

# INTERACTIONS ENTRE L'HOMME ET LA MACHINE SUR LE WEB : LANGAGE JAVASCRIPT



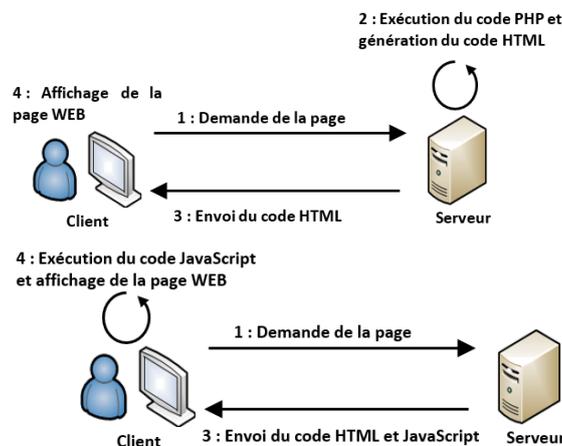
JavaScript est un langage créé en **1995** par **Brendan Eich** qui travaille chez Netscape Communication Corporation principal concurrent à l'époque d'Internet Explorer. JavaScript est alors transmis à l'ECMA (European Computer Manufacturers Association) pour standardisation sous la dénomination d'ECMAScript abrégé en ES. JavaScript dont le nom est une référence à Java de Sun Microsystems, ne doit pas être confondu avec Java. En effet, Java est fortement typé et précompilé alors que JavaScript est à typage dynamique et interprété. JavaScript est un langage orienté objet.

## 1 – LE LANGAGE JAVASCRIPT

### 1.1 – Présentation

Les pages créées à l'aide des langages HTML et CSS sont des **pages WEB statiques** c'est-à-dire que l'interaction avec l'utilisateur est réduite à la possibilité de cliquer sur un lien hypertexte présent dans la page. Il est possible de rendre une page internet dynamique, de deux manières :

- soit du **côté serveur** avec le langage **PHP** (ou Hypertext Preprocessor) qui peut, par exemple, ajouter le résultat d'une requête à une base de données dans la page qui sera fournie au navigateur ;
- soit du **côté client** avec **JavaScript** qui peut, par exemple, faire apparaître des info-bulles contextuelles ou réaliser des animations.



Le JavaScript est donc un langage dit **client-side**, c'est-à-dire que les scripts sont exécutés par le navigateur chez l'internaute (le **client**).



D'un point de vue historique, JavaScript a été principalement défini pour être exécuté dans un navigateur web. Pour autant, depuis quelques années, il a gagné en popularité et est aujourd'hui utilisé dans bien d'autres contextes (programmation côté serveur, scripts et macros dans des applications, etc.).

JavaScript est un **langage interprété**. Cela veut dire qu'il est exécuté par un interpréteur sans qu'une phase de compilation soit nécessaire. **NodeJS** est l'interpréteur le plus populaire de JavaScript. Tout navigateur web dispose d'un interpréteur JavaScript et peut ainsi exécuter du code JavaScript. Lorsqu'une page HTML référence du code JavaScript, celui-ci est téléchargé par le navigateur et interprété par l'interpréteur.

## 1.2 – Intégration de code javascript dans une page HTML

Le code javascript peut être intégré directement dans le document html grâce à la balise `<script>` au sein de l'entête (entre les balises `<head>` et `</head>`). Il peut également être placé dans le corps (entre les balises `<body>` et `</body>`) de la page HTML.

### exemple1.html

```
<html>
  <head>
    <meta charset="utf-8" />
    <title>Un peu de javascript</title>
    <script type="text/javascript" charset="utf-8">
      // Je suis un commentaire javascript!
      // Alert crée une fenêtre d'affichage sur l'écran pour l'utilisateur (popup)
      alert('Bienvenue en NSI !') ;
    </script>
  </head>
  <body>
    <h1>Découverte du langage JavaScript</h1>
    <p>Ma première page dynamique (quoique !!!).</p>
  </body>
</html>
```

#### Exemple 1 : Intégration du code JavaScript dans le code HTML

1. **Lancer** le logiciel **Visual Studio Code**. **Créer** un nouveau document nommé **exemple1.html**.
2. **Editer** le code ci-dessus dans le document **exemple1.html**. **Sauvegarder** le script.
3. **Ouvrir** le fichier **exemple1.html** à l'aide d'un navigateur WEB.
4. **Vérifier** et **justifier** le l'affichage obtenu.

Le code javascript peut être également intégré dans un fichier séparé possédant l'extension **.js**. Il est alors nécessaire de l'importer dans la page html à l'aide de l'attribut `src` la balise `<script>`.



## exemple2.html

```
<html>
  <head>
    <meta charset="utf-8" />
    <title>Un peu de javascript</title>
    <script src="text/scriptEx2.js"></script>
  </head>
  <body>
    <h1>Découverte du langage JavaScript</h1>
    <p>Ma première page dynamique (quoi que !!!).</p>
  </body>
</html>
```

## scriptEx2.js

```
alert('Bienvenue en NSI !') ;
```

### Exemple 2 : Intégration du code JavaScript dans un fichier .js

1. **Créer** dans, **Visual Studio Code**, un nouveau document nommé **exemple2.html** et **éditer** le code html ci-dessus. **Sauvegarder** le fichier.
2. **Créer** dans, **Visual Studio Code**, un nouveau document nommé **scriptEx2.js** et **éditer** le code Javascript ci-dessus. **Sauvegarder** le fichier.
3. **Ouvrir** le fichier **exemple2.html** à l'aide d'un navigateur WEB.
4. **Vérifier** que l'affichage obtenu correspond à celui attendu.

## 1.3 – Syntaxe de JavaScript

Voici quelques bases de la syntaxe de JavaScript.

- Les variables doivent être déclarées grâce au mot clé **var**.
- Tout comme en python, les **chaînes de caractères** sont entourées de **guillemets**.
- Les instructions simples sont terminées par un **point-virgule ;**.
- Les **blocs d'instructions** sont entourés d'**accolades**.
- Les **commentaires** sont notés précédés de deux barres obliques: **//**.
- L'indentation des blocs d'instruction n'est pas obligatoire comme en Python, mais souhaitable.

La référence la plus complète et à jour est le site Mozilla [Developer Network MDN](https://developer.mozilla.org/).



## 2 – MODIFIER UNE PAGE WEB AVEC JAVASCRIPT

### exemple3.html

```
<html>
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="style.css">
    <title>Exemple 3</title>
    <script>
      var prenom = prompt('Quel est votre prénom ?');
      alert('Bonjour ' + prenom);
    </script>
  </head>
  <body>
    <h1>Page de salutations</h1>
  </body>
</html>
```

### Exemple 3

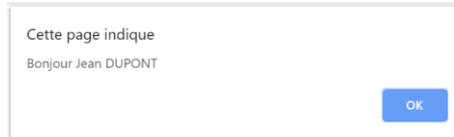
1. **Créer** dans, **Visual Studio Code**, un nouveau document nommé **exemple3.html** et **éditer** le code html ci-dessus. **Sauvegarder** le fichier.
2. **Créer** un nouveau document nommé **style.css** et **éditer** le code afin d'avoir le titre h1 en rouge et centré.
3. **Ouvrir** le fichier **exemple3.html** à l'aide du navigateur WEB.
4. **Vérifier** et **justifier** l'affichage obtenu.

### exemple4.html

```
<html>
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="style.css">
    <title>Exemple 4</title>
    <script src="scriptEx4.js"></script>
  </head>
  <body>
    <h1>Page de salutations bis</h1>
  </body>
</html>
```

### Exemple 4

1. **Editer** le code html ci-dessus. **Sauvegarder** le fichier.
2. **Créer** un nouveau document nommé **scriptEx4.js** et **éditer** le code afin de demander à l'utilisateur son prénom et son nom et de le saluer.



3. **Ouvrir** le fichier **exemple4.html** à l'aide d'un navigateur WEB.
4. **Vérifier** et **justifier** l'affichage obtenu.
5. **Ajouter** les lignes de code suivantes au fichier **scriptEx4.js**.

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

6. **Actualiser** la page WEB (Touche F5) et **justifier** pourquoi l'affichage obtenu n'est pas satisfaisant.
7. **Modifier** la ligne de code suivante.

```
var age = parseInt(prompt('Quel est votre age'));
```

8. **Actualiser** la page WEB et **vérifier** que l'affichage obtenu est, cette fois-ci, satisfaisant.

## 3 – GESTION D'ÉVÉNEMENTS AVEC JAVASCRIPT

### 3.1 – Fonctions

Comme dans les autres langages, il peut être intéressant de placer les instructions dans des fonctions. Pour cela il faut utiliser le mot clé **function**. Ces fonctions sont particulièrement intéressantes pour la gestion des événements.

#### exemple5.html

```
<html>  
  <head>  
    <meta charset="UTF-8">  
    <link rel="stylesheet" href="style.css">  
    <script src="scriptEx5.js"></script>  
    <title>Exemple 5</title>  
  </head>  
  <body onload="salutation();">  
    <h1>Encore une page de salutations</h1>  
  </body>  
</html>
```



### scriptEx5.html

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

#### Exemple 5

1. **Editer** le code html et le code JavaScript ci-dessus. **Sauvegarder** les fichiers.
2. **Ouvrir** le fichier **exemple5.html** à l'aide d'un navigateur WEB.
3. **Vérifier** et **justifier** l'affichage obtenu.

## 3.2 – Evènements

Un évènement est par exemple le chargement d'une page, le passage du pointeur sur la page ou un élément de la page, un clic de la souris. Un événement permet de déclencher l'exécution d'une fonction. Pour cela un évènement est ajouté à un élément de la page, une balise par exemple, et lorsque l'utilisateur agit sur l'évènement d'une certaine manière, il déclenche le code JavaScript.

Dans l'exemple précédent, la fonction JavaScript `salutation()` est exécutée au chargement de la page grâce à l'attribut `onLoad` correspondant à l'évènement Load (Chargement).

Il existe un certain nombre d'évènements dont les principaux sont les suivants :

<b>Load</b>	Chargement d'un page
<b>Click</b>	Clic de la souris, sélection d'un élément
<b>Dblclick</b>	Double-clic sur l'élément
<b>Mouseover</b>	Entrée du pointeur sur un élément
<b>Mouseout</b>	Sortie du pointeur de l'élément
<b>KeyPress</b>	Appui sur une touche produisant un caractère
<b>Focus</b>	Ciblage de l'élément
<b>Blur</b>	Annulation du ciblage de l'élément
<b>Submit</b>	Envoi d'un formulaire
<b>Reset</b>	Réinitialisation d'un formulaire
<b>Change</b>	Modification de la valeur d'un élément d'un formulaire



### 3.3 – Gestion d'un évènements

L'utilisateur déclenche un évènement qui réagit en appelant une fonction JavaScript. Le code s'exécute et provoque une modification dans la page. Les attributs associés aux éléments sont formés du mot « on » et du nom de l'évènement : `onLoad`, `onClick`, `onmouseover`...

#### exemple6.html

```
<html>
  <head>
    <meta charset="UTF-8">
    <script src="scriptEx6.js"></script>
    <title>Exemple 6</title>
  </head>
  <body>
    <h1 onClick="message()">Cliquez ici</h1>
    <h2 onDbclick="message()">Double-cliquez ici</h2>
    <h3 onmouseover="message()">Approchez le curseur</h3>
    <h3 onmouseout="message()">Eloignez le curseur</h3>
    <p>Tapez sur une touche</p><input type="text" onkeypress="message()">
  </body>
</html>
```

#### scriptEx6.html

```
function message() {
  alert("Un évènement a été détecté");
}
```

#### Exemple 6

1. **Editer** le code HTML et le code JavaScript ci-dessus. **Sauvegarder** les fichiers.
2. **Ouvrir** le fichier **exemple6.html** à l'aide d'un navigateur WEB.
3. **Vérifier** et **justifier** l'affichage obtenu.

## 4 – RECUPERATION DES DONNEES UTILISATEUR

La balise html `<input>` permet de récupérer les entrées utilisateur, il en existe de divers types :

- `<input type="text"…>` : entrée de texte (par défaut).
- `<input type="button"…>` : bouton.
- `<input type="checkbox"…>` : case à cocher.
- `<input type="radio"…>` : bouton radio qui permet de sélectionner une seule valeur parmi un groupe de différentes valeurs.



### exemple7.html

```
<html>
  <head>
    <meta charset="utf-8" />
    <title>Exemple 8</title>
    <script src="scriptEx8.js"></script>
  </head>
  <body>
    <!-- Zone de texte -->
    <input id="id1" type="text" placeholder="Taper votre texte ici" />
    <!-- Bouton à cliquer -->
    <input type="button" onclick="afficheEntrée();" value="Clique ici" />
  </body>
</html>
```

### scriptEx7.html

```
function afficheEntrée() {
  // On récupère la valeur du texte rentré sur l'élément input avec l'id "id1"
  const texte = document.querySelector('#id1').value
  alert('Affichage avec alert: ' + texte)
}
```

#### Exemple 7

1. **Editer** le code HTML et le code JavaScript ci-dessus. **Sauvegarder** les fichiers.
2. **Ouvrir** le fichier **exemple7.html** à l'aide d'un navigateur WEB.
3. **Vérifier** et **justifier** l'affichage obtenu.

## 5 – MODIFIER UN ELEMENT HTML

### 5.1 – Modifier le contenu

Il existe plusieurs méthodes pour avoir accès à un élément HTML. La méthode `querySelector()`, appliquée à partir de la racine `document`, permet d'accéder au premier élément rencontré.

Pour modifier le contenu d'un élément il faut utiliser la propriété `innerHTML`.



### Exemple8.html

```
<html>
  <head>
    <meta charset="utf-8" />
    <title>Exemple 8</title>
    <script src="scriptEx8.js"></script>
  </head>
  <body>
    <h1> Modification du contenu d'un élément HTML</h1>
    <p id="texte"> Ce texte ne me plait pas, je vais le modifier </p>
    <input type="button" onclick="modifTexte();" value="Cliquer ici pour modifier le
    texte"/>
  </body>
</html>
```

### scriptEx8.html

```
function modifTexte() {
  document.querySelector('#texte').innerHTML = "Je n'aimais pas l'ancien texte, je l'ai
changé."
}
```

### Exemple 8

1. **Editer** le code HTML et le code JavaScript ci-dessus. **Sauvegarder** les fichiers.
2. **Ouvrir** le fichier **exemple8.html** à l'aide d'un navigateur WEB.
3. **Vérifier** et **justifier** l'affichage obtenu.

## 5.2 – Modifier le style

Pour modifier les propriétés de style d'un élément HTML il faut utiliser la propriété style.

### exemple9.html

```
<html>
  <head>
    <meta charset="utf-8" />
    <title>Exemple 9</title>
    <link rel="stylesheet" href="style.css">
    <script src="scriptEx9.js"></script>
  </head>
  <body>
    <h1 id="titre"> Modification du style d'un élément HTML </h1>
    <p id="texte"> Ces couleurs ne me plaisent pas, je vais les modifier </p>
    <input type="button" onclick="modifStyle();" value="Clique ici pour modifier
    les couleurs" />
  </body>
</html>
```



### style.css

```
h1{color: red;text-align: center;}
p{color: blue;text-align: justify;}
```

### scriptEx9.html

```
function modifStyle() {
  document.querySelector('#titre').style.color = "#17657D"
  document.querySelector('#texte').style.color = "#B233FF"
}
```

### Exemple 9

1. **Editer** les codes HTML, CSS et JavaScript ci-dessus. **Sauvegarder** les fichiers.
2. **Ouvrir** le fichier **exemple9.html** à l'aide d'un navigateur WEB.
3. **Vérifier** et **justifier** l'affichage obtenu.

Il est possible de travailler plus "proprement" en utilisant les classes CSS.

### exemple10.html

```
<html>
  <head>
    <meta charset="utf-8" />
    <title>Exemple 10</title>
    <link rel="stylesheet" href="style.css">
    <script src="scriptEx10.js"></script>
  </head>
  <body>
    <h1> Modification du style d'un élément HTML </h1>
    <p id="texte"> Ces couleurs ne me plaisent pas, je vais les modifier </p>
    <input type="button" onclick="foncRouge();" value="Rouge" />
    <input type="button" onclick="foncVert();" value="Vert" />
  </body>
</html>
```

### style.css

```
h1{ text-align: center;}
.rouge { color:red; font-size:20px;}
.vert { color:green; font-size:30px;}
```

### scriptEx10.html

```
function foncRouge() {
  document.querySelector("#texte").classList.remove("vert");
  document.querySelector("#texte").classList.add("rouge");
}
function foncVert() {
  document.querySelector("#texte").classList.remove("rouge");
  document.querySelector("#texte").classList.add("vert");
}
```



### Exemple 10

1. **Editer** les codes HTML, CSS et JavaScript ci-dessus. **Sauvegarder** les fichiers.
2. **Ouvrir** le fichier **exemple10.html** à l'aide d'un navigateur WEB.
3. **Vérifier** et **justifier** l'affichage obtenu.

### exemple11.html

```
<html>
  <head>
    <meta charset="utf-8" />
    <title>Exemple 10</title>
    <link rel="stylesheet" href="style.css">
    <script src="scriptEx10.js"></script>
  </head>
  <body>
    <h1> Modification du style d'un élément HTML </h1>
    <div onmouseover="foncEntre()" onmouseout="foncQuitte()" id="id1"><p>Survolez
    moi</p></div>
  </body>
</html>
```

### style.css

```
h1{ text-align: center; }
p{ text-align : center; }
#id1 { width : 200px; height : 100px;
      margin : 0 auto; border : 2px solid black; }
.rouge { background-color:red; }
.blanc { background-color : white; }
```

### scriptEx11.html

```
function foncEntre(){
  document.querySelector("#id1").classList.remove("blanc");
  document.querySelector("#id1").classList.add("rouge");
}
function foncQuitte() {
  document.querySelector("#id1").classList.remove("rouge");
  document.querySelector("#id1").classList.add("blanc");
}
```

### Exemple 11

1. **Editer** les codes HTML, CSS et JavaScript ci-dessus. **Sauvegarder** les fichiers.
2. **Ouvrir** le fichier **exemple11.html** à l'aide d'un navigateur WEB.
3. **Vérifier** et **justifier** l'affichage obtenu.



## 6 – EXERCICE

### Exercice

On veut réaliser une page dynamique avec le fonctionnement suivant :

- **Affichage au démarrage ou lors du clic sur le bouton « Autre » :**  
**Bienvenue sur ma page dynamique**

Cette page s'adapte dynamiquement à votre réponse

Garçon  Fille  Autre

Bonjour

- **Affichage lors du clic sur le bouton « Fille » :**  
**Bienvenue sur ma page dynamique**

Cette page s'adapte dynamiquement à votre réponse

Garçon  Fille  Autre

Bonjour **Madame**

- **Affichage lors du clic sur le bouton « Garçon » :**  
**Bienvenue sur ma page dynamique**

Cette page s'adapte dynamiquement à votre réponse

Garçon  Fille  Autre

Bonjour **Monsieur**

1. **Editer** les fichiers HTML, CSS et JavaScript permettant d'obtenir la page dynamique ci-dessus.
2. **Vérifier** le fonctionnement de la page WEB.