

## Table des matières

|   |   |
|---|---|
| Section 2. Sélection des éléments .....             | 1 |
| Section 3. Éléments traversants.....                | 1 |
| Section 4. Manipulation des éléments .....          | 1 |
| Section 5. Travailler avec des attributs.....       | 2 |
| Section 6. Manipulation des styles d'éléments ..... | 2 |
| Section 7. Travailler avec des événements .....     | 2 |
| Section 8. Script de formulaires Web .....          | 3 |

## Section 2. Sélection des éléments

- [getElementById\(\)](#) – sélectionnez un élément par identifiant.
- [getElementsByName\(\)](#) – sélectionne les éléments par nom.
- [getElementsByTagName\(\)](#) – sélectionne les éléments par un nom de balise.
- [getElementsByClassName\(\)](#) – sélectionne les éléments par un ou plusieurs noms de classe.
- [querySelector\(\)](#) – sélectionnez les éléments par des sélecteurs CSS.

## Section 3. Éléments traversants

- [Obtenez l'élément parent](#) – obtenez le nœud parent d'un élément.
- [Obtenir des éléments enfants](#) – obtenez les enfants d'un élément.
- [Obtenez les frères et sœurs d'un élément](#) – obtenez les frères et sœurs d'un élément.

## Section 4. Manipulation des éléments

- [createElement\(\)](#) – crée un nouvel élément.
- [appendChild\(\)](#) – ajoute un nœud à une liste de nœuds enfants d'un nœud parent spécifié.
- [textContent](#) – récupère et définit le contenu textuel d'un nœud.
- [innerHTML](#) – récupère et définit le contenu HTML d'un élément.
- [innerHTML vs createElement](#) – expliquez les différences entre innerHTML et createElement lorsqu'il s'agit de créer de nouveaux éléments.
- [DocumentFragment](#) – apprenez à composer des nœuds DOM et à les insérer dans l'arborescence DOM active.

- [after\(\)](#) – insère un nœud après un élément.
- [append\(\)](#) – insère un nœud après le dernier nœud enfant d'un nœud parent.
- [prepend\(\)](#) – insère un nœud avant le premier nœud enfant d'un nœud parent.
- [insertAdjacentHTML\(\)](#) – analyse un texte au format HTML et insère les nœuds résultants dans le document à une position spécifiée.
- [replaceChild\(\)](#) – remplace un élément enfant par un nouvel élément.
- [cloneNode\(\)](#) – clone un élément et tous ses descendants.
- [RemoveChild\(\)](#) – supprime les éléments enfants d'un nœud.
- [insertBefore\(\)](#) – insère un nouveau nœud avant un nœud existant en tant que nœud enfant d'un nœud parent spécifié.
- [Fonction d'assistance insertAfter\(\)](#) – insère un nouveau nœud après un nœud existant en tant que nœud enfant d'un nœud parent spécifié.

## Section 5. Travailler avec des attributs

- [Attributs HTML et propriétés de l'objet DOM](#) – comprenez la relation entre les attributs HTML et les propriétés de l'objet DOM.
- [setAttribute\(\)](#) – définit la valeur d'un attribut spécifié sur un élément.
- [getAttribute\(\)](#) – récupère la valeur d'un attribut sur un élément.
- [removeAttribute\(\)](#) – supprime un attribut d'un élément spécifié.
- [hasAttribute\(\)](#) – vérifie si un élément a un attribut spécifié ou non.

## Section 6. Manipulation des styles d'éléments

- [propriété style](#) – obtient ou définit les styles en ligne d'un élément.
- [getComputedStyle\(\)](#) – renvoie le style calculé d'un élément.
- [Propriété className](#) – renvoie une liste de classes CSS séparées par des espaces.
- [Propriété classList](#) – manipule les classes CSS d'un élément.
- [Largeur et hauteur de l'élément](#) – obtenez la largeur et la hauteur d'un élément.

## Section 7. Travailler avec des événements

- [Événements JavaScript](#) – vous présentent les événements JavaScript, les modèles d'événements et comment gérer les événements.
- [Gestion des événements](#) – vous montre trois façons de gérer les événements en JavaScript.

- [Événements de chargement de page](#) – découvrez les événements de chargement et de déchargement de page.
- [événement de chargement](#) – vous guide à travers les étapes de gestion de l'événement de chargement provenant des éléments de document, d'image et de script.
- [DOMContentLoaded](#) – apprenez à utiliser DOMContentLoaded correctement l'événement.
- [Événement beforeunload](#) – vous explique comment afficher une boîte de dialogue de confirmation avant que les utilisateurs ne quittent la page.
- [événement de déchargement](#) – vous montre comment gérer l'événement de déchargement qui se déclenche lorsque la page est complètement déchargée.
- [Événements de souris](#) – comment gérer les événements de souris.
- [Événements de clavier](#) – comment gérer les événements de clavier.
- [Événements de défilement](#) – comment gérer efficacement les événements de défilement.
- [scrollIntoView](#) – apprenez à faire défiler un élément pour l'afficher.
- [Événements ciblés](#) – couvrent les événements ciblés.
- [Événement haschange](#) – découvrez comment gérer l'événement lorsque le hachage d'URL change.
- [Délégation d'événement](#) – est une technique permettant d'exploiter le bouillonnement d'événements pour gérer les événements à un niveau plus élevé dans le DOM que l'élément d'origine de l'événement.
- [dispatchEvent](#) – apprenez à générer un événement à partir du code et à le déclencher.
- [Événements personnalisés](#) – définissez un événement JavaScript personnalisé et attachez-le à un élément.
- [MutationObserver](#) – surveillez les modifications du DOM et invoquez un rappel lorsque les modifications se produisent.

## Section 8. Script de formulaires Web

- [Formulaire JavaScript](#) – découvrez comment gérer submit les événements de formulaire et effectuer une validation simple pour un formulaire Web.
- [Bouton radio](#) – vous montre comment écrire le JavaScript pour les boutons radio.
- [Case à cocher](#) – vous guide sur la façon de manipuler la case à cocher en JavaScript.
- [Zone de sélection](#) – découvrez comment gérer la zone de sélection et son option en JavaScript.
- [Ajouter/Supprimer des options](#) – vous montre comment ajouter et supprimer dynamiquement des options dans une zone de sélection.

- [Suppression d'éléments de l'élément <select> sous condition](#) - vous montre comment supprimer des éléments d'un <select>élément de manière conditionnelle.
- [Gestion de l'événement de changement](#) – découvrez comment gérer l'événement de changement du texte saisi, du bouton radio, de la case à cocher et des éléments de sélection.
- [Gestion de l'événement d'entrée](#) – gère l'événement d'entrée lorsque la valeur de l'élément d'entrée change.