TP3 – Le Query Selector

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 ayant comme identifiant menu et de classe item seront en rouge.

Il y a deux méthodes pour les **querySelector** :

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

Nous allons tester ces deux méthodes.

Créez un fichier index.html dans le répertoire /JS/S1/TP3. Ajoutez le fichier CSS query.js avec la balise « <SCRIPT> ».

Dans le sous-dossier js 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 :

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

Ouvrez la page HTML dans Firefox, puis ouvrez la console du développeur ( CTRL + MAJ + 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 + MAJ + 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, 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.

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.

# 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);

Quelques constats :

* 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 cliquerez 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, += permet d'ajouter du contenu à la variable. (Cela reviendrais à écrire : important.innerHTML = important.innerHTML + "<br> Cette citation date de 1951"; )

# Un exercice pour s’exercer ? (Et une lapalissade ?)

Créez une page HTML contenant des paragraphes avec des titres (H1, H2…)

Ajouter du code dans le fichier js pour que :

🡺 L'utilisateur clique sur un bouton et une fenêtre de type alert s'affiche.

Elle contient le texte : " Attention les titres vont changer"

Ensuite, après un clic sur ok,

🡺 Il faudra ajouter " super intéressant !" à la fin de chaque titre.

Par exemple : le premier titre deviendra : Un premier article super intéressant !