JavaScript : Découverte

1. Modeler une page web avec JavaScript

Vous allez créer une page html, un fichier css et un fichier js.

Ce dernier va permettre d'ajouter de l'interactivité à votre page html.

1. Vous allez créer une page html `test1.html` et le mettre dans le dossier test1 crée dans ThemeD/JavaScript.
   Voici le contenu de la page :

```
<!DOCTYPE html>
<html lang="fr">
<head>
   <meta charset="UTF-8">
   <title>Test 1</title>
</head>
<body>
   <h1>Voici un premier exemple de JS.</h1>
</body>
<script>
   var prenom = prompt('Quel est votre prénom ?');
   alert('Bonjour ' + prenom);
</script>
</html>
```

2. Ouvrez la page dans le navigateur **Firefox**.
3. Maintenant, modifiez `alert` en `console.log` .
   Vous n'avez plus de fenêtre vous saluant, mais vous allez découvrir la console cachée dans votre navigateur.
4. Ouvrez les outils de développement (menu en haut à droite Développement Web/Outils de développement) ou avec le raccourci Ctrl + SHIFT + i
   Ensuite, sélectionnez la console et observez la salutation.

   Cet fonctionnalité est bien pratique pour les développeurs quand il faut débogger un code.

2. Deuxième exemple :

1. Dans le dossier ThemeD/JavaScript créer le répertoire test2. Ensuite créer dans ce répertoire le fichier test2.html dont le contenu est le suivant :

```
<!DOCTYPE html>
<html lang="fr">
<head>
   <meta charset="UTF-8">
   <title>Test 2</title>
</head>
<body>
   <h1>Voici un second exemple de JS.</h1>
```

```
</body>
<script src="second.js"></script>
</html>
```

Celui-ci fera appel à un fichier JavaScript (nommé second.js) placé dans le même repertoire ThemeD/JavaScript/test2.

2. Créez le fichier second.js dans le dossier ThemeD/JavaScript/test2
   En voici le contenu :

```
var prenom = prompt('Quel est votre prénom ?');
var nom = prompt('Quel est votre nom ?');
var age = prompt('Quel est votre age');
var age_futur = age + 1 ;
alert('Bonjour ' + prenom + nom);
alert('Dans un an vous aurez ' + age_futur + ' ans');
```

3. A votre avis, que fais ce script ?
   Vérifiez votre conjecture ! Ouvrez le fichier test2.html et regardez le résultat. Est-ce que vous vous attendiez à voir ?
4. Corrigeons notre fichier. En fait, la variable age du type "chaîne de caractère" au lien d'être un entier. D'où le résultat précédent.
   Il faut convertir cette variable. Ajoutez la ligne suivante après la ligne var age = prompt('Quel est votre age');
   ```
   age = parseInt(age);
   ```
5. Testez le résultat, c'est mieux non ?

3. Exercice d'application :

En vous inspirant du dernier exemple, créez un fichier html et un fichier JavaScript dans le répertoire ThemeD/JavaScript/Exercice1

A l'ouverture du fichier html, une fenêtre demandera la largeur d'un rectangle, puis une autre demandera la longueur d'un rectangle.

Puis la page affichera dans une fenêtre l'aire du rectangle.

Vous avez réussi ? Passez à la suite !

Nous allons nous intéresser à la manipulation des codes html et css via le JavaScript.

4. Modeler une page web avec JavaScript :

Le Javascript va modifier les pages html et css en accédant aux éléments cible du DOM.
(https://fr.wikipedia.org/wiki/Document_Object_Model)

L'accès aux éléments html peut se faire via 4 méthodes :

- **getElementById()**
- **getElementByName**()
- **getElementByClassName()**
- **getElementByTagName()**

Mise en pratique :

1. Créez un fichier `interraction.html` et saisissez le code ci-dessous :
2. ```
   <!DOCTYPE html>
   <html lang="fr">
   <head>
      <meta charset="UTF-8">
      <title>Interraction html/css et JS</title>
   </head>
   <body>
      <h1>Voici un joli titre.</h1>
      Ceci est un texte qu'il faut mettre en valeur !

      <button onclick="change_couleur()">Cliquez ici !</button>
   </body>
   <script src="interraction.js"></script>
   </html>
   ```
3. Dans le code précédent, `<button onclick="change_couleur()">Cliquez ici !</button>` est une balise qui va créer un bouton. Un clic dessus va déclencher la fonction `change_couleur()` Cette fonction va être définie dans un fichier javascript (`interraction.js`) qui va être créé par ce qui suit ...
4. Ensuite créez un fichier `interraction.js` dans le dossier qui contient `interraction.html` Saisissez le code suivant :
5. ```
   function change_couleur() {
      var paragraphe = document.getElementById("important");
      console.log(paragraphe);
   }
   ```
6. Ouvrez la console des outils de développement web.
   Vous devez constater que la variable `paragraphe` contient maintenant la balise paragraphe et son contenu.

   Dans la variable paragraphe, l'élément d'ID (identifiant) "important" est stocké.

7. Ajoutez le code ci-dessous au fichier Js.

   ```
   paragraphe.style.color="red"
   ```

8. Appuyez sur le bouton et constatez le changement.
   Essayons de faire le changement plus proprement avec le css

9. Ajoutez le fichier interraction.css dans le même dossier .

   A ce stade , l'arborescence de votre dossier doit être :

```
├─Interraction.html
├─  interraction.css
└─  interraction.js
```

   Insérez le contenu ci-dessous dans le fichier :

```
h1{
   text-align: center;
   font-size: 40px;
}

.rouge {
   color:red;
   font-size:30px;
}
```

10. Puis ajoutez la ligne suivante dans le html pour lier le css avec votre fichier html.
```
<link rel="stylesheet" href="interraction.css">
```
11. Et enfin modifiez le js comme ce qui suit :

```
function change_couleur() {
    var paragraphe = document.getElementById("important");
    console.log(paragraphe);
    paragraphe.classList.add("rouge");
}
```

    Ce code aura pour effet d'ajouter la classe "rouge" à notre élément "paragraphe" sélectionné.
12. Ajoutez le bouton suivant dans le html :
```
    <button onclick="reset_couleur()">Reset</button>
```

    Puis ajoutez la fonction suivante dans le Js :

```
function reset_couleur() {
    var paragraphe = document.getElementById("important");
    console.log(paragraphe);
    paragraphe.classList.remove("rouge");
}
```

    Testez le résultat !

5. Exercice 2 :

Ajoutez un bouton supplémentaire qui changera paragraphe de la façon suivante :

- couleur bleue
- taille : 70 px

- centré à droite

Pensez à modifier le code du bouton reset pour qu'il fonctionne encore !

6. Une autre façon de sélectionner des éléments html : les querySelector

Le principe de sélection d'un querySelector est le suivant, on donne en paramètre un seul argument : une chaîne de caractère qui doit être un sélecteur CSS.

Par exemple si dans votre fichier css vous avez :

```
#menu .item p {
color : red;
}
```

alors vos paragraphes d'identifiant menu et de classe item seront en rouge.

Il y a deux méthodes pour les querySelector .

1. querySelector() : renvoie le premier élément trouvé correspondant au sélecteur
2. querySelectorAll() : renvoie tous les éléments correspondant au sélecteur

Nous allons tester ces deux méthodes avec le fichier suivant :

Enregistrez le fichier query.html qui se trouve dans classroom et le mettre dans le dossier themeD/JavaScript/query

Dans le même dossier créez un fichier `query.js` que vous allez compléter comme qui suit :

```
var query = document.querySelector("nav ul li");

console.log("Affichage de query");

console.log(query);
```

Dans ce code :

1. Une variable nommée `query` contient le premier élément de la page html dont le sélecteur css est `nav ul li`
2. Ensuite un affichage est effectué dans la console du développeur

Ouvrez la page HTML dans Firefox, puis ouvrez la console du développeur (CTRL + SHIFT+ i)

Observez l'élément obtenu, vous pouvez dérouler le contenu de cet élément en appuyant sur le petit triangle à gauche de son nom.

Passons à querySelectorAll(), ajoutez les lignes suivantes dans le fichier `query.js`

```
var queryAll_1 = document.querySelectorAll(".menu li");

console.log("Affichage de queryAll_1");

console.log(queryAll_1);
```

Cette fois dans la variable queryAll_1 , il y aura tous les éléments dont le sélecteur CSS est ".menu li", c'est à dire les items de listes dont la classe est "menu".
Ouvrez le fichier HTML dans un navigateur et ouvrez la console du développeur (CTRL + SHIFT + i).
Vous constatez qu'il y a TROIS éléments, ils sont stockés dans un tableau.
Pour afficher les éléments de façon individuelle dans un tableau, cela se passe comme en python.
queryAll_1[0] affichera le 1er élément du tableau.

Ajoutez les lignes suivantes dans votre fichier js.

```
console.log("Affichage de chaque éléments de queryAll_1 par appels successifs")

console.log(queryAll_1[0]);

console.log(queryAll_1[1]);

console.log(queryAll_1[2]);
```

Les trois éléments sélectionnés seront affichés. Mais s'il y en a plus, cette méthode sera fastidieuse.

Utilisons une boucle pour afficher ces éléments un à un.
Pour ce faire, nous allons sélectionner tous les éléments "li" enfants de balise <nav>.
Un sélecteur valide est : nav li. Ajoutez les lignes suivantes dans le fichier js.

```
var queryAll_2 = document.querySelectorAll("nav ul li");

console.log("Affichage de queryAll_2");

console.log(queryAll_2);
```

Vous allez trouver 6 éléments. Affichons-les dans la console de façon individuelle avec une boucle (voir cours js écrit avec le diaporama)

```
console.log('Affichage de chaque éléments de queryAll_1 avec une boucle pour : ');
```

```
for (let index = 0; index < queryAll_2.length; index++) {

    const element = queryAll_2[index];

    console.log(element);

}
```

Ce type de boucle sera pratique pour modifier chaque élément sélectionner avec un querySelectorAll.

---

7. Modification du contenu des éléments HTML avec JS.

Nous sommes maintenant capables de sélectionner divers éléments HTML, mais il serait pratique de modifier leur contenu. Par exemple, modifier ou ajouter du texte dans un paragraphe.

Pour cela la méthode `innerHTML` va nous servir. Ajoutez les lignes suivantes à votre fichier JS.

```
var important = document.querySelector("#important");

/* Affichage de l'élément sélectionné ayant l'ID important. */

alert(important);

/* Affichage du contenu du paragraphe ayant l'ID important. */


alert(important.innerHTML);
```

la variable important contient l'élément HTML dont l'ID est "important". Pour accéder à son contenu textuel, il faut utiliser la méthode `innerHTML` .
La syntaxe est la suivante : objet.méthode donc ici important.innerHTML est le contenu textuel de l' "objet " important.

Il est possible de changer ce texte

```


/* Modification du paragraphe */

important.innerHTML = " Voici une citation d'Alan Turing : <q> Les tentatives de création de machines pensantes

nous seront d'une grande aide pour découvrir comment nous pensons nous-mêmes. </q>";
```

```
alert('Observez le 2eme paragraphe du deuxième article. Il va encore se modifier quand vous c
liquerez sur OK. ');

important.innerHTML += "<br> Cette citation date de 1951";
```

Avec la première ligne de code, la variable important.innerHTML reçoit un nouveau texte entre guillemets. Il est possible d'y placer des balises html.
Dans la dernière ligne de code, comme en python, += permet d'ajouter du contenu à la variable. (cela reviendrais à écrire : important.innerHTML = important.innerHTML + "
 Cette citation date de 1951";)

8. Interactions avec l'utilisateur :

Les événements permettent de déclencher une fonction selon qu'une action s'est produite ou non. Par exemple, faire apparaître une fenêtre alert() lorsque l'utilisateur survole une zone d'une page web, ou ajouter un élément lors d'un clic de bouton de la souris (ou du clavier).

Liste des évènements :

Nom de l'événement	Action pour le déclencher
`click`	Cliquer (appuyer puis relâcher) sur l'élément
`dblclick`	Double-cliquer sur l'élément
`mouseover`	Faire entrer le curseur sur l'élément
`mouseout`	Faire sortir le curseur de l'élément
`mousedown`	Appuyer (sans relâcher) sur le bouton gauche de la souris sur l'élément
`mouseup`	Relâcher le bouton gauche de la souris sur l'élément
`mousemove`	Faire déplacer le curseur sur l'élément
`keydown`	Appuyer (sans relâcher) sur une touche de clavier sur l'élément
`keyup`	Relâcher une touche de clavier sur l'élément
`keypress`	Frapper (appuyer puis relâcher) une touche de clavier sur l'élément
`focus`	« Cibler » l'élément
`blur`	Annuler le « ciblage » de l'élément
`change`	Changer la valeur d'un élément spécifique aux formulaires (`input`,`checkbox`, etc.)
`input`	Taper un caractère dans un champ de texte (son support n'est pas complet sur tous les navigateurs)
`select`	Sélectionner le contenu d'un champ de texte (`input`,`textarea`, etc.)

La pratique :

Commençons par un évènement déjà rencontré, le clic !

Toujours dans le même dossier, créez un fichier HTML nommé : `evenements.html`

Écrivez à l'intérieur :

```
```

```
<span onclick="alert('Hello')" > Cliquez ici !</span>
```

Puis visualisez ce fichier dans Firefox.
Un attribut de la balise est onclic, ce qui signifie que la fonction alert(..) ne sera déclanchée que lors d'un clic sur cet élément span (une sorte de boîte inline)

Voici un autre exemple avec le focus.
Ajoutez les lignes suivantes dans le fichier HTML :

```
<br>

<input type="text" id="input" size="50" value="Cliquez ici"

    onfocus="this.value = 'Appuyez sur la tabulation pour perdre le focus'" onblur="this.valu
e= 'Cliquez ici !'">
```

Le mot clé this permet de pointer sur l'objet qui a déclenché l'évènement.
onfocus : se déclenchera si le "focus" est pris par la balise input.
onblur : se déclenchera si le "focus" est perdu par la balise input.

Testez dans le navigateur !

Il est recommandé de ne pas mettre les fonctions javascript dans les balises html. Nous allons les séparer dans la suite de ce TP.
Fabriquons un bouton interactif :

Dans le HTML, ajoutez les lignes suivantes :

```
<br>

    <button id="clickme"> Cliquez sur moi !!! </button>
```

et juste avant la balise fermante HTML, ajoutez la ligne suivante :

```
<script src="js/evenements.js"></script>
```

Puis créez un fichier evenements.js. Dans celui-ci ajoutez :

```
var element = document.querySelector('#clickme');



element.onclick = function() {

    alert('Je sais que vous avez cliqué !');

}
```

et testez dans votre navigateur.

Cette dernière méthode n'est pas la plus récente, il est recommandé maintenant d'utiliser des eventListener() , c'est-à-dire des méthodes qui vont surveiller l'apparition d'évènements.

À la place des dernières lignes de codes du fichier JS, utilisez les lignes suivantes :

```
var element = document.querySelector('#clickme');



element.addEventListener('click', function() {

    alert("J'ai surveillé le click !");

})
```

9. Exercice 4 :

Difficile : Faites un bouton qui change de couleur lors du passage de la souris.
Indice : setAttribute ...