|  |  |
| --- | --- |
| SLAM5 | TP DOM |

**Pré-requis**

* Visual Studio 2019, installé
* Connaissance de base des concepts et des commandes de Git, comme les dépôts, les duplications (fork), les clones, les branches, l’indexation et la désindexation des changements, et l’envoi (push) de commits
* Un compte GitHub

**Travail à réaliser**

Le travail à réaliser ci-dessous doit être référencé dans git et synchronisé sur github.

Chaque nouvel exercice doit faire l’objet de la création d’une branche et doit être rattaché au master lorsque l’exercice fonctionne et avant de passer au suivant.

**DOM**

Le navigateur permet d'accéder « programmativement » à la structure de la page qu'il affiche. En écrivant du code, on peut ainsi modifier dynamiquement la page web : ajouter ou retirer des éléments, changer leur style, etc.

La représentation de la structure d'une page web offerte par un navigateur et exploitable via JavaScript est appelée DOM, pour **Document Object Model**.

Une page web n'est rien d'autre qu'un ensemble de balises imbriquées les unes dans les autres. On peut la représenter sous une forme hiérarchisée appelée une **arborescence**. L'élément <html> en constitue la racine et contient deux éléments : <head> et <body>, qui contiennent eux-mêmes plusieurs sous-éléments.

Prenons la page web suivante :

<!doctype html>

<html>

<head>

<meta charset="utf-8">

<title>Ma page web</title>

</head>

<body>

<h1>Titre de ma page</h1>

<p>Je suis une page Web.</p>

<p>Voici un moteur de recherche de <a href="http://www.google.fr">Google</a>.</p>

<div id="contenu">

<p>Voici une liste de points :</p>

<ul class="pointsClass" id="pointsId">

<li class="point" id="point1">Point 1</li>

<li class="point" id="point2">Point 2</li>

<Li class="point" id="point3">Point 3</Li>

</ul>

</div>

<button id="bouton">Cliquez-moi !</button>

</body>

</html>

Lorsqu'un programme JavaScript s'exécute dans le contexte d'un navigateur web, il peut accéder à la racine du DOM en utilisant la variable document.

La variable document correspond à l'élément<html>.

**La console**

Dans firefox pour afficher la console, utilisez le raccourci Ctrl + Maj + K.

Il est possible d’interagir avec la console à partir de javascript en utilisant la commande :

*console.log("A afficher");*

**Exercice 1 :**

Créer une page *maPage.html* dans laquelle vous copierez le code ci-dessus.  
Ajoutez le script ci-dessous.

*var h = document.head; // La variable h contient l'objet head du DOM*

*console.log(h);*

*var b = document.body; // La variable b contient l'objet body du DOM*

*console.log(b);*

Qu’observez-vous ?

**Le script affiche le doc head dans la console du navigateur**

**Accéder aux éléments**

Vous trouverez ci-dessous plusieurs instructions permettant d’interagir avec le DOM. Pour modifier, supprimer, ajouter des éléments.

// Accès au premier enfant du noeud body

console.log(document.body.childNodes[0]);

// Accès au deuxième enfant du noeud body

console.log(document.body.childNodes[1]);

// Accès au deuxième enfant du deuxième enfant du noeud body

console.log(document.body.childNodes[1].childNodes[1]);

// Affiche les noeuds enfant du noeud body

for (var i = 0; i < document.body.childNodes.length; i++) {

console.log(document.body.childNodes[i]);

}

// Sélection d’éléments selon la balise

var titresElts = document.getElementsByTagName("h2"); // Tous les titres h1

console.log(titresElts[0]); // Affiche le premier titre h1

console.log(titresElts.length);

// Sélection d’élément selon leur classe

// Tous les éléments ayant la classe "pointsClasse"

var pointsClasseElts = document.getElementsByClassName("pointsClasse");

for (var i = 0; i < pointsClasseElts.length; i++) {

console.log(pointsClasseElts[i]);

}

// Sélection d’élément selon leur identifiant

// Elément portant l'identifiant "pointsId"

console.log(document.getElementById("pointsId"));

// Le contenu HTML de l'élément identifié par "contenu"

console.log(document.getElementById("contenu").innerHTML);

// Le contenu textuel de l'élément identifié par "contenu"

console.log(document.getElementById("contenu").textContent);

// L'attribut href du premier lien

console.log(document.querySelector("a").getAttribute("href"));

// L'identifiant de la première liste

console.log(document.querySelector("ul").id);

// L'attribut href du premier lien

console.log(document.querySelector("a").href);

// Vérifier la présence d’un attribut

if (document.querySelector("a").hasAttribute("target")) {

console.log("Le premier lien possède l'attribut target");

} else {

console.log("Le premier lien ne possède pas l'attribut target");

}

// Modification du contenu HTML de la liste : ajout d'un point

document.getElementById("pointsId").innerHTML += '<li id="point">Point4</li>';

// Suppression du contenu HTML de la liste

document.getElementById("pointsId").innerHTML = "";

// Modification du contenu textuel du premier titre

document.querySelector("h1").textContent += " à moi !";

// Définition de l'attribut "id" du premier titre

document.querySelector("h1").setAttribute("id", "titre");

// Ajout d’un nouvel élément

var pt5Elt = document.createElement("li"); // Création d'un élément li

pt5Elt.id = "point5"; // Définition de son identifiant

pt5Elt.textContent = "Point 5"; // Définition de son contenu textuel

document.getElementById("pointsId").appendChild(pt5Elt); // Insertion du nouvel élément

// Ajout d’un nœud avant un autre nœud

var pt4Elt = document.createElement("li"); // Création d'un élément li

pt4Elt.id = "point4"; // Définition de son identifiant

pt4Elt.textContent = "Point 4"; // Définition de son contenu textuel

// Ajout du nouvel élément avant l'identifiant identifié par "point5"

document.getElementById("pointsId").insertBefore(pt4Elt,document.getElementById("point5"));

// Ajout d'un élément au tout début de la liste

document.getElementById('pointsId').insertAdjacentHTML("afterBegin",

'<li id="point0">Point 0</li>');

// Autres possibilités :

// afterbegin: juste à l'intérieur de l'élément existant, avant son premier enfant.

// beforeend: juste à l'intérieur de l'élément existant, après son dernier enfant.

// afterend: après l'élément existant lui-même.

// Remplacer un nœud

var pt6Elt = document.createElement("li"); // Création d'un élément li

pt6Elt.id = "point6"; // Définition de son identifiant

pt6Elt.textContent = "Point 6"; // Définition de son contenu textuel

// Remplacement de l'élément identifié par "point5" par le nouvel élément

document.getElementById("pointsId").replaceChild(pt6Elt, document.getElementById("point5"));

// Supprimer un noeud

// Suppression de l'élément identifié par "point6"

document.getElementById("PointsId").removeChild(document.getElementById("point6"));

//Modifier le style d’un élément

var pElt = document.querySelector("p");

pElt.style.color = "red";

pElt.style.margin = "50px";

pElt.style.fontFamily = "Arial";

pElt.style.backgroundColor = "black";

**Exercice 2 :**

Utilisez tous les éléments présents ci-dessous dans la page *maPage.html* pour vous familiariser avec le DOM

**Les événements**

Il est possible d’interagir directement avec la page web sans passer par le serveur.

Il est même possible de placer des écoutes sur les évènements afin de déclencher ces interactions.

Ci-dessous quelques exemples de captures d’évènements.

//Ajout d’un gestionnaire d’événement

var boutonElt = document.getElementById("bouton");

// Ajout d'un gestionnaire pour l'événement click

boutonElt.addEventListener("click", function () {

console.log("clic");

});

// Suppression du gestionnaire pour l'événement click

boutonElt.removeEventListener("click", clic);

// Gestion de l'appui sur une touche du clavier produisant un caractère

document.addEventListener("keypress", function (e) {

console.log("Vous avez appuyé sur la touche " + String.fromCharCode(e.charCode));

});

// Affiche des informations sur un événement clavier

function infosClavier(e) {

console.log("Evènement clavier : " + e.type + ", touche : " + e.keyCode);

}

// Gestion de l'appui et du relâchement d'une touche du clavier

document.addEventListener("keydown", infosClavier);

document.addEventListener("keyup", infosClavier);

///// Un exemple plus complet pour la gestion de la souris ///////////////////////////////////////////

// Renvoie le nom du bouton souris à partir de son code

function getBoutonSouris(code) {

var bouton = "inconnu";

switch (code) {

case 0: // 0 est le code du bouton gauche

bouton = "gauche";

break;

case 1: // 1 est le code du bouton du milieu

bouton = "milieu";

break;

case 2: // 2 est le code du bouton droit

bouton = "droit";

break;

}

return bouton;

}

// Affiche des informations sur un événement souris

function infosSouris(e) {

console.log("Evènement souris : " + e.type + ", bouton " +

getBoutonSouris(e.button) + ", X : " + e.clientX + ", Y : " + e.clientY);

}

// Gestion du clic souris

document.addEventListener("click", infosSouris);

///// Fin exemple plus complet pour la gestion de la souris ///////////////////////////////////////////

// Gestion de l'appui et du relâchement d'un bouton de la souris

document.addEventListener("mousedown", infosSouris);

document.addEventListener("mouseup", infosSouris);

// Gestion de la fin du chargement de la page web

window.addEventListener("load", function () {

console.log("Page entièrement chargée");

});

// Gestion de la fermeture de la page web

window.addEventListener("beforeunload", function (e) {

var message = "On est bien ici !";

e.returnValue = message; // Provoque une demande de confirmation (standard)

return message; // Provoque une demande de confirmation (certains navigateurs)

});

**Exercice 3 :**

Créez une page html nommée arcEnCiel.html

Dans cette page placez un texte « ARC EN CIEL » prenant la largeur de la page.

Sur chaque événement clavier ou souris. Changez la couleur d’un mot du texte ou du fond.

**Bravo ! Vous êtes désormais initié au DOM !**