

Chapitre 2

Interactions avec l'utilisateur dans une page WEB



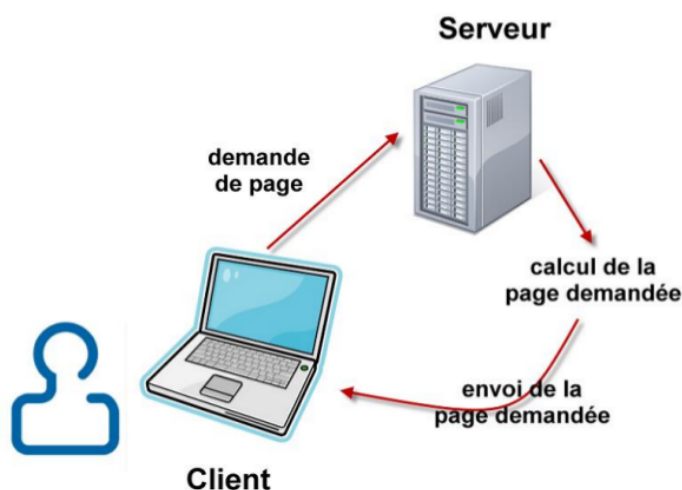
2.1 Interactions sur le WEB

Pour créer une page WEB, on utilise les langages HTML et CSS, qui permettent de gérer le squelette de notre page WEB (HTML) ainsi que sa mise en forme, son design (CSS). Ce sont des **langages de description**, car ils indiquent au navigateur la manière d'afficher les différentes informations. Ces deux langages sont cependant insuffisants.

En effet, les pages créées avec HTML et CSS sont dites **statiques**. L'utilisation de CSS permet d'animer graphiquement nos pages (souligner un texte au passage de la souris) mais il est **impossible pour l'utilisateur d'interagir** avec le site WEB.

Lorsque vous ouvrez une page WEB avec votre navigateur, vous contactez un ordinateur un peu spécial, dont le rôle est de vous transmettre les fichiers HTML et CSS correspondants au site WEB visité. Cet ordinateur particulier est appelé un **serveur**. Et il porte bien son nom : son rôle est de vous **servir** des pages WEB.

Votre navigateur, c'est le **client**. C'est lui qui reçoit le code HTML et l'affiche correctement.



Imaginons l'incroyable site WEB `calculator.com`, sur lequel l'utilisateur peut réaliser toutes sortes de calculs. Supposons que l'utilisateur souhaite connaître le résultat du calcul `2+2`. Peut-on faire cela avec HTML et CSS ?

La réponse est **non**, car rappelez-vous : HTML et CSS sont des langages de description, et non des langages de programmation. Or, tout calcul, aussi simple soit-il, nécessite l'emploi d'un langage de programmation.

Deux solutions s'offrent à nous :

1. Demander **au serveur** de faire les calculs, et utiliser un langage de programmation serveur, comme le **langage PHP**
2. Demander **au client** de faire les calculs, et utiliser un langage de programmation client, comme le **langage Javascript**

Dans notre cas, on pourrait simplement demander au client (le navigateur) de faire les calculs.

▷ À quoi peut bien servir un langage serveur comme PHP ?

Le langage PHP permet de **générer des pages HTML** de façon **dynamique**.

Un exemple très simple est l'affichage de l'heure sur un site WEB. Lorsque le client demande une page au serveur, celui-ci génère une page HTML en inscrivant dans celle-ci l'heure courante (l'heure sur le serveur) et envoie cette page au client.

Un autre exemple est l'utilisation de **formulaires**. Prenons un moteur de recherche comme DuckDuckGo¹, « le moteur de recherche qui ne vous espionne pas ».



Lorsque vous recherchez une information, vous utilisez le formulaire à votre disposition, ici un **champ texte** bien visible. En cherchant le mot-clé *informatique*, on peut voir que l'adresse demandée par le navigateur change :



On remarque notamment le `q=informatique` correspondant à notre recherche (`q` signifie *query* pour *requête* en anglais). On demande ici au serveur d'afficher la page `duckduckgo.com` **en lui transmettant des informations** : le serveur génère alors la page HTML correspondant à notre recherche et notre navigateur s'occupe de l'affichage.

Bien entendu, l'utilisation des formulaires ne se limite pas à la recherche. Ils permettent de recueillir des informations sur l'utilisateur (comme ses coordonnées, son adresse mail...).

Enfin, PHP est un langage très puissant, permettant entre autres :

- d'établir un compteur de visites
- de gérer un espace membre
- de générer des images
- d'enregistrer des informations fournies par l'utilisateur dans une *base de données*

1. Plus d'informations : <https://fr.wikipedia.org/wiki/DuckDuckGo>. Anecdote amusante : son siège social est situé à *Paoli* (États-Unis)

2.2 Le langage Javascript

Dans le cadre de ce cours, nous allons laisser PHP de côté et nous intéresser au **langage Javascript**.

Javascript est un langage de programmation **client**, nous n'allons donc transmettre aucune information au serveur mais plutôt manipuler le HTML uniquement via le navigateur. Aucun site WEB actuel ne saurait se passer de Javascript.

Grâce à Javascript, on peut par exemple :

- Afficher des messages d'informations
- Afficher l'heure en temps réel sur une page WEB
- Modifier instantanément l'apparence d'une page WEB
- Créer une calculatrice en ligne
- Créer des jeux simples

Nous allons mettre en oeuvre plusieurs exemples en mélangeant HTML, CSS et Javascript. Vous trouverez ces exemples à l'adresse suivante (dossier ZIP) :

<http://nsi.dellasantina.corsica/documents/web/javascript.zip>

Question 01 Télécharger le dossier ZIP ci-dessus et le décompresser à l'endroit de votre choix.

Chaque exemple se décompose en trois fichiers :

- un fichier `.html` contenant les différents éléments de la page WEB
- un fichier `.css` pour la mise en forme de la page WEB
- un fichier `.js` contenant le code Javascript

Le chargement du fichier Javascript se fait dans le fichier HTML, généralement dans les dernières lignes du fichier, à l'aide d'une balise `<script>` de la manière suivante :

```
<script type="text/javascript" src="code.js"></script>
```

Pour exécuter le code Javascript, il suffit d'ouvrir la page HTML avec votre navigateur.

2.2.1 Premier programme Javascript : afficher un message

Question 02 Ouvrir le dossier `1.Afficher un message` puis le fichier `page.html` avec votre navigateur préféré.

Question 03 Ouvrir les trois fichiers `page.html`, `style.css` et `code.js` avec Notepad++ ou tout autre éditeur de texte.

Pour afficher un message avec Javascript, on peut utiliser la fonction `alert`.

Question 04 Modifier le fichier `code.js` (vide) et écrire le code suivant :

```
alert("Hello World !");
```

▷ Attention à la syntaxe Javascript!

- Comme en Python, les chaînes de caractères sont écrites entre guillemets (ou apostrophes)
- Il faut ajouter le symbole `;` (point-virgule) à la fin de la ligne

Vous pouvez maintenant recharger la page et admirer le résultat : un message hautement intrusif s'affiche à l'écran. On appelle ce genre de message un *popup*. C'est assez gênant pour la navigation WEB car le reste de la page est bloqué tant qu'on n'appuie pas sur le bouton OK.

Une manière plus civilisée d'afficher un message avec Javascript est de placer ce message directement dans le code HTML, en modifiant ce dernier.

Question 05 Modifier le fichier `code.js` et écrire le code suivant, en supprimant le code précédent :

```
document.write("Hello World !");
```

En remplaçant `alert` par `document.write`, nous demandons à Javascript d'écrire directement dans la page WEB.

☞ *Actualiser la page WEB pour voir la modification.*

Le code HTML est modifié par Javascript, et le navigateur adapte l'affichage en conséquence. On peut voir le changement du code HTML en ouvrant le menu *Outils de développement* (raccourci : F12) et en sélectionnant l'onglet *Inspecteur*.



Notre texte est donc ajouté **à la fin du corps de notre page**. Il est cependant possible d'afficher ce texte dans un endroit bien précis de la page.

Question 06 Modifier le fichier `code.js` et écrire le code suivant, en supprimant le code précédent :

```
document.querySelector("#message").innerHTML = "Hello World !";
```

Si vous rechargez la page, vous constaterez que la phrase « *Voici un paragraphe.* » a disparu et a été remplacée par notre message « *Hello World!* ». Le code Javascript que nous avons entré permet de modifier le code HTML (*innerHTML*) de la balise identifiée par le mot *message*. Cette balise est une balise `<p>`, identifiée grâce à l'attribut `id` :

```
<p id="message">Voici un paragraphe.</p>
```

Grâce à cet identifiant, on peut cibler la balise à modifier, grâce à Javascript et *document.querySelector*.

Gestion d'événements avec Javascript

Une fonctionnalité très puissante de Javascript est la **gestion d'événements**. Un événement est une action réalisée par l'utilisateur dans le navigateur, comme un clic avec la souris, le passage de la souris sur une zone, un défilement de la page... Il est alors possible de **déclencher** un code Javascript lors d'un événement précis... comme l'appui sur un bouton.

Question 07 Modifier le fichier `page.html`, en ajoutant un attribut `onclick` à la balise `<button>` de la manière suivante :

```
<p><button onclick="alert('Quand finira ce TP ?');">Cliquez ici !</button></p>
```

Rechargez alors la page, et cliquez sur le bouton. Magie du Javascript, un message s'affiche.

Que s'est-il passé? Nous avons inscrit du Javascript directement dans la balise HTML avec l'attribut `onclick`. Cependant, comme son nom l'indique, le `alert` n'est exécuté que lorsque on clique sur le bouton avec la souris. Le message est affiché suite à un **événement** bien précis, un clic.

Il est néanmoins assez maladroit d'inscrire du code Javascript directement dans le fichier HTML. On préfère utiliser une **fonction** qui sera appelée lors du clic. Une fonction Javascript, c'est comme une fonction Python : c'est un sous-programme, une suite d'instructions exécutée lorsqu'on appelle cette fonction.

Pour définir une fonction en Javascript, on utilise la syntaxe suivante :

```
function nomDeLaFonction(arguments) {  
    instructions...  
}
```

Remarquez l'utilisation d'accolades, dont le rôle est de signaler le début et la fin de la fonction. L'indentation (décalage vers la droite) n'a pas lieu d'être en Javascript! On utilise un décalage pour que le code soit plus lisible, mais ce n'est pas obligatoire.

Question 08 Définir la fonction `afficherMessage()` dans le fichier `code.js` :

```
function afficherMessage() {  
    alert("Hello World !");  
}
```

Question 09 Modifier le fichier `page.html` afin d'appeler la fonction `afficherMessage()` lorsqu'on clique sur le bouton :

```
<p><button onclick="afficherMessage();">Cliquez ici !</button></p>
```

Rechargez la page HTML et admirez le résultat !

Question 10 Modifier la fonction `afficherMessage()` afin qu'elle affiche « Le Javascript c'est super! » dans la balise `<p>` identifiée par l'id `message`. Vérifiez votre code en cliquant sur le bouton (sans oublier de recharger la page!).

Question 11 Écrire une fonction `modifierTitre()` qui modifie le titre de la page WEB en « Site WEB hacké avec Javascript ». On pourra ajouter un identifiant à une balise bien précise du code HTML, et appeler cette fonction lors du clic sur un second bouton « Hacker le site WEB ».

☞ Dans cette dernière question, il faut modifier les fichiers `page.html` et `code.js`.

2.2.2 Heure en temps réel

Question 12 Ouvrir le dossier `2.Heure en temps réel` et les 3 fichiers associés.

Javascript permet de récupérer toutes les informations sur la date et l'heure actuelle utilisée par l'ordinateur. Pour faire cela, nous allons créer une **variable** (comme en Python) avec la syntaxe suivante :

```
var nom = valeur;
```

Contrairement à Python, on utilise le mot-clé `var` lorsqu'on **déclare** notre variable. On utilise également le symbole `;` à la fin de la ligne, comme souvent en Javascript (et dans bon nombre d'autres langages de programmation).

Question 13 Dans le fichier `code.js`, créer une variable `date` de la manière suivante, et afficher sa valeur avec `alert` :

```
var date = new Date();  
alert(date);
```

Les informations contenues dans la variable `date` sont nombreuses. On peut cependant récupérer uniquement celles qui nous intéressent, à savoir les heures et les minutes de la manière suivante (`date` étant la variable définie ci-dessus) :

```
var heures = date.getHours();  
var minutes = date.getMinutes();
```

Question 14 Écrire dans le fichier `code.js` les instructions précédentes.

Question 15 Créer une variable `heureLisible` comme ci-dessous :

```
var heureLisible = heures + "h" + minutes;
```

☞ Comme en Python, nous avons ici **concaténé** des chaînes de caractères, afin d'obtenir une chaîne comme `17h32`.

Dans le code HTML se trouve une balise `<p id="heure"></p>`. En reprenant ce qui a été fait auparavant, on peut créer une fonction dont le but est d'afficher l'heure dans cette balise. Voici le code complet à écrire dans `code.js` :

```
function afficherHeure() {  
    var date = new Date(); // On crée la variable contenant les informations sur la date actuelle  
  
    var heures = date.getHours(); // On récupère les heures  
    var minutes = date.getMinutes(); // Et les minutes  
    var dateLisible = heures + "h" + minutes; // On met en forme  
  
    // On écrit cette date dans la balise <p> identifiée par l'id 'heure'  
    document.querySelector("#heure").innerHTML = dateLisible;  
}
```

☞ Remarquez l'utilisation de commentaires en Javascript à l'aide des caractères `//`

Il ne reste plus qu'à appeler notre fonction. On peut le faire à la fin du fichier Javascript, avec la commande :

```
afficherHeure();
```

Question 16 Modifier le fichier `code.js` et recharger la page HTML. Admirez le résultat !

Question 17 Modifier le code précédent pour afficher également les secondes. On utilisera la commande `date.getSeconds()` .

Heure en temps réel avec Javascript

Quelle heure est-il ?

18:45:22

▷ Mais... L'heure ne se met pas à jour!

Nous n'avons jamais demandé ça ! Nous avons simplement écrit une fonction qui affiche l'heure. L'affichage est effectué une seule fois, lors de l'appel de la fonction. Magie du Javascript oblige, il est possible d'appeler une fonction de façon répétée, grâce à `setInterval` .

Question 18 Ajouter la commande suivante à la fin du fichier `code.js` :

```
setInterval(afficherHeure, 1000);
```

Grâce à cette commande, la fonction `afficherHeure()` (à écrire sans parenthèses dans le `setInterval`) est appelée toutes les 1000 milli-secondes (soit toutes les secondes). L'heure s'affiche enfin en temps réel.

2.2.3 Modifier l'apparence de la page

Nous avons vu que la couleur de fond d'un élément est modifiable par la propriété CSS `background`. En plus de permettre la modification du code HTML, Javascript permet également la modification des propriétés CSS de chaque élément.

Question 19 Ouvrir le dossier `3.Modifier le CSS` et les 3 fichiers associés.

Question 20 Quel est le code hexadécimal utilisé pour le fond de la page WEB? On pourra jeter un oeil au fichier `style.css` ...

Nous allons écrire une fonction Javascript dont le but sera de modifier cette valeur, afin de changer la couleur de fond de notre page en un clic. Quelle couleur utiliser? Celle qui sera donnée par l'utilisateur à l'aide d'un **champ texte** affiché grâce à la balise HTML `<input>`.

Question 21 Écrire la fonction suivante dans le fichier `code.js` :

```
function modifierFond() {  
    var couleur = document.querySelector("#couleur").value;  
    document.querySelector("html").style.background = couleur;  
}
```

Cette fonction permet de récupérer la couleur inscrite par l'utilisateur dans le champ texte prévu à cet effet, et de modifier le CSS de la balise `<html>` en conséquence.

Question 22 Pour que notre code soit opérationnel, reste à appeler notre fonction **au clic sur le bouton**.

Rajouter un attribut `onclick` au bouton, comme vu précédemment.

Vous êtes maintenant capable de modifier la couleur de fond de la page! Vous pouvez utiliser des couleurs HTML (en anglais, comme *red*, *blue*, *pink*...) ou hexadécimales (comme *#ddd* ou *#a23f64*).

Question 23 Ajouter un champ texte et un bouton afin de modifier la couleur du texte dans la page.

La propriété CSS à modifier est `color`.

Le sapin de Noël (niveau avancé)

Question 24 Écrire une fonction Javascript `modifierFondAleatoire()` qui modifie la couleur de fond de la page de manière aléatoire.

Pour faire cela, on pourra utiliser la notation RGB d'une couleur : au lieu d'écrire une valeur en anglais (*red*, *green*...) ou une valeur hexadécimale (*#ddd*) pour la couleur, on écrit *rgb(R, V, B)* où R, V et B sont des nombres compris entre 0 et 255, correspondant à la quantité de chaque couleur primaire. Par exemple, le rouge s'écrit *rgb(255,0,0)*.

Pour générer un nombre aléatoire compris entre 0 et 255 en Javascript, on peut écrire :

```
var nombre = Math.floor(Math.random() * Math.floor(256));
```

Question 25 Appeler cette fonction toutes les 200ms avec `setInterval` pour rendre votre page psychédélique et fêter Noël avant l'heure.

2.3 Envie d'en savoir (beaucoup) plus ?

Pour en savoir plus sur le langage Javascript, une seule adresse :

<https://openclassrooms.com/fr/courses/1916641-dynamisez-vos-sites-web-avec-javascript>