

TP JavaScript

Exemple du code HTML avec une balise script à la dernière ligne du body :

```
<html>
  <head>
    <title>Exemple JS</title>
  </head>
  <body>
    <h1 id="titre">Titre</h1>
    <p id="para">Paragraphe</p>
    <script>
      document.getElementById("titre").innerHTML = "Nouveau titre";
    </script>
  </body>
</html>
```

Que fait-elle ?

- document désigne l'ensemble du document, de la page web.
- getElementById("titre") est un *sélecteur* qui va récupérer la première balise dont l'id est "titre".
- .innerHTML = "Nouveau titre" va remplacer le contenu html de la balise sélectionnée par "Nouveau titre".

Ainsi JS permet de changer le contenu d'une balise.

Il existe de nombreuses méthodes permettant de faire ce genre de choses.

Modifier le style

Une fois qu'on a récupéré un *élément html*, on peut lui attribuer du style, par exemple :

```
var element = document.getElementById("titre");
element.style.color = "red";
```

Exercice : lire attentivement et écrire le CSS équivalent à ce script.

id

L'attribut universel **id** définit un identifiant qui doit être unique pour l'ensemble du document. Le but de cet attribut est de pouvoir identifier un élément lorsqu'on crée un lien, avec un fragment et qu'on souhaite le manipuler avec un script ou qu'on le met en forme avec CSS.

Éléments de syntaxe

- Commentaire

- `// ceci es un commentaire`

- Variable

- `a = 1;` // une variable globale

- `var b = 2;` // une variable locale

- `let c = 3;` // une variable dont la portée est limitée au bloc courant

- conditions

- `if (condition) {`

- `inscrution;`

- `}`

- boucles

- `for (let i = 0; i < 10; i++) {`

- `// corps de la boucle`

- `}`

Pour `i` entre 0 et 9 et progressant de 1.

Équivalent à :

```
for i in range(10):  
  message
```

- Afficher un message dans la console de développeur

- `var prenom = "Personne";`

- `console.log("bonjour", prenom);`

Va afficher “bonjour personne” dans la console.

TP 1 : Modifier les attributs html : `document.getElementById`

Considérons ce contenu html

```
<html>  
  <body>  
    <p class="monPara">bonjour</p>  
  </body>  
</html>
```

Insérons maintenant un script JS à la fin du body et le fichier css suivant :

```
.rouge {  
  color: red;  
}  
.vert {  
  color: green;  
}  
<html>  
  <head>  
    <link rel="stylesheet" href="style.css" />
```

```

</head>
<body>
  <p id="monPara">bonjour</p>
  <p class="rouge">rouge</p>
  <p class="vert">vert</p>
  <script>
    var monPara = document.getElementById("monPara");
  </script>
</body>
</html>

```

Exercice : écrire le rendu de cette page.

Maintenant nous allons ajouter à l'élément sélectionné dans le script :

```

<html>
  <head>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <p id="monPara">bonjour</p>
    <p class="rouge">rouge</p>
    <p class="vert">vert</p>
    <script>
      var monPara = document.getElementById("monPara");
      monPara.classList.add("rouge");
    </script>
  </body>
</html>

```

Et javascript va ajouter la classe "rouge" et son style à notre paragraphe.

Interactivité

Ajoutons maintenant des boutons...

```

<html>
  <head>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <p id="monPara">bonjour</p>
    <p class="rouge">rouge</p>
    <p class="vert">vert</p>
    <button onclick="foncRouge()">Rouge</button>
    <button onclick="foncVert()">Vert</button>
    <script>
      var monPara = document.getElementById("monPara");
      monPara.classList.add("rouge");
    </script>
  </body>
</html>

```

Que font-ils ? Rien.

Ou plutôt, lorsqu'on clique dessus, JS déclenche un événement "onclick" qui à son tour, *tente* d'exécuter une fonction `foncRouge` ou `foncVert` mais celles-ci n'étant pas définies... il ne se passe rien.

Définissons les !

```
<html>
  <head>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <p id="monPara">bonjour</p>
    <p class="rouge">rouge</p>
    <p class="vert">vert</p>
    <button onclick="foncRouge()">Rouge</button>
    <button onclick="foncVert()">Vert</button>
    <script>
      var monPara = document.getElementById("monPara");
      monPara.classList.add("rouge");

      function foncRouge() {
        monPara.classList.remove("vert");
        monPara.classList.add("rouge");
      }
      function foncVert() {
        monPara.classList.remove("rouge");
        monPara.classList.add("vert");
      }
    </script>
  </body>
</html>
```

Remarquons d'abord la syntaxe des fonctions (il en existe d'autres) :

```
function maFonction() {
  inscriptions;
}
```

Que font ces fonctions :

- la première retire la classe "vert" et ajoute la classe "rouge"
- la seconde fait le contraire.

Résumons, lorsqu'on appuie sur le bouton "rouge", on colorie le paragraphe "monPara" en rouge, et lorsqu'on appuie sur le bouton "vert", on colorie le paragraphe "monPara" en vert.

Exercice : Modifier les fonctions afin de changer aussi le contenu du paragraphe lorsqu'on clique sur les boutons.

TP2 : les alternatives et les boucles en js : `document.write`

```
<!DOCTYPE html>
<html lang="fr">
```

```

<head>
<title>Premiers algorithmes en JavaScript</title>
<meta charset="utf-8" />
<link rel="stylesheet" href="idemm.css" />
</head>
<body>

<h1>Premiers algorithmes en JavaScript</h1>
<script>
document.write('bonjour !');

    var temperature = 12;
    if (temperature>20) {
        document.write('il fait chaud !');
    } else {
        document.write('il fait froid !');
        document.write('<br>');
    }

    for (var i=1 ; i<=100 ; i=i+1) {
        document.write(i);
        document.write('<br>');
    }

    var i=1;
    while (i<=100) {
        document.write(i);
        document.write('<br>');
        i = i+1;
    }
</script>
ça marche ?

</body>
</html>

```

TP 3 : les fonctions en js : document.write

```

<!DOCTYPE html>
<html lang="fr">
<head>
<title>Figures avec procédures JavaScript</title>
<meta charset="utf-8" />
</head>
<body>

<h1>Figures avec procédures JavaScript</h1>
<script>

```

```

function unecase (c) {
    document.write('<td>');
    document.write(c);
    document.write('</td>');
}

function repeter (c,n) {
    for (var i=1 ; i<=n ; i=i+1) {
        unecase(c);
    }
}

function lignepleine (c,n) {
    document.write('<tr>');
    repeter(c,n);
    document.write('</tr>');
}

function lignecreuse (c1,c2,n) {
    document.write('<tr>');
    unecase(c1);
    repeter(c2,n-2);
    unecase(c1);
    document.write('</tr>');
}

function carrecreux (c1,c2,n) {
    document.write('<table border="1">');
    lignepleine (c1,n);
    for (var i=1 ; i<=n-2 ; i=i+1) {
        lignecreuse (c1,c2,n);
    }
    lignepleine (c1,n);
    document.write('</table>');
}

function carreplein (c,n) {
    carrecreux(c,c,n);
}

carrecreux('X','o',12);
carreplein('@',5);
carrecreux('', '',8);

</script>
</body>
</html>

```

Exercice : proposer des script js pour activer les boutons des différentes cases des carrés.

Références Bibliographiques

https://qkzk.xyz/docs/nsi/cours_premiere/ihm_web/js/