le nom de la classe (`class`) qui lui est affecté

- **Trouver des éléments HTML par le nom de balise HTML (Tag)**

Cet exemple trouve tous les éléments <p>:

```
var x = document.getElementsByTagName("p");
```

Essayer l'exemple sur **w3schools**

Cet exemple trouve l'élément avec id="main" et trouve alors tous les éléments <p> qu'il contient :

```
var x = document.getElementById("main");
var y = x.getElementsByTagName("p");
```

Essayer l'exemple sur **w3schools**

- **Trouver des éléments HTML par le nom de la classe qui leur sont affectés**

Cet exemple retourne une liste de tous les éléments affectés par class="intro".

```
var x = document.getElementsByClassName("intro");
```

Essayer l'exemple sur **w3schools**

3.3.3. Travail à faire : Manipuler l'arbre DOM HTML

- Faire les 5 premiers exercices de w3school JS HTML DOM
- Ajouter dans le code HTML suivant un script qui met en gras et ajoute la balise <blockquote> à la citation d'Albert Einstein :
  - créer une variable `laCitation` qui contiendra le contenu de l'élément HTML d'id `citation`
  - Ajouter lui le code HTML `<blockquote><strong>` avant et `</strong></blockquote>` après
  - affecter le contenu de la variable `laCitation` à l'élément HTML d'id `citation`.
- Ajouter dans l'entête du fichier HTML une déclaration interne de style qui met en bleu le texte contenu dans les balises `<blockquote>`

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <h2>Une citation</h2>
    <div id="citation">
      <p>Nous aurons le destin que nous aurons mérité.<p>
```

```
      <cite>Albert Einstein</cite>
    </div>
      <script>
        // ici le script qui met en gras <strong> et ajoute la balise
<blockquote> à la citation
      </script>
  </body>
</html>
```

3.4. JavaScript DOM CSS

Pour changer le style d'un élément HTML, utilisez la syntaxe suivante :

```
document.getElementById(id).style.property = new style
```

L'exemple suivant modifie le style d'un élément <p> :

```
<html>
 <html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <h1 id="h1">Hello World!</p>
    <script>
        document.getElementById("h1").style.color = "blue";
    </script>
    <p>La couleur du titre a été changée par un script.</p>
  </body>
</html>
```

3.4.1. Liste des propriétés de style CSS

Toutes les propriétés CSS peuvent être manipulées par JavaScript. Leur nom n'est pas toujours identique à celui utilisé par CSS. Ci-dessous, une liste simplifiée des principales propriétés CSS et leur équivalent en JavaScript :

CSS	JavaScript
background	background
background-color	backgroundColor
background-image	backgroundImage
background-position	backgroundPosition
background-repeat	backgroundRepeat
border	border
border-color	borderColor
border-width	borderWidth
color	color
display	display
float	cssFloat
font	font
font-family	fontFamily
font-size	fontSize
font-weight	fontWeight

CSS	JavaScript
height	height
left	left
margin	margin
overflow	overflow
padding	padding
position	position
text-align	textAlign
text-decoration	textDecoration
text-indent	textIndent
text-transform	textTransform
top	top
vertical-align	verticalAlign
visibility	visibility
width	width

liste complète sur w3school

3.4.2. Travail à faire : Manipuler l'arbre DOM CSS

- Faire les 4 derniers exercices de w3school JS CSS DOM
- Compléter le code suivant pour faire afficher le texte ou le faire dispaitre :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <p id="p1">Coucou me voilà ... !</p>
    <input type="button" value="Hide text"
      onclick="document.getElementById(____).____.____='hidden'">
    <input type="button" value="Show text"
      onclick="document.getElementById(____).____.____='visible'">
  </body>
</html>
```

- Compléter le code suivant pour changer la couleur de fond de la page lorsque l'une des cellules du tableau est survolée :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>

    <h2>Change la couleur de fond !</h2>
    <p>Passe la souris sur les les carrés colorés</p>

        <table style="width:300px;height:100px">
```

```
          <tr>
            <td style="background-color:Khaki"
                onmouseover="document.body.____.____ =
this.style.backgroundColor"
                onmouseout="document.body.____.____ = 'transparent'"
            >
            </td>
            <td style="background-color:PaleGreen"
                onmouseover="document.body.____.____ =
this.style.backgroundColor"
                onmouseout="document.body.____.____ = 'transparent'"
            >
            </td>
            <td style="background-color:Silver"
                onmouseover="document.body.____.____ =
this.style.backgroundColor"
                onmouseout="document.body.____.____ = 'transparent'"
            >
            </td>
          </tr>
        </table>
  </body>
</html>
```

Dans cet exercice, on a pas utilisé une méthode comme `getElementById()` pour agir sur la page web (objet `document`) mais on a agit directement sur l'élément `body` de l'objet `document`.

- document.anchors
- document.body
- document.documentElement
- document.embeds
- document.forms
- document.head
- document.images
- document.links
- document.scripts
- titre du document

- Compléter le code suivant pour ouvrir une fenêtre lorsqu'on clic sur le bouton :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
        <p>Cliquez sur le bouton pour ouvrir une nouvelle fenêtre</p>
        <button onclick="myFunction()">Clic ici !</button>
        <script>
         function myFunction() {
           var w = window.open();
           w.document.open();
           w.document.write("<h2>Coucou !</h2>");
           w.document.close();
         }
        </script>
```

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

3.5. Les événements du DOM HTML

Comme vous avez pu l'expérimenter aux travers des derniers exercices, JavaScript permet de réagir aux événements HTML.

Un JavaScript peut être exécuté lorsqu'un événement se produit, comme lorsqu'un utilisateur clique sur un élément HTML.

Pour exécuter du code lorsqu'un utilisateur clique sur un élément, ajoutez le code JavaScript à un attribut d'événement HTML :

```
onclick=JavaScript
```

Exemples d'événements HTML:

- Lorsqu'un utilisateur clique sur la souris
- Lorsqu'une page Web a chargé
- Lorsqu'une image a été chargée
- Lorsque la souris survol sur un élément
- Lorsqu'un champ d'entrée de formulaire est modifié
- Lorsqu'un formulaire HTML est soumis
- Lorsqu'un utilisateur appuie sur une touche

Dans cet exemple, le contenu de l'élément <h1> est modifié lorsqu'un utilisateur clique dessus:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <h1 onclick="this.innerHTML = 'Ooops!'">Click on this text!</h1>
  </body>
</html>
```

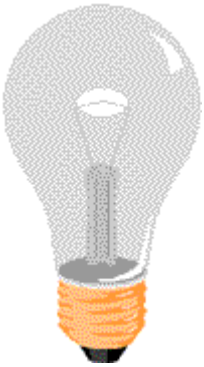
Essayez sur w3schools

Sachant que l'élément à mofifier est le même que celui sur lequel à lieu l'événement, on n'a pas besoin de le rechercher avec `document.getElementById()`, on utilise `this` qui représente l'objet courant.

Vous trouverez ici la liste complète des événements HTML

3.5.1. Travail à faire : Les événements

- Téléchargez les images suivantes (clic droit sur l'image → Enregistrer l'image sous…) :

nom du fichier	image
bulboff.gif	
bulbon.gif	

- Complétez le code suivant pour faire afficher une ampoule alumée lorsque la souris est dessus et éteinte lorsque la souris n'est pas dessus :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script>
        function lighton() {
          ...
        }
        function lightoff() {
          ...
        }
    </script>
  </head>
  <body>
    <img id="myimage" ______="lighton()" ______="lightoff()"
src="bulboff.gif" width="100" height="180" />
    <p>Survol l'ampoule...</p>
  </body>
</html>
```

Référence : www.silanus.fr