|  |  |  |
| --- | --- | --- |
|  | **UNIVERSITE IBN ZOHR**  **Ecole Supérieure de Technologie – Laâyoune**  EST, Quartier 25 Mars BP 3007, Laâyoune – Maroc  **Site Web:**[http://w2.estl.ac.ma](http://w2.estl.ac.ma/) |  |

**Rapport De Projet De Fin D’etudes**

**D**iplôme **U**niversitaire de **T**echnologie (DUT)

**Filière : Génie Informatique**

**Sous le théme :**

**« JavaScript »**

**Réalisé par :**

* **Mr Redouane AMGHNOUSS :** Etudiant en 2ième année de la filière Génie Informatique
* **Mr Azeddine GHARRADY :** Etudiant en 2ième année de la filière Génie Informatique

**Encadré par :**

* **Mr Youssef aribou :** Enseignant/Chercheur à l’EST de Laâyoune

**Jurys :**

**-Mr**

**-Mr**

**-Mr**

Année universitaire : 2020-2021

**Dédicace**

A nos parents :

Ce travail est avant tout le fruit des sacrificier qu’ils ont consentis afin de parfaire notre éducation et notre formation. Leurs prières et leur bénédiction ont été les cierges qui ont éclairé nos parcours. Nous ne saurons exprimer mon extrême reconnaissance et gratitude.

A toutes nos grandes familles

GHARRADY , AMGHNOUSS

A tous nos amis

En honneur à la grande amitié qui nous unit, des souvenirs et tous les moments passés ensembles.

A notre prestigieuse école

L’Ecole Supérieure de Technologie de Laayoune ESTL

Nous espérons que cet humble travail soit à la hauteur de vos attentes.

**Remerciement**

Ce rapport est le résultat d'un effort constant. Cet effort n'aurait pu aboutir sans la contribution de nombre de personnes. Ainsi se présente l'occasion de les remercier.

Je tiens à remercier dans un premier temps, toute l’équipe pédagogique de l’école Supérieure de technologie à Laayoune, de l’effort précieux qu’ils fournissent afin de nous offrir une formation complète et Bonne.

Nos plus profonds sentiments vont à **Mr. ARIBOU YOUSSEF**, notre encadrant à l’ESTL pour avoir accepté de diriger notre projet. Il nous a beaucoup soutenus et réconfortés malgré ses multiples engagements.

Nos remerciements aux membres du jury de nous avoir honoré en acceptant l’évaluation de ce présent projet.

Sans oublier de remercier tous nos parents, nos frères et toute personne qui nous ont aidé de près ou de loin pour bien réussir notre travail.

Enfin, que tous ceux qui ont participé à l’aboutissement de ce travail, trouvent ici, l’expression de nos sincères remerciements.

**Sommaire**

[**dédicace** 2](#_Toc68383844)

[**Remerciement** 3](#_Toc68383844)

[**Liste des abréviations** 6](#_Toc68383844)

[**Liste des figures**  7](#_Toc68383844)

[**Liste des tableaux** 7](#_Toc68383844)

[**1** **Chapitre  : Introduction de JavaScript** 10](#_Toc68383844)

[1.1 Introduction : 10](#_Toc68383845)

[1.2 Présentation de JavaScript : 10](#_Toc68383846)

[1.2.1 Définition de JavaScript : 10](#_Toc68383847)

[1.2.2 Histoire de JavaScript : 11](#_Toc68383848)

[1.2.3 Les versions de JavaScript : 12](#_Toc68383849)

[1.3 Les outils du langage : 13](#_Toc68383850)

[1.3.1 la différence entre java et javaScript : 13](#_Toc68383851)

[1.3.2 Présentation des outils*:* 14](#_Toc68383852)

[1.3.2.1 Les éditeurs : 14](#_Toc68383853)

[1.3.2.2 Les Navigateurs : 14](#_Toc68383854)

[1.3.3 Popularité des langages de programmation : 15](#_Toc68383855)

[1.4 Conclusion : 15](#_Toc68383856)

[**2** **Chapitre : Les bases de JavaScript** 16](#_Toc68383857)

[2.1 Introduction : 16](#_Toc68383858)

[2.2 Emplacement du code JavaScript : 16](#_Toc68383859)

[2.3 JavaScript Output : 18](#_Toc68383860)

[2.3.1 Alert() 18](#_Toc68383861)

[2.3.2 Prompt() 18](#_Toc68383862)

[2.3.3 Ducomment.write() 18](#_Toc68383863)

[2.3.4 console.log() 19](#_Toc68383864)

[2.3.5 confirm() 19](#_Toc68383865)

[2.4 Les commentaires : 20](#_Toc68383866)

[2.5 Les variables : 20](#_Toc68383867)

[2.5.1 Déclaration: 20](#_Toc68383868)

[2.5.2 Nom de variable : 20](#_Toc68383869)

[2.5.3 Affectation : 20](#_Toc68383870)

[2.5.4 Utilisation d’une variable : 20](#_Toc68383871)

[2.6 Type de Données : 21](#_Toc68383872)

[2.6.1 Type de Donneés – Number 21](#_Toc68383873)

[2.6.2 Type de Donneés – String 21](#_Toc68383874)

[2.6.3 Type de Donneés – Boolean 22](#_Toc68383875)

[2.6.4 Type de Donneés – Object 23](#_Toc68383876)

[2.6.5 Typeof 23](#_Toc68383877)

[2.7 Les opérateurs : 24](#_Toc68383878)

[2.8 Conclusion : 24](#_Toc68383879)

[**3** **Chapitre : Les structures et les fonctions** 25](#_Toc68383880)

[3.1 Introduction : 25](#_Toc68383881)

[3.2 Les instructions de contrôle : 25](#_Toc68383882)

[3.2.1 Instruction if, if … else : 25](#_Toc68383883)

[3.2.2 Les conditions ternaires : 25](#_Toc68383884)

[3.2.3 switch : 25](#_Toc68383885)

[3.3 Les boucles : 26](#_Toc68383886)

[3.3.1 La boucle while : 26](#_Toc68383887)

[3.3.2 La boucle do… while : 26](#_Toc68383888)

[3.3.3 La boucle for : 27](#_Toc68383889)

[3.4 Les fonctions 27](#_Toc68383890)

[3.4.1 Types de fonctions : 27](#_Toc68383891)

[3.4.2 Valeurs de Retour 28](#_Toc68383892)

[3.5 Les tableaux : 28](#_Toc68383895)

[3.5.1 Déclaration d’un tableau : 29](#_Toc68383896)

[3.5.2 Les fonctions de tableau : 29](#_Toc68383897)

[3.6 Conclusion 30](#_Toc68383898)

[**4** **Chapitre : Dom et les événements** 31](#_Toc68383899)

[4.1 Introduction : 31](#_Toc68383900)

[4.2 Le DOM : 31](#_Toc68383901)

[4.2.1 Definition : 31](#_Toc68383902)

[4.2.2 Structure d’un DOM : 31](#_Toc68383903)

[4.2.3 APIs Spécifiques : 32](#_Toc68383904)

[4.3 Les événements : 34](#_Toc68383905)

[4.4 Les Frameworks de JavaScript : 35](#_Toc68383906)

[4.4.1 Définition : 35](#_Toc68383907)

[4.4.2 Les trois principaux Frameworks : 36](#_Toc68383908)

[4.4.2.1 Angular 36](#_Toc68383909)

[4.4.2.2 React 36](#_Toc68383910)

[4.4.2.3 Vue.js 36](#_Toc68383911)

[4.5 Conclusion 37](#_Toc68383912)

**Liste des abréviations**

**JS**  : JavaScript.

**HTML**  : HyperText Markup Languuage.

**CSS**  : Cascading Style Sheets.

**ECMA** : European Computer Manufacturers Association.

**OOP** : Object Oriented Programming.

**DOM**   : Document Object Model.

**XML** : eXtensible Markup Language.

**API**  : Interface Applicative de Programmation.

**Liste des figures**

[Figure 1  : Logo JavaSvript 10](file:///D:\G.I\Semestre%204\PFe\Projet%20Fin%20d'Etudes.docx#_Toc68385128)

[Figure 2 : côte client et serveur 11](file:///D:\G.I\Semestre%204\PFe\Projet%20Fin%20d'Etudes.docx#_Toc68385129)

[Figure 3: Entreprise Netscape 11](file:///D:\G.I\Semestre%204\PFe\Projet%20Fin%20d'Etudes.docx#_Toc68385130)

[Figure 4: Brendan Eich 12](file:///D:\G.I\Semestre%204\PFe\Projet%20Fin%20d'Etudes.docx#_Toc68385131)

[Figure 5: java vs JavaScript 14](file:///D:\G.I\Semestre%204\PFe\Projet%20Fin%20d'Etudes.docx#_Toc68385132)

[Figure 6 : Les Navigateurs 14](#_Toc68385133)

[Figure 7: Popularité des langages de programmation 15](file:///D:\G.I\Semestre%204\PFe\Projet%20Fin%20d'Etudes.docx#_Toc68385134)

[Figure 9 : L’insertion de code dans head et body 17](#_Toc68385135)

[Figure 10 : L’insetion de code dans un fichier externe 17](#_Toc68385136)

[Figure 11 : Alert() 18](#_Toc68385137)

[Figure 12 : Prompt() 18](file:///D:\G.I\Semestre%204\PFe\Projet%20Fin%20d'Etudes.docx#_Toc68385138)

[Figure 13 : Document.write() 18](file:///D:\G.I\Semestre%204\PFe\Projet%20Fin%20d'Etudes.docx#_Toc68385139)

[Figure 14 : Console.log() 19](#_Toc68385140)

[Figure 15 : Confirm() 19](file:///D:\G.I\Semestre%204\PFe\Projet%20Fin%20d'Etudes.docx#_Toc68385141)

[Figure 16 : Utilisation d’une variable avec Date() 21](#_Toc68385142)

[Figure 17 : Type number 21](file:///D:\G.I\Semestre%204\PFe\Projet%20Fin%20d'Etudes.docx#_Toc68385143)

[Figure 18 :Type string avec la concaténation des variables 22](#_Toc68385144)

[Figure 19 :Type Boolean 22](#_Toc68385145)

[Figure 20 : La fonction Typeof() 23](#_Toc68385146)

[Figure 21: Les conditions if...else 25](#_Toc68385147)

[Figure 22: Switch 26](#_Toc68385148)

[Figure 23 : La boucle while 26](#_Toc68385149)

[Figure 24 : La boucle for 27](#_Toc68385150)

[Figure 25 : la fonction addition 27](#_Toc68385151)

[Figure 26 : la fonction retourne une valeur 28](file:///D:\G.I\Semestre%204\PFe\Projet%20Fin%20d'Etudes.docx#_Toc68385152)

[Figure 27 : Exemple de DOM 31](#_Toc68385153)

[Figure 28 : Structure d'un DOM 32](#_Toc68385154)

[Figure 29: La méthode getElementById() 33](#_Toc68385155)

[Figure 30 : La méthode getElementByClassName() 33](#_Toc68385156)

[Figure 31 : La méthode querySelector et querySelectorAll() 34](#_Toc68385157)

[Figure 32:Les framework 35](file:///D:\G.I\Semestre%204\PFe\Projet%20Fin%20d'Etudes.docx#_Toc68385158)

[Figure 33 : Angular 36](file:///D:\G.I\Semestre%204\PFe\Projet%20Fin%20d'Etudes.docx#_Toc68385159)

[Figure 34 : React 36](file:///D:\G.I\Semestre%204\PFe\Projet%20Fin%20d'Etudes.docx#_Toc68385160)

[Figure 35 : Vue 36](file:///D:\G.I\Semestre%204\PFe\Projet%20Fin%20d'Etudes.docx#_Toc68385161)

**Liste des tableaux**

[Table 1 : Les Versions 13](#_Toc68385252)

[Table 2 :Les opérateurs 24](#_Toc68385253)

[Table 3: Exemple de tableau 29](#_Toc68385254)

[Table 4 : Les fonctions de tableau 30](#_Toc68385255)

[Table 5 : Les événements 35](#_Toc68385256)

**Résume :**

**Abstract :**

**ملخص :**

**Introduction Générale**

Dans le cadre du formation DUT génie informatique à l'école supérieure de Technologie de Laayoune en vue de l'obtention du diplôme en fin d'année j'ai été chargées de réaliser un projet complet en situation professionnelle qui marque la fin des études pour but développer l'autonomie et la responsabilité des étudiants, et mettre en pratique les enseignements reçus et permettre ainsi aux étudiants d'affirmer leurs savoir-faire et à considérer leurs compétences.

De nos jours, l'informatique est considérée comme un outil indispensable à toute entreprise qui ne veut pas rester en marge de la mondialisation. Mais nul ne peut nier qu'il y a encore une absence des système informatiques dans le parcours des plupart des étudiants, même des filières de l'informatique.

Lorsque vous consultez une agence ou entreprise dans le cadre du développement d’un site web ou d’une application, il y a fort à parier que vous entendrez parler du JavaScript au cours du projet.Cette langage de programmation utilisé notamment lors de la conception de sites web et d’applications. Il est particulièrement utile pour concevoir des sites dynamiques.

Aujourd’hui, le langage de programmation JavaScript est massivement utilisé sur les sites web grand public. Les navigateurs web sont quant à eux devenus beaucoup plus efficaces pour le traduire. Donc tout les sites web modernes utilisent Javascript pour optimiser au maximum l’expérience de l’utilisateur.

Enfin, le projet de fin d'études est une étape importante de notre carrière professionnelle, afin de nous préparer au stage qui n'aura lieu que quelques mois après.

# **Chapitre  : Introduction de JavaScript**

## **Introduction :**

Dans le présent chapitre, nous entamons la présentation de notre projet. Tout d’abord, nous enchaînons avec la définition de JavaScript. Nous présentons l’hstoire et les versions de cette langage. Ensuite la différence entre Java et JavaScript. Nous terminons par la Présentation des outils.

## **Présentation de JavaScript :**

### **Définition de JavaScript :**

**J**avaScript, souvent abrégé en JS, est un langage de script très utilisé, surtout au niveau des pages/applications web. Cette page propose une description des différents aspects de JavaScript d’un point de vue plutôt théorique et conceptuel, sans entre dans les détails du codage. Pour une approche plus concrète, basée directement sur les caractéristiques du langage, Le JavaScript est aujourd’hui l’un des langages de programmation les plus populaires et il fait partie des langages web dits « standards » avec le HTML et le CSS. Son évolution est gérée par le groupe ECMA International qui se charge de publier les standards de ce langage.

Figure 1  : Logo JavaSvript

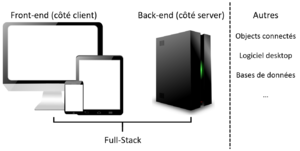
On dit que le HTML, le CSS et le JavaScript sont des standards du web car les principaux navigateurs du web (Google Chrome, Safari, Firefox, etc.) savent tous « lire » (ou « comprendre » ou « interpréter ») ces langages et les interprètent généralement de la même façon ce qui signifie qu’un même code va généralement produire le même résultat dans chaque navigateur.

Pour définir ce qu’est le JavaScript et le situer par rapport aux autres langages, et donc pour comprendre les intérêts et usages du JavaScript il faut savoir que :

* **Le JavaScript est un langage dynamique :**

Le JavaScript est un langage dynamique, c’est-à-dire un langage qui va nous permettre de générer du contenu dynamique pour nos pages web.

* **Le Javascript est un langage (principalement)cote client :**

Un langage « côté client » ou « client side » est un langage qui va être exécuté dans le navigateur des utilisateurs qui demandent la page. On peut également appeler ces langages des langages

« web » puisqu’ils sont principalement utilisés dans un contexte web.

Il existe aujourd’hui 3 langages côté client incontournables qui sont le HTML le CSS et le JavaScript.

Figure 2 : côte client et serveur

Les langages côté serveur sont des langages qui vont s’exécuter sur le serveur. Les navigateurs ne sont dans la grande majorité des cas pas capables de comprendre les langages serveur.

* **Le JavaScript est un langage interprété :**

Le JavaScript est un langage interprété. Cela signifie qu’il va pouvoir être exécuté directement sous réserve qu’on possède le logiciel interpréteur. Pas de panique ici : tous les navigateurs connus possèdent leur interpréteur JavaScript.

* **Le JavaScript est un langage orienté objet :**

Cela signifie simplement que le JavaScript va utiliser des objets dans son fonctionnement, et que ces objets sont prototypés.

### **Histoire de JavaScript :**

JavaScript a été créé en dix jours en 1995 par [Brendan Eich](https://fr.wikipedia.org/wiki/Brendan_Eich) qui travaillait à l’époque pour Netscape Le langage (également connu au début avec les noms de Mocha ou LiveScript) a ensuite intégré les standards [ECMA](https://www.ecma-international.org/) pour favoriser la diffusion et l'utilisation. Formellement, donc, le nom du langage est [ECMAScript](https://www.ecma-international.org/memento/TC39.htm) et ses spécifications sont établies par le groupe TC39. Ce groupe s'occupe de définir les propriétés d'un langage de Scripting, et JavaScript est un langage qui implémente ces propriétés.

Figure 3: Entreprise Netscape

Dans la plupart des cas, cependant, on ne fait pas cette distinction technique et on utilise tout simplement le terme JavaScript de manière omnicompréhensive. Ce nom crée parfois de la confusion avec un autre langage de programmation très connu : [Java](http://edutechwiki.unige.ch/fr/Java). Toutefois, malgré le fait que les deux langages partagent certaines similitudes, JavaScript et Java sont deux langages différents !

On peut séparer l’histoire de JavaScript en deux périodes :

* Avant le [Web 2.0](http://edutechwiki.unige.ch/fr/Web_2.0) : langage peu et mal utilisé, considéré comme un pseudo-langage à éviter pour les développeurs « sérieux ».
* Après le [Web 2.0](http://edutechwiki.unige.ch/fr/Web_2.0) : boom du langage qui est utilisé pour ajouter des fonctionnalités interactives aux pages web, grâce notamment à son intégration avec [HTML5](http://edutechwiki.unige.ch/fr/HTML5).

Depuis 2010-2015, il y a également une tendance pour utiliser JavaScript en dehors des pages web, notamment dans le développement côté-serveur ou dans le développement d’application desktop et mobile. Cette tendance est bien synthétisée par la [loi de Atwood](https://blog.codinghorror.com/the-principle-of-least-power/): **« any application that be can written in JavaScrpt, will eventually be written in JavaScript ».**

******

***"There was some pressure from management to make the syntax look like  
Java. There was also some pressure to make it not too big, because after all,  
people should use Java if they're doing any real programming.***

***This is just Java's dumb little brother. But if I put classes in, I’d be in big  
trouble.“***

***Brendan Eich***

Figure 4: Brendan Eich

### **Les versions de JavaScript :**

L'histoire compliquée de JavaScript se reflète également dans les noms de ses versions. En effet, comme tout langage de programmation, JavaScript a évolué dans le temps et des changements ont été apportés aux propriétés de l'implémentation ECMAScript. Voici un tableau récapitulatif des changements jusqu'en 2021 :

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom officiel** | **Abréviation** | **Nom non officiel** | **Date** |
| ECMAScript 1 | ES1 | - | 1997 |
| ECMAScript 2 | ES2 | - | 1998 |
| ECMAScript 3 | ES3 | - | 1999 |
| ECMAScript 4 | ES4 | - | Version abandonnée et jamais implémentée |
| ECMAScript 5 | ES5 | - | 2009 |
| ECMAScript 6 | ES6 | ES2015 | 2015 |
| ECMAScript 7 | ES7 | ES2016 | 2016 |
| ECMAScript 8 | ES8 | ES2017 | 2017 |
| ECMAScript 9 | ES9 | ES2018 | 2018 |
| ECMAScript 10 | ES10 | ES2019 | 2019 |
| ECMAScript 11 | ES11 | ES2020 | 2020 |
| ECMAScript 12 | ES12 | ES2021 | 2021 |

Table 1 : Les Versions

À partir de 2021, il y a le souhait de relâcher des nouvelles versions de manière plus fréquente, par exemple une fois par année. Il faut néanmoins tenir compte qu'une nouvelle version du langage nécessite également d'une version de l'interprète qui puisse comprendre les nouveautés du langage. Ceci est l'une des sources d'incompatibilité les plus répandues, surtout dans les pages web qui nécessitent d'une version du navigateur récent pour pouvoir être interprétée correctement.

## **Les outils du langage :**

### **la différence entre java et javaScript :**

Encore aujourd’hui, certaines personnes ont tendance à confondre les deux langages « Java » et « JavaScript ».

JavaScript et Java ne sont en aucune manière aparentés.il s’agit de deux langages différents :

* Java est un langage de programmation POO (Programmation orienté Object) tandis que JavaScript est un langage de script OOP.
* Java permet de créer des applications qui sont exécutées sur une machine ou un navigateur virtuel tandis que le code JavaScript est exécuté uniquement sur un navigateur.

Figure 5: java vs JavaScript

* JavaScript est le souvent interprété coté client sur la machine sur laquelle tourne le navigateur qui analyse et affiche la page Web, alors que Java peut être complié ou interprété coté serveur.
* Un code JavaScript est appelé dans une page Web différemment d’une applet Java

### Présentation des outils*:*

#### Les éditeurs :

* + - * + SublimeText.
        + Atom.
        + Microsoft Visual Studio Code.
        + Notepad++.

#### Les Navigateurs :

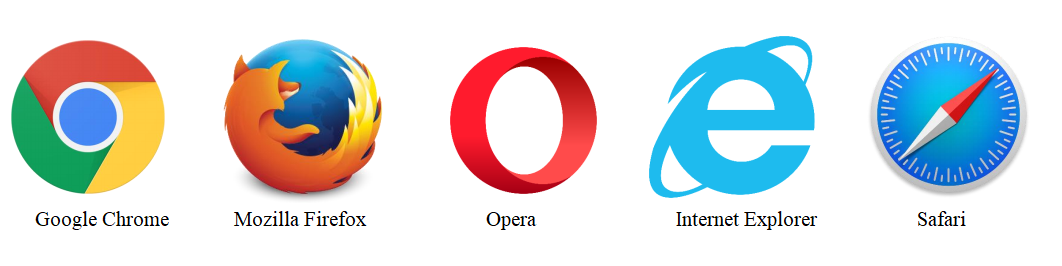


Figure 6 : Les Navigateurs

### Popularité des langages de programmation :

Figure 7: Popularité des langages de programmation

## **Conclusion :**

* Introduction au langage JavaScript.
* Les versions.
* Les outils.

# **Chapitre : Les bases de JavaScript**

## **Introduction :**

Nous exposons dans ce chapitre l’insertion de code à trois place et les outputs de ce langage, Nous présentons comment faire les commentaire de deux manières, Ensuite les variables et les quatre types des données , Nous terminons par les différentes opérateurs.

## **Emplacement du code JavaScript :**

* Où peut-on insérer un code JavaScript ?
* Un code JavaScript peut être placer dans:
* le corps **<body>**
* l’entête **<head>**
* **un fichier externe** portant l’extension « .js »

Dans l’entête **<head> :**

* On peut déclarer les fonctions JavaScript dans l’entête d’un document HTML et elles sont séparées du contenu.
* On utilisera la balise **<script>.**

Le code peut accéder à toutes les informations du document (ids, classes, etc.).

****Exemple de code dans Head :

Pour la fonction **« alert »,** Voir plus loin.



Figure 9 : L’insertion de code dans head et body

**Utiliser un fichier Externe :**

• Lorsque les fonctions JavaScript sont situées dans un fichier externe, il est possible de les utiliser dans plusieurs codes.  
• Dans ce cas, on va utiliser la balise **<script>**

Exemple de code dans un fichier externe :

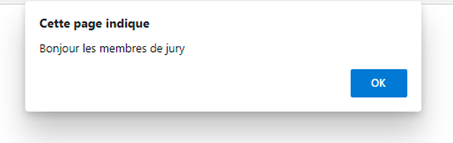


Figure 10 : L’insetion de code dans un fichier externe

## **JavaScript Output :**

• Les données sont affichées via :

* Une boite de dialogue en utilisant alert().
* prompt().
* “HTML output” en utilisant document.write().
* Le console du navigateur en utilisant console.log().
* Prendre une décision – confirm().

### Alert()

En JavaScript, une alerte est une fenêtre pop-up qui sert à afficher des informations.

Exemple : alert(‘EST Laayoune’);



Figure 11 : Alert()

### Prompt()

Similaire à “alert”, mais l’utilisateur aura la possibilité d’entrer des informations.



Figure 12 : Prompt()

### Ducomment.write()

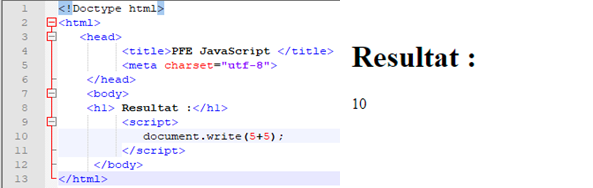
* Utile pour écrire du texte permanente dans la page.
* Document.write() écrire directement sur la page.

Figure 13 : Document.write()

### console.log()

* Cette option permet d’écrire des données dans le console du navigateur.

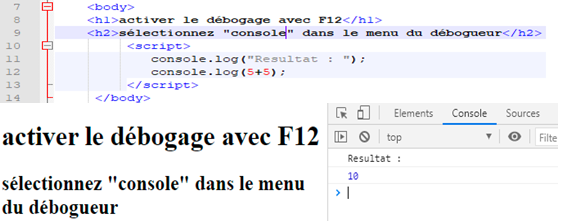


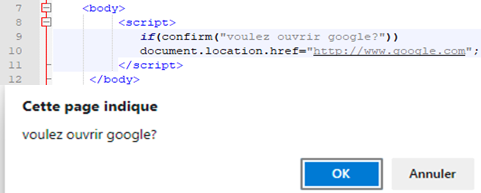
Figure 14 : Console.log()

Debugging :

* **Safari :** Préférences **-›** Avancé Vérifiez le menu de développement de spectacle dans Boîte de menu
* **Google Chrome :** plus d’outils **-›** Outils de développement **-›** Console
* **Firefox :** Tools **-›** Console
* **Internet Explorer :** F12

### confirm()

* confirm() affiche une boite de dialogue avec un message contenant les boutons  
  OK et Cancel .



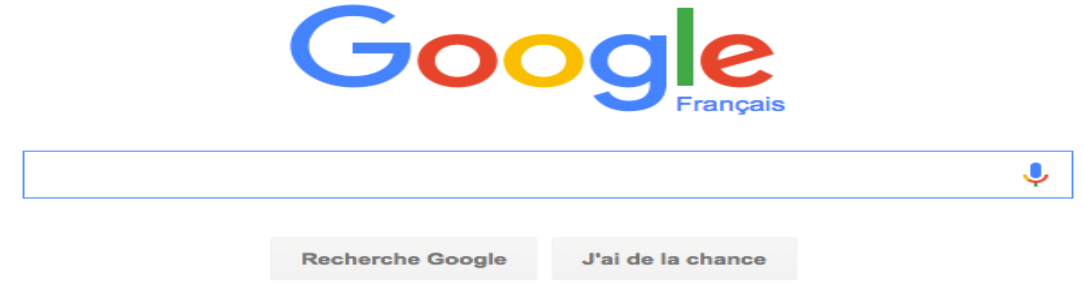


Figure 15 : Confirm()

## **Les commentaires :**

// un commentaire sur une ligne

/\* un commentaire plus

Long sur plusieurs lignes \*/

/\*la syntaxe multilignes peut être utilisée pour un commentaire monoligne \*/

/\* Par contre on ne peut pas /\* imbriquer des commentaires \*/ SyntaxError \*/

## **Les variables :**

### Déclaration:

* En JavaScript, les données sont gardées dans des variables
* Il existe trois types de déclarations de variable en JavaScript :
* **Let :**

On déclare une variable dont la portée est celle du bloc courant, éventuellement en initialisant sa valeur.

* **Var :**

On déclare une variable, éventuellement en initialisant sa valeur.

* **Const :**

On déclare une constante nommée, dont la portée est celle du bloc courant, accessible en lecture seule.

• Pour utiliser une variable, il faut la déclarer

**Var name ;**

Mot clé nom de variable

### Nom de variable :

• Constituer des lettres, chiffres, underscores et le signe dollar ($)  
 • Ne peut pas commencer par un chiffre  
 • Sensitive à la casse Name, name, naMe, NAME ne représente pas la même  
 variable  
 • Il est conseillé d’être significative

### Affectation :

• C’est ridicule de déclarer une variable sans l’utiliser par la suite

• L’affectation d’une valeur à une variable est faite par l’intermédiaire de l’opérateur “=” .

var name = “ESTL”;

Opérateur d’affectation

### Utilisation d’une variable :

Exemple :

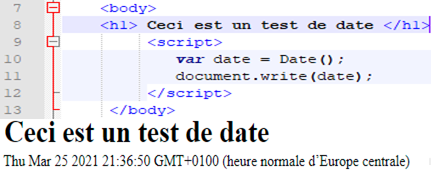


Figure 16 : Utilisation d’une variable avec Date()

## **Type de Données :**

### Type de Donneés – Number

Valeurs Numériques

* JavaScript possède un seul type de nombres
* Ecriture avec ou sans virgule  
   var number1 = 3.145;  
   var number2 = 200;

Exemple :

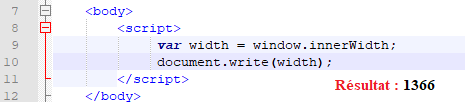
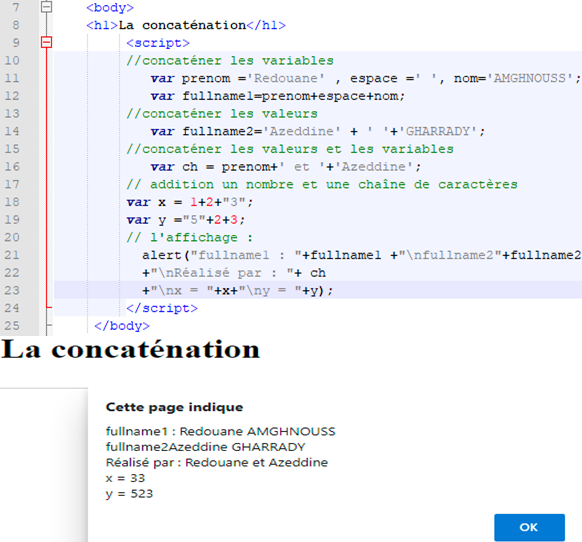


Figure 17 : Type number

### Type de Donneés – String

* A string est une chaine de caractères (lettres, nombres, ponctuation, …)
* Pour créer une chaine de caractères, on utilise les cotes “…” ou encore ‘….’
* var name = ‘Redouane’;
* var message = “j’habit à laayoune“;
* Concaténer, c’est donc « additionner » des chaînes de caractères
* En JavaScript, on va pouvoir concaténer grâce à l’opérateur « + »

Exemple :

Figure 18 :Type string avec la concaténation des variables

### Type de Donneés – Boolean

* En programmation, boolean est une valeur qui peut prendre deux valeurs : true ou false (vrai ou faux)
* var statues = true;
* var windowStatus = window.closed;

Exemple :

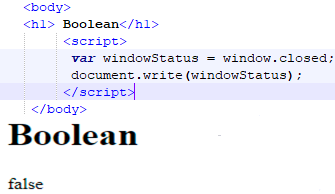


Figure 19 :Type Boolean

### Type de Donneés – Object

* Dans certains cas, les variables peuvent être complexe
* Un nœud dans un DOM est un bon exemple d’un objet
* var nom = document.getElementById(‘ID’);
* Voir après.

Remarque  
• Le type de la variable peut changer  
• On considère l’affectation suivante :  
 var name= “Hicham”;  
• Ensuite, l’affectation suivante :  
 name = 98;  
• Le type de la variable est immédiatement changé

### Typeof

On peut utiliser **TYPEOF** pour récupérer ou vérifier le type de la variable.

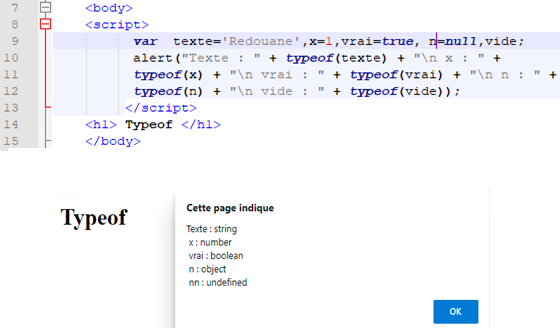


Figure 20 : La fonction Typeof()

## **Les opérateurs :**

|  |  |  |
| --- | --- | --- |
| Opérateur | Exemple | Valeur |
| Opérateurs d’affectation et des autre Types | | |
| = | X = 5 | 5 |
| ++ | Y = 5;  X++; | 6 |
| -- | X = 12;  X--; | 11 |
| += | X = 5;  X+=2; | 7 |
| Opérateurs sur les chaines de Caractères | | |
| + | X = “Hi” + “Azeddine”;  X =”Hi” + ‘5’; | “HiAzeddine”  “Hi5” |
| += | X = “Hi”  X+= “Redouane”; | “HiRedouane” |
| Opèrateurs Booléenn (soit X = 12) | | |
| = = | X == 5 | false |
| = = | X == 12 | true |
| != | X!=5 | true |
| > | X>12 | false |
| >= | X>=12 | true |
| < | X<12 | false |
| Opèrateurs Logique (soit X = 12) | | |
| && (and) | (15>x)&&(x<5) | false |
| | | (Or) | (15>x) | | (x>5) | true |
| ! (Not) | !(x==12) | false |

Table 2 :Les opérateurs

## **Conclusion :**

* Emplacemant de code.
* JavaScript Outputs.
* Les Commentaires.
* Les variables.
* Les types de Données.
* Les opérateurs.

# **Chapitre : Les structures et les fonctions**

## **Introduction :**

Nous exposons dans ce chapitre les instructions de contrôle et les conditions ternaires de ce langage, Ensuite nous présentons les boucle, Nous terminons par les foncyions el les tableaux .

## **Les instructions de contrôle :**

### Instruction if, if … else :

L'instruction if exécute une instruction si une condition donnée est vraie ou équivalente à vrai. Si la condition n'est pas vérifiée, il est possible d'utiliser une autre instruction.

Exemple :

On peut choisir d’afficher le message « Génie Informatique » si une variable **x** stocke une valeur égale 1,sinon d’afficher le message « Estlaayoune »

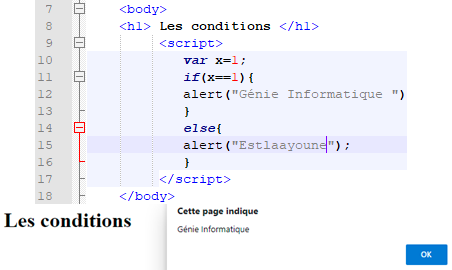


Figure 21: Les conditions if...else

### Les conditions ternaires :

**Syntaxe :**

Si condition true alors l'opérateur retourne valeur\_si\_true

Si condition false alors l'opérateur retourne valeur\_si\_false

**condition ? valeur\_si\_true : valeur\_si\_false ;**

### switch :

Parfois, dans vos conditions, vous voudrez pouvoir dissocier de nombreux cas afin de traiter chacun d’entre eux différemment.

Attention cependant à une limitation importante du switch, cette instruction ne peut tester que l’égalité de valeurs et non pas l’inégalité.

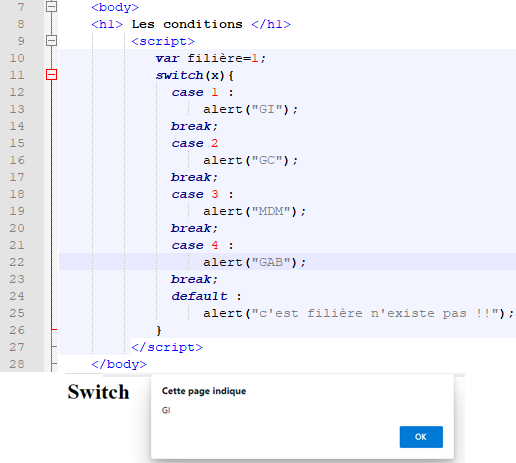


Figure 22: Switch

## **Les boucles :**

Nous allons étudier trois boucles dans cette partie :

### La boucle while :

La boucle while va nous permettre de répéter une série d’instructions tant qu’une condition donnée est vraie.

Exemple :

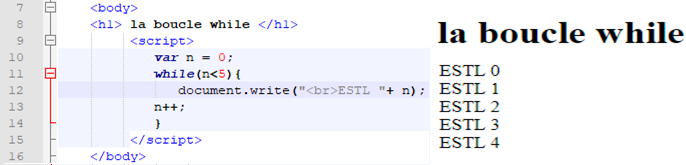


Figure 23 : La boucle while

### La boucle do… while :

En effet, dans une boucle while, si la condition donnée est fausse dès le départ, la boucle ne sera jamais exécutée.

En revanche, dans une boucle do… while, on effectuera toujours au moins un passage dans la boucle, même si la condition est fausse au départ.

### La boucle for :

La boucle for (« pour ») va nous permettre de répéter une série d’instructions à partir d’une valeur donnée, jusqu’à une certaine valeur seuil.

Dans une boucle for, nous allons tout d’abord devoir initialiser une variable, puis poser notre condition, et finalement incrémenter ou décrémenter notre variable.

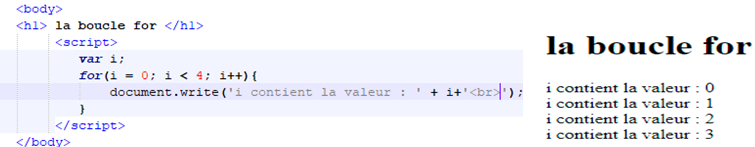


Figure 24 : La boucle for

## **Les fonctions**

**Présentation de fonctions :**

Une fonction correspond à un bloc de code dont le but est d’effectuer une tâche précise.

### Types de fonctions :

On peut dissocier deux grands types de fonctions en JavaScript :

* les fonctions natives: qui ont déjà été créées par d’autres développeurs et qui ont été intégrées au langage JavaScript lui même. alert();
* celles que nous allons créer nous mêmes.

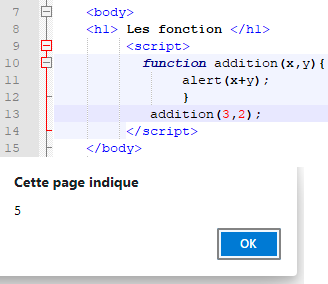
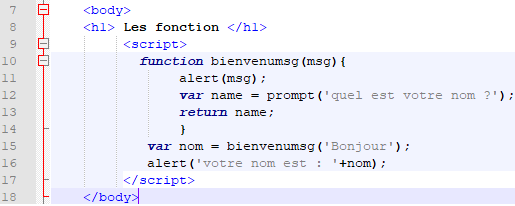


Figure 25 : la fonction addition

### Valeurs de Retour

### Quelques fonctions retournent des valeurs

### Ces valeurs peuvent être utilisées pour des instructions d’affectation ou des expressions conditionnelles



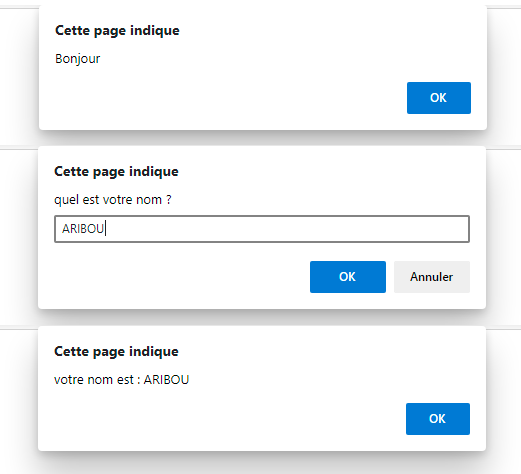


Figure 26 : la fonction retourne une valeur

## **Les tableaux :**

Les variables que nous avons utilisé jusqu’à maintenant stocke une seule information : Number, String, Boolean, or Object

Les tableaux a stocké plusieurs valeurs ou informations.

**utiliser les tableaux :**

Un tableau est un espace linéaire continu de stockage :

* On peut considérer qu’un tableau est un groupe de zones.
* Chaque zone possède son identité unique, appelé indice ou indexe.
* Le premier indice est 0.

### Déclaration d’un tableau :

var nombre = [1,2,3,4,5,6];

var nom = [‘Redouane’,’Azeddine’,’youssef’];

* On peut déclarer un tableau sans initialisation

var tab = new Array (10)

* Ce tableau peut contenir 10 valeurs
* On peut ajouter n’importe quel valeurs

**L’Accès à un Tableau :**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 80 | 81 | 10 | 11 | 62 | 0 | 0 | 999 | 125 | 19 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Table 3: Exemple de tableau

* Chaque valeurs est appelée un élément
* Les éléments sont référencés par des indices
* L’indice de départ est “0”
* nombre[0] indique la valeur 80 (1ère valeur)
* nombre[4] indique la valeur 62 (5ème valeur)

Remarque :

* En JavaScript, il n’est pas nécessaire que les éléments d’un tableau sont de même type
* Var Tab = [‘Hicham’, 1, “”, 10.4, “Salma”];

### Les fonctions de tableau :

Exemple :

**var ville= ["laayoune", "azrou", "agadir"];**

|  |  |  |
| --- | --- | --- |
| Nom de fonction | Description | Exemple |
| join(élément) | pour convertir un tableau à une chaine de caractères. | alert(ville.join("et"));  🡺résultat : laayoune et azrou et agadir |
| length | Pour récupérer la taille d’un tableau. | alert(ville.length); 🡺résultat : 3 |
| unshift(élément) | Pour ajouter un élément au début d’un tableau. | ville. unshift("smara")  🡺résultat : ["smara", "laayoune" , "azrou", "agadir" ]; |
| push(élément) | Pour ajouter un élément à la fin d’un tableau. | ville.push("dakhla");  🡺Résultat : ["laayoune", "azrou", "agadir","dakhla"]; |
| shift() | Pour supprimer un élément au début d’un tableau :  -Les indices du tableau sont automatiquement mis à jour | ville.shift()  🡺Résultat : ["azrou", "agadir" ]; |
| pop() | Pour supprimer un élément à la fin d’un tableau. | ville.pop()  🡺Résultat : ["laayoune", "azrou"]; |
| reverse() | • Le premier élément devient le dernier • Le dernier élément devient le premier. | ville.reverse()  🡺Résultat : ville = ["agadir" , "azrou" , " laayoune "]; |
| indexOf(élément) | pour trouver l’index du premier élément « élément » dans le tableau.  Si l’élément cherché ne se trouve pas dans le tableau, la valeur « -1 » sera retourné. | ville. indexOf("agadir")  🡺Résultat : 2  ville. indexOf("agadir",1)  🡺Résultat : -1 |
| lastIndexOf(élément) | pour trouver l’index du premier élément « élément » en commençant depuis la fin du tableau. | ville. indexOf("agadir")  🡺Résultat : 2  ville. indexOf("agadir",1)  🡺Résultat : -1 |
| slice(startPosition) | Pour extraire une partie d’un tableau. | Var R = ville.slice(1)  🡺Résultat : R contient: ["azrou", "agadir"] |
| splice() | -Pour supprimer/ajouter un élément d’un tableau | var R = ville.splice(1, 1);  Résultat : R contient: ["azrou",];  var R = ville.splice(1,0, "smara");  Résultat : R contient ["laayoune "," smara" , "azrou", "agadir"] |

Table 4 : Les fonctions de tableau

## Conclusion

* Les instructions de Controle.
* Les conditions ternaires.
* Les boucles.
* Les fonctions et Les tableaux.

# **Chapitre : Dom et les événements**

## **Introduction :**

Nous exposons dans ce chapitre l’introduction de Dom et sa structure, Nous présentons les APIs Spécifiques, Ensuite les différentes événements , Nous terminons par les Frameworks .

## **Le DOM :**

### Definition :

Le DOM est une interface de programmation pour des documents HTML ou XML qui représente le document (la page web actuelle) sous une forme qui permet aux langages de script comme le JavaScript d’y accéder et d’en manipuler le contenu et les styles.

Le DOM est ainsi une représentation structurée du document sous forme « d’arbre » crée automatiquement par le navigateur. Chaque branche de cet arbre se termine par ce qu’on appelle un nœud qui va contenir des objets. On va finalement pouvoir utiliser ces objets, leurs propriétés et leurs méthodes en JavaScript.

Le DOM n'est pas un langage de programmation, mais sans lui, le JavaScript ne pourrait avoir aucun modèle pour appréhender la page Web.

Exemple :

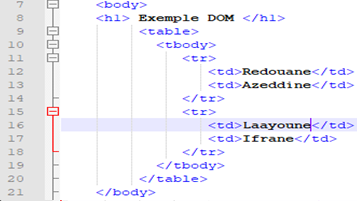


Figure 27 : Exemple de DOM

### Structure d’un DOM :

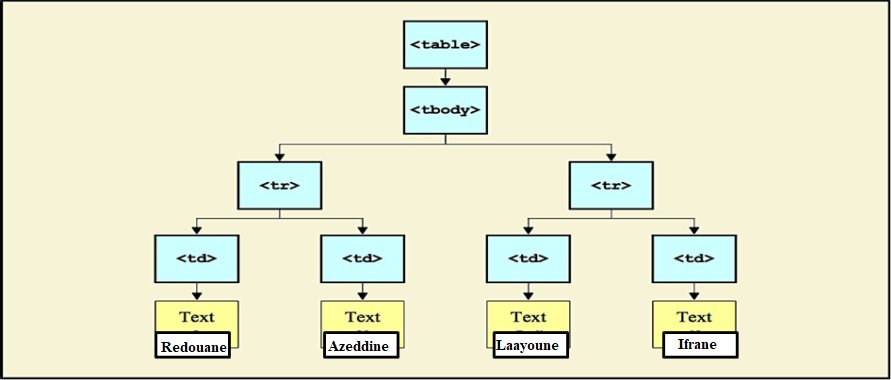


Figure 28 : Structure d'un DOM

**il fonctionne :**

* L’accès à un DOM est fait par l’intermédiaire d’une API (Application Programming Interface).
* L’API utilisée est la même indépendamment du langage de script et le navigateur utilisés.

### APIs Spécifiques :

* document.getElementById().
* document.getElementsByClassName().
* document.getElementsByTagName().
* document.getElementsByName().
* La méthode querySelector().
* La méthode querySelectorAll().
* element.innerHTML : pour obtenir le texte d’un paragraphe
* element.style Etc

**Accéder à un élément html en JavaScript**

La première méthode, **getElementById(),** va nous permettre de cibler un élément HTML possédant un attribut id en particulier. C’est certainement la méthode la plus utilisée.  
Si l’élément est trouvé, **getElementById()** va renvoyer l’élément en tant qu’objet.  
Si aucun élément n’est trouvé, la méthode renverra la valeur null.

Exemple :

- Pour **«onclick»,** Voir plus loin

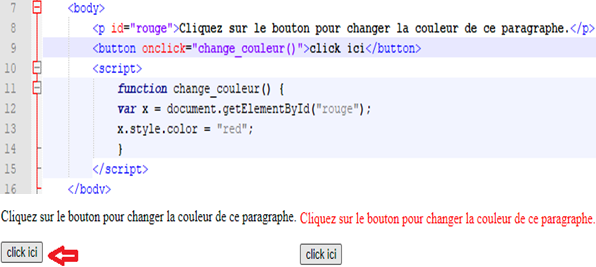
****

Figure 29: La méthode getElementById()

La méthode **getElementsByClassName()** va nous permettre d’accéder aux éléments HTML disposant d’un attribut class en particulier.  
 **getElementsByClassName()** va prendre la valeur d’un attribut class en argument. Elle va s’utiliser de manière similaire à la méthode précédente et renvoyer également un tableau.

Exemple :

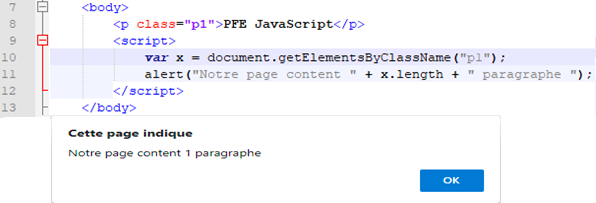


Figure 30 : La méthode getElementByClassName()

Les méthodes **querySelector()** et **querySelectorAll()** vont nous permettre d’accéder à des éléments HTML correspondant à un certain sélecteur CSS, que ce soit un id, une class.

**querySelector()** va renvoyer des informations relatives au premier élément trouvé correspondant au sélecteur CSS sélectionné.

tandis que **querySelectorAll()** va renvoyer des informations sur tous les éléments correspondants.

Exemple :

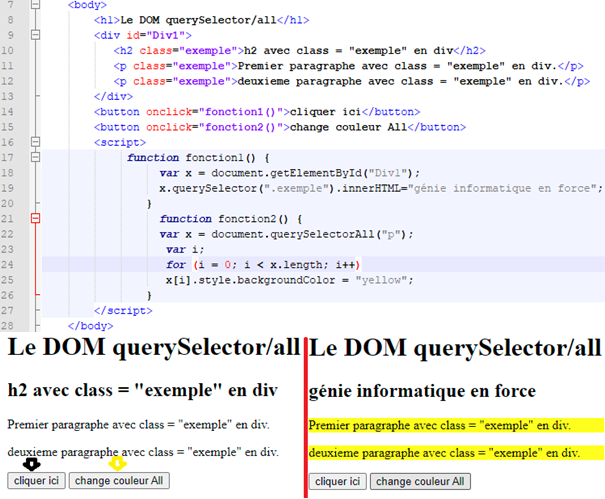
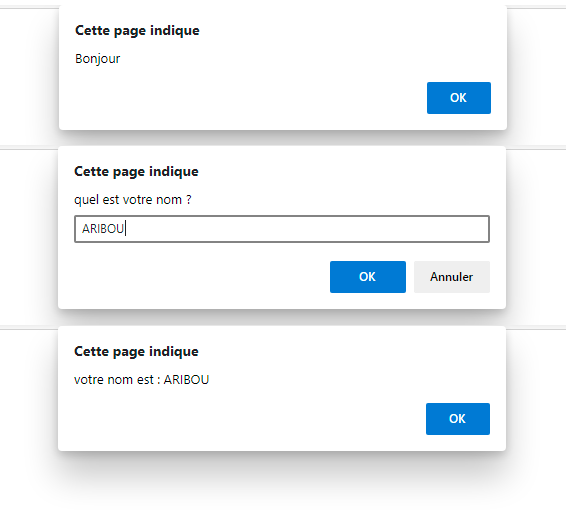


Figure 31 : La méthode querySelector et querySelectorAll()



## **Les événements :**

|  |  |  |
| --- | --- | --- |
| **Evélement** | **Description** | **Les balises concernées** |
| onclick | Lorsque l'utilisateu clique sur un bouton, un lien ou tout autre élément | <a>, <area>,<input>,etc |
| onLoad | Lorsque la page est chargée par lebrowser ou le navigateur | <Body>,<img>,etc |
| onUnload | Lorsque l'utilisateur quitte la page. | <Body>,etc |
| onmousover | Lorsque l'utilisateur place le pointeur de la souris sur un lien, une image ou tout autre élément. | <a>, <area>,<img>,etc |
| onmouseout | Lorsque le pointeur de la souris quitte un lien ou tout autre élément. | <a>, <area>,<input>,etc |
| onChange | Lorsque la valeur d'un champ de formulaire est modifiée | <input txt,pwd> ,<txtarea> ,etc |
| onFocus | Lorsqu’un élément de formulaire a le focus c-à-d devient la zone d'entrée active | <Body>, <input txt,pwd>, <txtarea> ,etc |
| onBlur | Lorsqu’un élément de formulaire perd le focus c-à-d que l'utilisateur clique hors du champs et que la zone d'entrée n'est plus active | <Body>, <input txt,pwd>, <txtarea> ,etc |
| onDbClick | Lorsque l'utilisateur double-clique sur un objet. | <a>, <area>,<input>,etc |
| onSubmit | Lorsque l'utilisateur clique sur le bouton Submit pour envoyer un formulaire | <Form> |
| onSelect | Lorsque l'utilisateur séléctionne un champ dans un élément de formulaire | <input txt,pwd> ,<txtarea>, etc |
| … | … | … |

Table 5 : Les événements

## **Les Frameworks de JavaScript :**

### Définition :

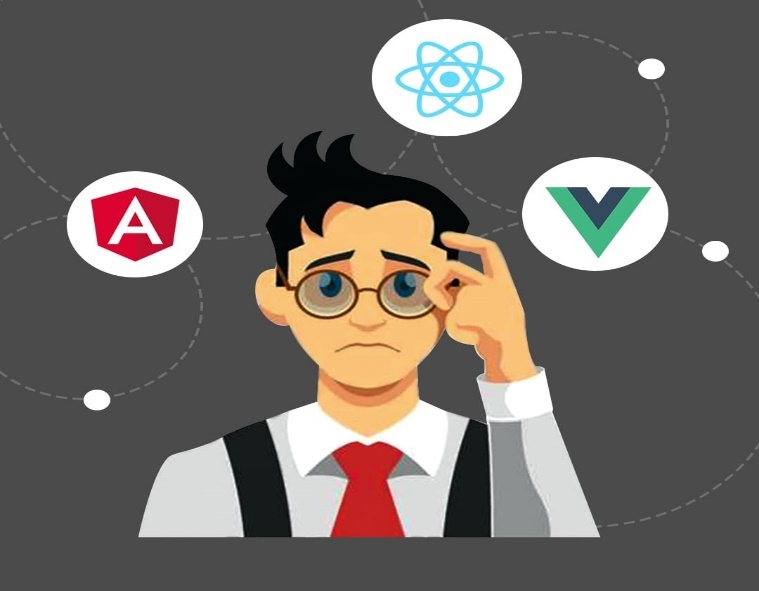
Un framework est une boite à outil doublée d’une philosophie de développement et d’architecture. Il permet de répondre à des besoins fonctionnels et de faciliter le travail des développeurs. Le framework JavaScript (ou JS) se base quant à lui sur le langage de programmation JavaScript. Il donne un cadre de travail au développeur, lui permettant ainsi de bénéficier d’une bibliothèque de composants déjà développés et d’une architecture déjà posée. Il répond donc à un souci de sécurisation et d’optimisation du développement du dispositif web.

Figure 32:Les framework

Difficile de connaître tous les frameworks, et de savoir rapidement lesquelles conviennent à son projet.

C’est pourquoi nous avons choisi de vous proposer de trois des frameworks les plus populaires.

### Les trois principaux Frameworks :

#### Angular

****Mis à flot par Google, Angular, aussi connu sous le nom Angular.js,est le doyen de cette sélection puisqu’il a vu le jour en 2010. Ce framework est basé sur TypeScript et reste sans aucun doute le plus utilisé actuellement, 10 ans après son lancement, notamment pour les sites et applications SPA (Single Page Application).

Figure 33 : Angular

Particulièrement efficace pour produire rapidement une quantité importante de code et pour tester différentes sections d’une application, Angular a l’avantage d’offrir de nombreuses possibilités de développement.

#### React

****React, que l’on appelle aussi ReactJS, est une bibliothèque JavaScript libre, créée par Facebook en 2013.

Figure 34 : React

Elle permet de concevoir des applications web via la création de composants dépendants d’un état constituant une portion HTML pour chaque changement d’état.

C’est un framework qui est recommandé par les entreprises proposant des solutions de développement JavaScript. Il s’agit d’un framework de développement web convivial. React peut être utilisé pour créer une interface graphique front et propose un modèle qui convient parfaitement à tous les développeurs web.

#### Vue.js

Vue.js, également appelé Vue, est le plus récent des trois puisqu’il a été lancé en 2014 par un ancien employé de Google.

Figure 35 : Vue

Ce framework dédié au front-end est réputé pour être parmi les plus rapides, ce qui en fait un excellent choix pour les projets assez lourds qui font de la performance une priorité.

## **Conclusion**

* DOM.
* Structure DOM.
* APIs Spécifiques.
* La gestion des événements.
* Les frameworks.

**Conclusion générale**

Cette recherche a été une occasion pour mettre en œuvre nos acquis théoriques et pratiques. Surtout JavaScript fait partie des langages « faciles à apprendre, difficiles à maitriser » et le premier langage de programmation utilisé pour la conception web et d'applications. Il est, plus précisément, un langage de script qui permet de dynamiser des sites "côté client". Javascript est principalement employé pour faire apparaître et bouger les éléments d'un site web sur nos ordinateurs et smartphones.

La réalisation de cette recherche nous a tellement aidés à développer notre créativité et notre esprit du travail en groupe, et de tester nos connaissances informatiques en pratique, et de nous focaliser sur l'auto apprentissage et la recherche pour surmonter les obstacles semés sur parcours.

Enfin, nous tiendrons à exprimer notre satisfaction d'avoir pu travailler dans de bonnes conditions matérielles et un environnement agréable tout en espérant que notre travail et notre rapport satisferont aussi bien notre encadrant, les membres de jury et nos enseignants.

**Bibliographique**

Livre Formation « Web Avancé ») javaScript full-stack Guillaume MOREL-BAILLY.

<https://www.pierre-giraud.com/javascript-apprendre-coder-cours/introduction/>

<https://developer.mozilla.org/fr/docs/Web/JavaScript/Guide/Introduction>

<https://www.w3schools.com/js/DEFAULT.asp>

<https://www.gchagnon.fr/cours/dhtml/introjavascript.html>

<http://edutechwiki.unige.ch/fr/JavaScript>