MÉMO1 - DÉVELOPPEMENT JOUR 1

Cours Javascript - Frédéric LOSSIGNOL

Ce qu'il faut retenir

Introduction

Qu'est-ce que Javascript

Si l'HTML est un langage de *description,* **Javascript lui est un langage de programmation**.

Il est la troisième brique essentielle aux pages web côté client. HTML/CSS et JS sont aujourd'hui majoritairement utilisés sur les pages web. S'il est un langage connu pour être côté client (s'exécutant sur le navigateur), il est aussi de plus en plus utilisé côté serveur.

Javascript vous permettra à la fois de construire des applications mais aussi d'ajouter une couche d'**interactivité** et de **dynamisme** à vos pages web.

SOMMAIRE

1 Comment charger Javascript dans votre page web

Il y a 2 façons d'injecter du Javascript dans une page HTML :

Directement entre 2 balises <script>

```
<!DOCTYPE html>
<html lang="fr">
<head>
	<meta charset="utf-8">
	<title>JavaScript - Jour 1</title>
</head>
<body>
	<h1>Hello World !</h1>
<script>
	alert('Et si on apprenait le JS ?');
</script>
</body>
</html>
```

Ou en chargeant un fichier javascript externe via une requête HTTP

```
<!DOCTYPE html>
<html lang="fr">
<head>
	<meta charset="utf-8">
	<title>JavaScript - Jour 1</title>
</head>
<body>
	<h1>Hello JS !</h1>
	<!-- Ajout d'un attribut HTML src pour charger un fichier JS externe-->
	<script src="fichier-javascript.js"></script>
</body>
</html>
```

A retenir :

La méthode d'insertion d'un fichier Javascript externe permet de ne pas mélanger notre code JS avec le HTML, et de conserver une structure compartimentée entre vos fichiers HTML/CSS/JS.

On peut insérer le fichier Javascript entre les balises <head> et </head> de la page, mais on l'insère le plus souvent avant la balise de fermeture </body> (à la fin du document HTML) pour des raisons de performances.

2 Les variables et les Constantes

Une variable est un élément nommé qui prend une valeur de type chaîne de caractère (string) ou nombre (Integer, Float, …) ou booléen (vrai ou faux). La variable est réutilisable et sa valeur peut être modifiée lors de l'exécution de notre code Javascript.

2.1 Pourquoi utiliser des variables ?

Pour stocker des données réutilisables dans votre code.

Exemple 1 : pour une petite application qui calculera et affichera le montant TTC, nous allons avoir besoin des variables montantHT, tauxTVA, montantTVA et montantTTC.

Exemple 2 : si l'utilisateur envoie un message via un formulaire de contact, nous aurons besoin des variables nom, email et message afin de faire passer leur valeur respective dans l'email à envoyer.

2.2 Comment utiliser les variables ?

Pour utiliser les variables dans votre code, il suffit de les déclarer comme existantes en les nommant vous-même après le mot clé **var**

```
var monPrenom ; // On déclare la variable monPrenom
```

Votre variable existe et est alors accessible dans votre code javascript en la nommant de nouveau et vous pouvez lui assigner une valeur. (ici la chaîne de caractère ***Frédéric)***.

```
monPrenom = 'Frédéric' ;
// on dit ici qu'on assigne la valeur 'Frederic' à la variable monPrenom
```

Conseil bonne pratique : nommer vos variables de façon descriptive, aide à la compréhension de votre code. Vous vous y retrouverez bien plus facilement quand vous reviendrez dans votre propre code car elles auront un sens pour les humains que vous êtes.

2.3 Les règles de nommage et les conventions de nommage des variables

2.3.1 Les règles

un nom de variable **doit** commencer par une lettre ou un underscore (_). Les autres caractères ne peuvent pas contenir de signes représentants un opérateur mathématique (-,+,/,*, …). Exemple :

```
var monPrenom;    // est correct
var 1Prenom;      // est un mauvais nommage (pas de chiffre dans le premier caractère)
var mon-prenom;   //  est un mauvais nommage (à cause du signe - )
var var;          //  est un mauvais nommage (car var un nom réservé)
```

Info : Attention aussi, le Javascript est sensible à la casse. Par exemple :

```
var monPrenom;
var MONPRENOM;
```

sont bien 2 variables différentes. Les erreurs dues à une mauvaise écriture des variables sont courantes, surtout lorsque l'on débute. Les *conventions de nommage* permettent d'éviter ce type d'erreur dans notre code.

2.3.2 Le cas des noms de variables en plusieurs mots

Lorsque vous avez besoin de nommer des variables à plusieurs mots, la règle simple pour une meilleure compréhension : Le nom de votre variable devient de plus en plus précis de droite à gauche, dans le sens de la lecture.

```
// Exemple ici en nommant plusieurs variables qui sont les caractéristiques d'un utilisateur.
// Déclaration des variables
var userFirstname;
var userLastname;
var userEmail;

// Affectation de valeurs à chacune des variables
userFirstname = 'Mikael';
userLastname = 'Jordan';
userEmail = 'm.jordan@nba.com';
```

Les règles de nommage et les conventions de nommage des variables (suite et fin)

2.3.3 Under_score ou camelCase ?

http://bit.ly/1BR51kh (~ -51.5% d'erreur de code avec le camelCase par exemple)

Si vous devez nommer un nom de variable qui contient plusieurs mots, vous rencontrerez généralement 2 façons de le faire :

```
var mon_prenom; // under_score
var monPrenom; //  camelCase
```

Les 2 sont utilisées mais le camelCase est le plus courant.

En résumé :

1. utiliser des noms de variables descriptifs et compréhensibles
2. Les mots multiples doivent être plus précis de gauche à droite
3. Utiliser le camelCase est une convention courante et recommandée

2.3.4 Les constantes

A l'inverse d'une variable et comme son nom l'indique, une constante est une donnée qui aura une valeur fixe. On la déclare généralement au début du programme **en utilisant la convention de nommage en lettres majuscules et underscore.**

```
const TAUX_TVA = 20;
// on utilise ici le mot-clé const pour déclarer notre constante
// elle est alors réutilisable dans tout notre code.

//  exemple ici sur le calcul de la TVA
// on déclare d'abord les variables dont on va avoir besoin
var montantHT;
var montantTVA;

montantHT = 100;

// nous pouvons utiliser ici notre constante pour calculer le taux montant de la TVA
montantTVA = montantHT * TAUX_TVA / 100;
document.write(montantTVA); // affichera 20
```

3 Les tableaux (array)

Un tableau est une variable capable de stocker plusieurs données.
Vous pouvez vous représenter un tableau sous cette forme.

Index	0	1	2	3
Valeur	Christian	Aurélien	Kevin	Zakaria

Ici la variable ***prenoms*** contient plusieurs valeurs associées chacune à un index.

Comment utiliser les tableaux ?

Nous avons 2 façons de déclarer un tableau :

- la syntaxe longue (nous la trouvons de moins en moins)
- la syntaxe courte (plus lisible)

La syntaxe longue

```
var prenoms = new Array();
// la variable prenoms est maintenant déclarée comme un tableau

prenoms[0] = 'Christian';
prenoms[1] = 'Aurélien';
prenoms[2] = 'Kevin';
prenoms[3] = 'Zakaria';

document.write(prenoms[1]); // affichera Aurélien
```

La syntaxe courte

```
var prenoms = [];
// en ajoutant les crochets, on indique que la variable prenoms est un tableau

prenoms = ['christian', 'Aurélien', 'Kevin', 'Zakaria'];
// On assigne les valeurs à l'intérieur du tableau
// Notez que cette syntaxe ne nécessite pas d'index,

// automatiquement la première valeur est assigné à l'index 0
document.write(prenoms[3]); // affichera Zakaria
```

4 La concaténation

La concaténation consiste à assembler plusieurs chaînes de caractères en une seule. L'opérateur utiliser est le + (à pas confondre avec l'opérateur addition qui s'applique uniquement aux données de type nombre (Integer, Float).

```
/*
   Affichage de deux paragraphes directement dans la page HTML.
   Les chaînes de caractères sont concaténées avec le +
 */

var prenom = 'Mikael Jordan';
document.write("<p>Hello, je m'appelle " + prenom + ', et toi ? </p>') ;
// affichera dans un paragraphe <p>Hello, je m'appelle Mikael Jordan, et toi ?</p>

/* Remarquez que l'on utilise des guillemets doubles ou simples pour englober une chaîne de    caractère alors qu'une variable est juste appelée par son nom, l'opérateur + sert ici à concaténer, c'est à dire à assembler 2 chaînes.
 */
```

Info : Attention à la particularité du guillemet simple ou double à l'intérieur d'une chaîne de caractère. Lorsque qu'un guillemet simple ou double se trouve dans une chaîne de caractère alors que celle-ci est entourée par un guillemet de même type, nous utilisons ce qu'on appelle un caractère d'échappement.

```
var comment = 'J\'ai retrouvé mon chien qui s\'était égaré.' ;
var reply = "Heureusement que tu as retrouvé \"Doogy\" c'est super !";

/*
Le caractère d'échappement \ placé juste avant le guillemet permet de dire que le caractère suivant n'est pas le caractère de fermeture de la chaîne. On dit qu'on échappe un caractère.
 */
```

***Doogy** va dormir au chaud et vous connaissez tout sur la concaténation en JS.*

5 Les types de variables et les transformations

Les variables en Javascript peuvent être de type :

- Chaîne de caractères (String)
- Nombre (Number)

Nous verrons plus tard qu'il existe d'autres types de variables notamment les Booléens qui prennent comment valeur Vrai ou Faux.

Les chaînes de caractères sont des caractères alphanumériques. Lorsque l'on assigne une valeur de type chaîne de caractère à une variable, on la place entre guillemets.

Les variables de type nombre sont des caractères numériques. Lorsque l'on assigne une valeur de type nombre à une variable, on ne mets pas de guillemets.

```
var maChaine = 'Ceci est une chaîne de caractère' ;

var nombre = 20; // ceci est une variable de type nombre
var nombre = '20';  // ceci est une variable de type chaîne de caractère
```

> Transformation d'une variable de type **string** en une variable de type **number**
Dans certains cas, il peut être utile de transformer un type de variable en un autre type de variable. Par exemple, pour exécuter une opération mathématique courante (ici une addition), Javascript a besoin d'utiliser des variables de type nombre.

Pour transformer une chaîne de caractère en nombre, on peut noter les fonctions
parseInt() utile pour transformer une chaîne de caractère en nombre entier.
parseFloat() utile pour transformer une chaîne de caractère en nombre décimal.

http://devdocs.io/javascript/global_objects/parseint

Ouvrez la console dans votre navigateur (Raccourci F12) et rechargez votre page.

```
console.log(3 + 2); // affichera 5 (type number)

console.log("3 + 2"); // affichera la chaîne de caractère  3+2 (type string)

console.log("3" + "2"); // affichera la chaîne de caractère 32 (type string)

console.log (parseInt("3")+ parseInt("2")); // affichera 5 (type number)

// Ici on a ici utilisé la fonction parseInt() qui transforme une chaîne de caractère en nombre
et // javascript peut effectuer une addition.
```

6 L'objet Date()

http://devdocs.io/javascript/global_objects/date

Javascript fournit l'Objet ***Date()*** afin de créer et de manipuler les dates.
Pour savoir ce que retourne l'objet date, il suffit de l'instancier (le créer) avec le mot-clé **new** et de stocker son résultat dans une variable.

```
var laDate = new Date() ;
console.log(laDate);
// affichera la date et l'heure du jour

// on peut aussi passer en paramètre une date définie
var dateAnniversaire = new Date("1976-11-02T08:30:00") ;

// les méthodes Getter permettent d'extraire des données de l'objet date
var jour = dateAnniversaire.getday();
console.log(jour);
// getDay() appliqué à l'objet Date Renvoie le jour de la semaine (0-6)
// getMonth()  Renvoie le mois de l'année (0-11)
// getFullYear() Renvoie l'année (avec 4 chiffres)
```

A partir de là nous pouvons afficher une date en Français si on le souhaite.

La méthode getDay() ***Renvoie le jour de la semaine (0-6)***
La méthode getMonth() ***Renvoie le mois de l'année (0-11)***

il nous suffit alors de déclarer un tableau avec les termes en Français indexés sur ces valeurs. (voir la suite)

Exercice : afficher une date d'anniversaire en Français

```
// Objectif : afficher une date en Français
var dateAnniversaire = new Date("1976-11-02T08:30:00") ;


// 1 - je déclare une variable jourEnFrançais de type tableau
//  (pour l'exemple avec la syntaxe courte)
var joursenFrancais = [
'Dimanche', 'Lundi', 'Mardi', 'Mercredi', 'Jeudi', 'Vendredi', 'Samedi'
];

// 2 je déclare une variable moisEnFrancais de type tableau
// (pour l'exemple la syntaxe longue)
var moisEnFrancais = new Array();
moisEnFrancais[0] = 'Janvier';
moisEnFrancais[1] = 'Février';
moisEnFrancais[2] = 'Mars';
moisEnFrancais[3] = 'Avril';
moisEnFrancais[4] = 'Mai';
moisEnFrancais[5] = 'Juin';
moisEnFrancais[6] = 'Juillet';
moisEnFrancais[7] = 'Août';
moisEnFrancais[8] = 'Septembre';
moisEnFrancais[9] = 'Octobre';
moisEnFrancais[10] = 'Novembre';
moisEnFrancais[11] = 'Décembre';


document.write('Je suis né le ' + joursEnFrancais[dateAnniversaire.getDay()] + ' ');
document.write(dateAnniversaire.getDate() + ' ');
document.write(moisEnFrancais[dateAnniversaire.getMonth()] + ' ');
document.write(dateAnniversaire.getFullYear() + ' ');
document.write('à ' + dateAnniversaire.getHours() + ':');
document.write('à ' + dateAnniversaire.getMinutes() + '');

document.write('<h2>Whao il est super jeune le prof de Dev !</h2>');
```

Ce code est testable.

7 Les outils du développeur / liens utiles

Les outils du développeur :

- Un éditeur de texte ou un IDE (Sublime Text, Netbeans, PHPStorm,...)
- L'outil développement sur les navigateurs (Firebug sur FireFox et l'outil de développement sur Chrome, tous deux accessibles avec le raccourci F12)
- DevDocs (www.devdocs.io)

Liens utiles :

Cours sur OpenClassRoom

- https://openclassrooms.com/courses/dynamisez-vos-sites-web-avec-javascript/

Application mobile pour apprendre les bases avec des petits exercices progressifs (sympa dans le métro :))

- (AppStore et PlayStore et windowsStore).
- => https://play.google.com/store/apps/details?id=com.sololearn.javascript&hl=fr
- => https://itunes.apple.com/us/app/learn-javascript/id952738987?mt=8
- => https://www.microsoft.com/fr-fr/store/apps/learn-javascript/9wzdncrdj6b4

Les prochaines fois nous verrons les Conditions et nous finiront par coder un petit jeu en utilisant ce que vous savez déjà faire.

Conseil : révisez les notions apprises en relisant ce document :

- variables
- tableaux
- concaténation
- afficher le contenu d'une variable avec document.write(), alert(), ou console.log();

En relisant le code des explication et des exercices corrigés.

et PRATIQUEZ , c'est la meilleure voie.
Si vous n'êtes pas inspiré(e), pratiquez avec des exemples trouvés sur le web, sur le lien d'OpenclassRoom par exemple.

Si vous avez des questions ou des remarques, je suis joignable sur par Email : frederic.lossignol@gmail.com

A bientôt pour les jours 2 et 3

Frédéric Lossignol