

# TP - JAVASCRIPT - SÉQUENCE 5 & 6 & 7

## 🚧 Étape 1 : Manipulation du DOM

- Créez un fichier `dom_bom.js`

```
<script>
// Fonction pour sélectionner un élément par son ID et modifier son contenu
function modifyElementContent() {
    var elementToModify = document.getElementById('elementToModify');
    if (elementToModify) {
        elementToModify.textContent = "Le contenu a été modifié!";
    }
}

// Fonction pour ajouter un nouvel élément à un élément existant
function addNewElement() {
    var parentElement = document.getElementById('elementToModify');
    if (parentElement) {
        var newElement = document.createElement('p');
        newElement.textContent = "Nouvel élément ajouté!";
        parentElement.appendChild(newElement);
    }
}

// Fonction pour afficher un message dans la console lorsqu'un bouton est cliqué
function handleClickEvent() {
    console.log("Le bouton a été cliqué! Bienvenue Ousmane SANKHARE");
}

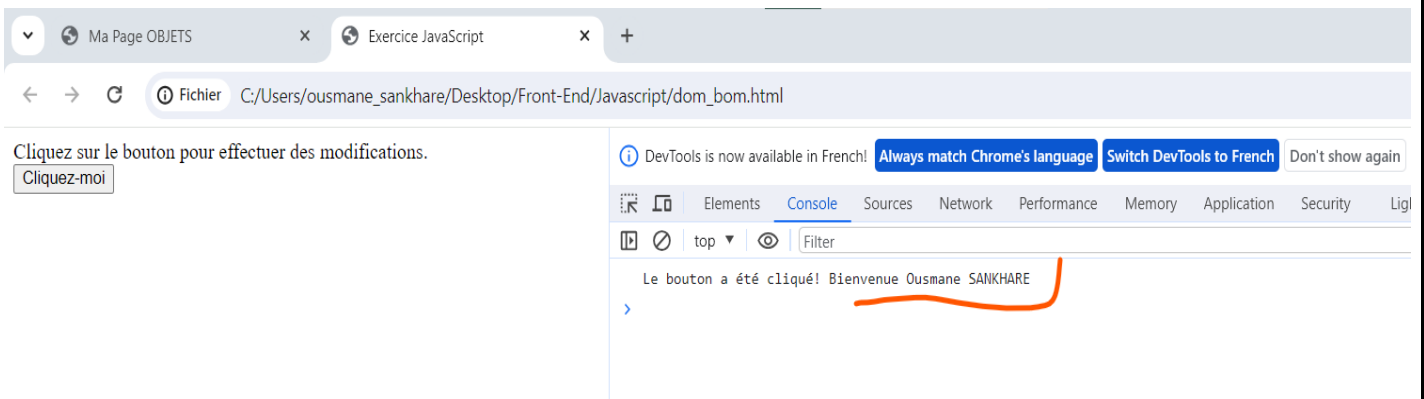
// Utilisation de addEventListener pour attacher un gestionnaire d'événements au bouton
var myButton = document.getElementById('myButton');
if (myButton) {
    myButton.addEventListener('click', handleClickEvent);
}
</script>
```

La fonction **modifyElementContent** sélectionne un élément par son ID ("elementToModify") et modifie son contenu en utilisant la propriété `textContent`.

La fonction **addNewElement** crée un nouvel élément (`<p>`) et l'ajoute à l'élément existant avec l'ID "elementToModify" en utilisant la méthode `appendChild`.

La fonction `handleClickEvent` est utilisée avec `addEventListener` pour afficher un message dans la console lorsque le bouton est cliqué.

## ➤ Exécution



## 🚦 Étape 2 : Interaction avec le BOM

```
<script>
// Fonction pour afficher la largeur et la hauteur de la fenêtre du navigateur
function afficherDimensionsFenetre() {
  var largeur = window.innerWidth || document.documentElement.clientWidth || document.body.clientWidth;
  var hauteur = window.innerHeight || document.documentElement.clientHeight || document.body.clientHeight;

  alert("Largeur de la fenêtre : " + largeur + " pixels\nHauteur de la fenêtre : " + hauteur + " pixels");
}

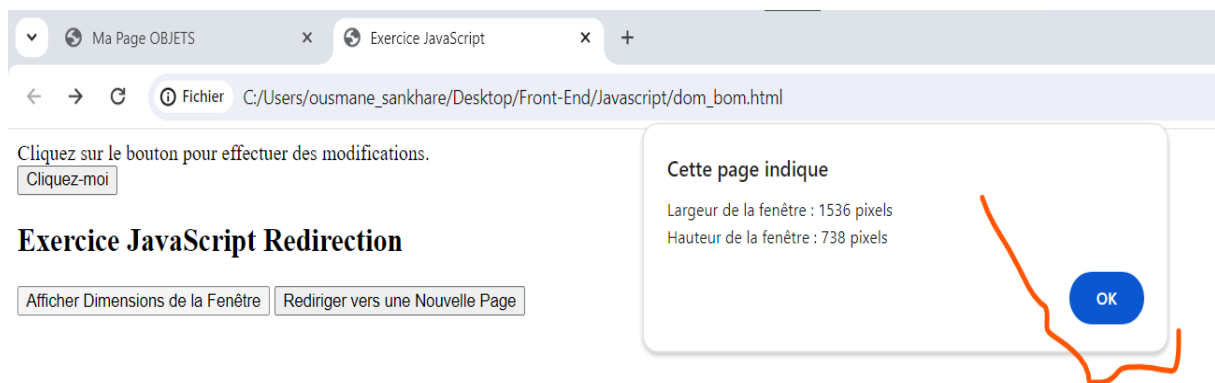
// Fonction pour rediriger l'utilisateur vers une nouvelle page
function redirigerVersNouvellePage() {
  // Remplacez "nouvelle_page.html" par l'URL de la nouvelle page
  window.location.href = "nouvelle_page.html";
}
</script>

<h2>Exercice JavaScript Redirection</h2>

<!-- Bouton pour afficher les dimensions de la fenêtre -->
<button onclick="afficherDimensionsFenetre()">Afficher Dimensions de la Fenêtre</button>

<!-- Bouton pour rediriger vers une nouvelle page -->
<button onclick="redirigerVersNouvellePage()">Rediriger vers une Nouvelle Page</button>
```

**afficherDimensionsFenetre** utilise l'objet **window** pour obtenir la largeur et la hauteur de la fenêtre du navigateur. La fonction **redirigerVersNouvellePage** utilise également l'objet **window** pour changer l'URL de la page, redirigeant ainsi l'utilisateur vers une nouvelle page. On peut personnaliser l'URL dans la fonction **redirigerVersNouvellePage** en remplaçant "nouvelle\_page.html" par l'URL réelle de la nouvelle page.



## ◆ Redirection



## ✚ Étape 3 : Manipulation d'événements

```

<form style="text-align: center;" id="myForm" onsubmit="showAlert()">
  <label for="inputText">Entrez du texte :</label>
  <input type="text" id="inputText" name="inputText" required>
  <br>
  <input type="submit" value="Soumettre">
</form>

<script>
  function showAlert() {
    alert("Formulaire soumis avec succès!");
    // Vous pouvez ajouter ici d'autres actions à effectuer lors de la soumission
  }
</script>

<div id="elementToChangeColor" onmouseenter="changeColor(true)" onmouseleave="changeColor(false)">
  Survolez-moi!
</div>

<script>
  function changeColor(isMouseOver) {
    var element = document.getElementById("elementToChangeColor");

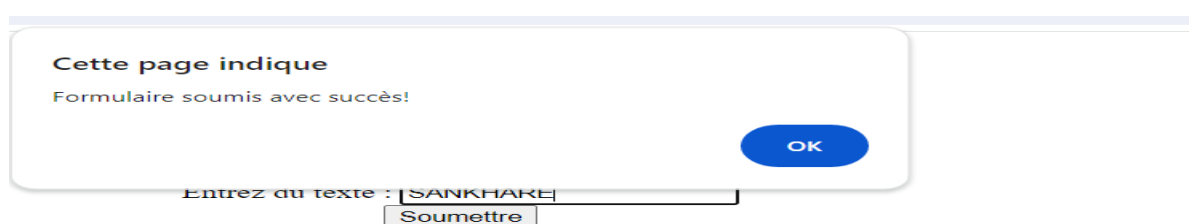
    if (isMouseOver) {
      element.style.backgroundColor = "#a0a0a0";
    } else {
      element.style.backgroundColor = "#e0e0e0";
    }
  }
</script>

```

## ➤ Exécution



## ➤ Soumission formulaire



## Envoi du formulaire par URL



Fichier C:/Users/ousmane\_sankhare/Desktop/Front-End/Javascript/dom\_bom.html?inputText=SANKHARE

### 🚦 Étape 4 : Expérimentation et utilisation avancée

Lorsque l'utilisateur clique sur le bouton, nous allons modifier certaines propriétés CSS pour créer une animation simple. Dans cet exemple, nous ferons glisser le bouton vers la droite.

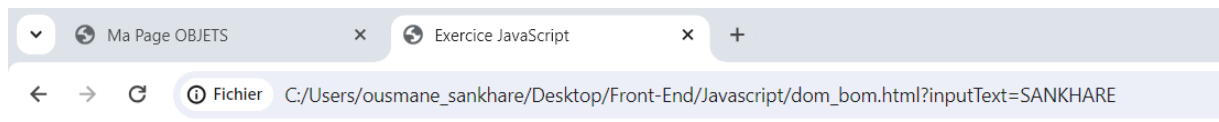
Nous utilisons la fonction `animateButton` déclenchée par le clic sur le bouton. Cette fonction modifie les propriétés CSS du bouton, provoquant une animation. La propriété `transition` dans le style CSS est utilisée pour créer une transition fluide. La fonction `set Timeout` est utilisée pour réinitialiser les propriétés après un certain délai, ce qui donne l'effet de retour à l'état initial après l'animation.

```
<button id="animatedButton" onclick="animateButton()">Cliquez-moi</button>

<script>
  function animateButton() {
    var button = document.getElementById("animatedButton");

    // Modification des propriétés CSS pour créer une animation
    button.style.backgroundColor = "#e74c3c"; // Changement de couleur
    button.style.transform = "translateX(100px)"; // Translation vers la droite

    // Réinitialisation des propriétés après un délai pour revenir à l'état initial
    setTimeout(function() {
      button.style.backgroundColor = "#3498db";
      button.style.transform = "translateX(0)";
    }, 1000); // Délai en millisecondes
  }
</script>
```



Cliquez-moi