

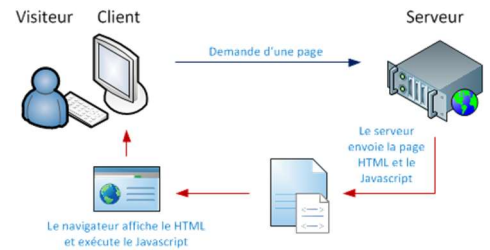
## 1. Le langage JavaScript

Les langages HTML et CSS permettent de créer des pages Web statiques, mais il n'est pas possible d'interagir avec. Le JavaScript permet de rendre les **pages Web interactives**.



Le JavaScript a été créé par Brendan Eich en 1995. Il est devenu incontournable dans le développement Web.

Le JavaScript est un langage dit **client-side** (comme le langage HTML), c'est-à-dire que les scripts sont exécutés par le navigateur chez l'internaute (le client). Cela diffère des langages de scripts dits **server-side** qui sont exécutés par le serveur Web. C'est le cas des langages comme le PHP.



## 2. Interactions

Les interactions qui suivent vont permettre de rendre interactives la page `lovelace.html` créée précédemment.

### 2.1. Apparition d'une fenêtre

✂ Dans le fichier `lovelace.html`, insérer dans la balise `<head>` la ligne suivante : `<script src="script.js"></script>`. Le code permet d'exécuter le code JavaScript contenu dans le fichier `script.js`.

✂ Ouvrir un nouveau fichier avec le nom **script.js**.

✂ Insérer le code suivant : `alert("Voici une page sur Ada Lovelace !")` et actualiser la page `lovelace.html`.

L'instruction `"alert"` est exécutée dès l'ouverture de la page web. Elle provoque l'apparition d'une fenêtre (pop-up) en plus de la page Web. Il est tout à fait possible de lier l'exécution de certaines instructions JavaScript à l'action d'un utilisateur (par exemple un clic sur un bouton).

### 2.2. Prise en compte d'un évènement à partir d'un bouton

✂ Dans la page HTML, insérer dans la balise `<nav>` le code suivant : `<button onclick="maFonction()">Cliquer ici</button>`.

✂ Modifier le JavaScript comme suit :

```
function maFonction() {
  alert("Voici une page sur Ada Lovelace !")
}
```

✂ Tester à nouveau la page Web.

L'instruction `"alert"` n'est plus exécutée à l'ouverture de la page web, mais uniquement dans le cas où l'utilisateur clique sur le bouton.

Dans la page HTML, on a associé au bouton un évènement `"onclick"`, en cas de clic sur la souris, la fonction JavaScript `"maFonction()"` est exécutée.

Dans la page JavaScript, on retrouve bien une fonction `"maFonction()"` ("`function maFonction(){...}`" en JavaScript est équivalent à un `"def maFonction() :"` en Python). Entre l'accolade ouvrante et l'accolade fermante (qui délimite la fonction), on retrouve uniquement notre instruction `"alert"`. À l'ouverture de la page web cette instruction `"alert"` n'est pas exécutée, car elle se trouve dans une fonction. Le clic sur le bouton entraîne l'exécution de la fonction `"maFonction()"` et donc de l'instruction `"alert"`.

Il est possible de modifier le style d'une balise, de modifier la classe (CSS) d'une balise ou encore de modifier le contenu d'une balise, voici quelques exemples.

## 2.3. Modification d'un contenu HTML

✂ Dans la page HTML, compléter la première balise `h3` afin d'obtenir ceci :

```
<h3 id="tit_parag">La première programmeuse de l'histoire</h3>
```

✂ Mettre après la balise `footer` le code suivant : `<button onclick="newTitle()">Nouveau titre</button>`

✂ Dans la page JavaScript, insérer la fonction suivante :

```
function newTitle() {  
    document.getElementById("tit_parag").innerHTML = "Pionnière de la science informatique" ;  
}
```

Dans le code HTML un identifiant `id` ("tit\_parag") a été ajouté à la balise `p`.

Dans le code JavaScript, `innerHTML` permet d'écrire le contenu HTML ("Pionnière de la science informatique") de l'élément `tit_parag` identifié par `getElementById()` contenu dans la page Web (`document`).

✂ Rajouter un autre bouton pour afficher le titre d'origine.

## 2.4. Modification d'une image

Charger dans votre répertoire de travail une autre image d'Ada Lovelace (pour l'exemple ci-dessous le nom est `ada.jpg`).

✂ Dans la page HTML, compléter la première balise `h3` afin d'obtenir ceci :

```

```

✂ Mettre après la balise `footer` le code suivant : `<button onclick="newImage()">Nouvelle Image</button>`

✂ Dans la page JavaScript, insérer la fonction suivante :

```
function newImage() {  
    document.getElementById("ada").src = "ada.jpg" ;  
}
```

La fonction `newImage()` permet de changer le `src` de l'image dans la balise `img`.

✂ Rajouter un autre bouton pour afficher l'image d'origine.

✂ Modifier le CSS de la balise `img` pour définir une hauteur d'image fixe.

Il est aussi possible de créer un bouton pour alterner les images.

✂ Créer un nouveau bouton pour alterner les images et lui associer la fonction suivante :

```
function altImage() {  
    x = document.getElementById("ada").getAttribute("src");  
    if (x == "ada-lovelace.png")  
        x = "ada.jpg";  
    else  
        x = "ada-lovelace.png";  
    document.getElementById("ada").src = x;  
}
```

La fonction `altImage()` lit la valeur de l'attribut `src` puis le compare au nom de la première image. Suivant le résultat de la comparaison, l'attribut est changé pour afficher l'autre image.

Il est possible de le faire automatiquement.

✂ Insérer la ligne suivante dans le code JavaScript : `window.setInterval("altImage()", 2000);`

La fonction `altImage()` est exécuté toutes les 2 000 millisecondes.

Ainsi dans une fonction en **JavaScript**, il est possible de faire des tests, de se servir de variable, de créer des boucles... car c'est un **langage de programmation**.

## 2.5. Mise au point d'un programme JavaScript

Pour la mise au point, il est possible d'utiliser la console.

✂ À la fin de la fonction `altImage()` insérer la commande suivante : `console.log(x)`; et afficher la fenêtre d'inspection (clic droit Inspecter sur la page HTML) et afficher l'onglet Console.

La variable `x` s'affiche dans la console à chaque exécution de la fonction.

## 2.6. Les commentaires

Sur une ligne :

```
// ceci est un commentaire sur une ligne
```

Sur plusieurs lignes :

```
/* ceci est un commentaire  
sur plusieurs lignes */
```

## 2.7. Modifier du CSS

☞ Dans le fichier CSS, rajouter les lignes suivantes :

```
.red{  
  color: red;  
  font-family: Comic Sans MS  
}
```

☞ Et dans le fonction `newTitle()` du Javascript, le code suivant :

```
document.getElementById("tit_parag").className="red" ;
```

Lors du clic sur le bouton, le CSS de la class `.red` est appliqué.

☞ Réaliser les modifications nécessaires pour obtenir le CSS d'origine lors de l'appui sur le bouton Ancien titre.

Deux solutions à ce problème :

- soit appliquer à nouveau le style h3 : `document.getElementById("tit_parag").className="h3" ;`
- soit supprimer la class : `document.getElementById("tit_parag").className="" ;`

Il est aussi possible d'appliquer directement un style.

☞ Dans la page HTML, compléter la balise `h2` avec l'identifiant `id="tit_h2"`.

☞ Dans le fonction `newTitle()` du Javascript, insérer code suivant :

```
document.getElementById("tit_h2").style.color="blue" ;
```

Lors du clic sur le bouton, la couleur du titre h2 devient bleu.

Pour obtenir le CSS d'origine lors de l'appui sur le bouton Ancien titre, les codes suivant ne fonctionnent pas.

```
document.getElementById("tit_h2").className="h2" ;  
document.getElementById("tit_h2").className="" ;
```

L'application de style est prioritaire sur la class h2.

Par conséquent, il faut d'abord supprimer l'attribut au style avec :

```
document.getElementById("tit_h2").style.color="" ;
```

## 2.8. Insérer un élément

☞ Créer un bouton Ajout qui exécute la fonction `ajout()` ;

☞ Ajouter la fonction `ajout()` au code JavaScript.

```
function ajout() {  
  //l'élément photo de type img est créé  
  var photo = document.createElement("img");  
  //le nom de l'image est indiqué  
  photo.src = "ada-lovelace.png";  
  //l'image est affichée dans la page HTML  
  document.getElementById("tit_parag").appendChild(photo);  
}
```

Dans le HTML, il est aussi possible d'insérer l'élément à un endroit localisé par une `div`.

☞ Placer le code suivant dans le HTML et modifier la fonction pour faire apparaître la photo à l'endroit de la `div`.

```
<div id="fig1">  
</div>
```

Pour aller plus loin : <https://openclassrooms.com/fr/courses/146276-tout-sur-le-javascript/146274-syntaxe-en-javascript>