**INSFP OULED FAYET**

**Spécialité :** Développeur web et mobile / S2

**Module :** Les interfaces utilisateurs interactives

# Chapitre 1 : Introduction au langage JavaScript

**Introduction**: Aujourd’hui, le développement web est en croissance continue, car, nous pouvons créer des sites web ou des applications qui marchent sur des ordinateurs, tablettes ou Smartphones.

**1- Définition :** Un **langage informatique** est un ensemble de caractères, symboles et de règles à appliquer pour la conception et la mise en œuvre d'un [système d'information](https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27information).

On distingue plusieurs types, comme les l**angages de conception (**UML, Merise..), de **programmation** (Java, PHP..), de **description** (HTML, CSS..), de **requête** (SQL, OQL..), etc.

**2- Classification des langages de programmation :** La classification se fait selon plusieurs **critères** dont on site :

**2.1- Langages interprétés / compilés :** Un **script** écrit avec un **langage interprété** a besoin d'un interpréteur pour traduire au fur et à mesure les instructions du script. Si on a des instructions erronées, la partie erronée ne va pas s’exécutée, mais la partie correcte s’exécute de façon normale.

**Exemples** : HTML, CSS, JavaScript, Python …etc.

**-** Un **programme** écrit avec un **langage compilé** va être traduit une fois pour toute si on n’a pas d’erreurs par un compilateur, afin de générer un nouveau fichier exécutable. Il a comme avantages de garantir la sécurité du code source et la rapidité. **Exemples** : Pascal, Delphi, C++, Python …etc.

**- Interpréteur/ compilateur :** C’est un logiciel permettant **d’interpréter/ compiler** un programme en vérifiant si il n’y a pas d’erreurs (traduire le code source en code machine).

### Remarques :

### a- Certains langages appartiennent aux deux catégories interprété/compilé (LISP, Java, Python, ..) car dans certaines cas, on a besoin d’une phase de compilation pour créer certaines applications, et dans d’autres cas, on a besoin d’un interpréteur comme le cas du web.

**b**- On appelle un **code**, un **programme** (langage C, Pascal) ou bien un **script** (HTML, JavaScript).

**c-** On **exécute** un code par la machine (moteur), soit c’est un programme ou bien un script.

**2.2- Langages fonctionnels/ Orientés Objet :** On distingue deux familles de langages de **programmation**, selon la manière dont les instructions sont traitées :

### - Langage fonctionnel : Parfois appelé langage procédural ou déclaratif, est un langage dans lequel le programme est construit de fonctions/procédures prédéfinies ou déclarées, dont on fait appel au besoin.

### Exemples : Pascal, C…etc.

**- Langages Orientés Objet :** La programmation orientée objet est bien dirigée à l’usage de la notion d’objet et de classe. Une classe est une entité qui possède un nom, des attributs et des méthodes, or que l’objet est une instance de la classe. (On peut créer un objet sans passer par la classe).

**Exemples :** Delphi, C#, C++, JavaScript, PHP, Java…etc.

**3- Les types de développeurs :**

**I- Développeur front-end :** Il est censé de travailler dans les champs de compétences suivants :

**a- Design** (web designer). **b-Développement** HTML**,** CSS, JavaScript (web master).

Permet d’améliorer **l’aspect visuel ([front-end :](https://fr.wikipedia.org/wiki/D%C3%A9veloppement_web_frontal" \t "_blank)** [façade du site](https://fr.wikipedia.org/wiki/D%C3%A9veloppement_web_frontal" \t "_blank)**[)](https://fr.wikipedia.org/wiki/D%C3%A9veloppement_web_frontal" \t "_blank)**sur le navigateur **(client side :**côté client**)**.

**II- Développeur back-end :** Censé de travailler dans les champs de compétences suivants :

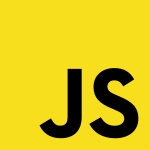
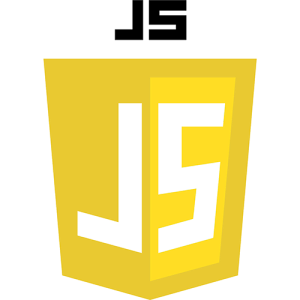
**a- Hébergement** du site web. **b-Administration** de la base de données du site web.

**c- Développement** Java, C++, Python, PHP, JavaScript, etc.

Permet de gérer le site **([back-end](https://fr.wikipedia.org/wiki/Backend" \t "_blank) :** partie invisible du site**)** sur le serveur **(server side :** côté serveur**)**.

**III- Développeur full-stack :** Capable d’intervenir aussi sur le front-end que le back-end.

**4- JavaScript (JS) :** JavaScript est un langage de programmation créé en 1995 par [Brendan Eich](https://fr.wikipedia.org/wiki/Brendan_Eich) pour le compte de la [Netscape Communications Corporation](https://fr.wikipedia.org/wiki/Netscape_Communications_Corporation), et standardisé sous les noms ECMA-262 et ISO/CEI 16262.

Une grande majorité des [sites web](https://fr.wikipedia.org/wiki/Site_web) l'utilisent, et la majorité des [navigateurs web](https://fr.wikipedia.org/wiki/Navigateur_web) disposent d'un [moteur JavaScript](https://fr.wikipedia.org/wiki/Moteur_JavaScript) dédié pour l'interpréter. La version actuelle de JavaScript est la version 13 édition - ECMA-262, publiée en juin 2022).

**Logo de JavaScript**

JavaScript support la programmation fonctionnelle et orientée objet.

JavaScript été utilisé à l’origine pour rendre dynamiques les pages web. Là où le HTML5 et le CSS3 permettent respectivement l’affichage et la mise en forme de vos éléments sur votre page web, le JavaScript rendra cette page interactive.

**Rôle de JavaScript dans le web :** Nous avons quelques exemples comme suite :

- Proposer un diaporama (changement d'image toute les x secondes, boutons pour mettre en pause, aller

à l'image précédente / suivante).

- Créer des applications interactives (par ex. tests et quiz).

- Vérification de formulaires, etc.

**5- Emplacement de JavaScript :** Nous avons deux possibilités pour écrire un code JavaScript.

**a-** A l’intérieur d’un document HTML.

|  |  |
| --- | --- |
|  |  |

**b-** Dans un fichier séparé portant l’extension **.js**.

**Remarque :** Il est recommandé d’utiliser un fichier JavaScript séparé pour des raisons de performances.

**a-A l’intérieur d’un document HTML :** Nous avons un exemple d’un code JavaScript permettant d’afficher un message dans la page web.

**<body>**

**<script language= "JavaScript" type="text/javascript">  
 document.write("Ceci est le message 1")   
</script>**

**</body>**

**b- Dans un fichier JavaScript appart :** Ici, nous allons créer un fichier dans notre éditeur de texte et l’enregistrer au format «**.js** », en plus, on doit lier ce fichier avec le fichier HTML.

- Code dans le fichier HTML :

**<script src="fichier.js">**

**</script>**

- Code dans le fichier JavaScript :

**document.write("Ceci est le message 2 ")**

**6- Notions de base sur JavaScript :** Nous avons les notions suivantes :

**a/** Un **identificateur** est le nom d’une variable, d’une fonction, ou d’une propriété.

**b/** Les minuscules et les majuscules ne sont pas équivalentes (JS est sensible à la casse ‘case sensitive’).

**Exemple** : **document.write("Bonjour");** n’est pas l’équivalent de : **Document.write("Bonjour");**

**c/** Pour déclarer une **variable**, on doit respecter les contraintes suivantes :

- Ne contient pas d’espace.

- Commence par une lettre, \_ ou $.

- Ne doit pas être un mot réservé comme par exemple : break, do, case, else, new, var, return, void, etc.

Les types de variables peuvent être :

**- Réels ou entiers**: 3.14 ou 200.

**- Chaînes de caractères** : "Cours de Programmation Web".

**- Logiques ou booléennes** : true/ false (vrai ou faux).

- **Null** : Qui signifie pas de valeur ou d'objet.

**- …etc.**

**d/ Les opérateurs :**

**- Opérateur d’affectation** : (**A = B**) mettre le contenu de B dans A.

1. **- Opérateurs arithmétiques** : **+, -, \*, /, %:** Modulo **++:** Incrémentation **--:** Décrémentation.

n = n + 1 ⇔n++ ⇔++n ⇔n += 1 (Incrémentation). Même principe pour la décrémentation.

a = a + b ⇔a += b, a = a - b ⇔a -= b, a = 2 \* a ⇔ a \*= 2 c=a% b.

**- Opérateurs de comparaison** : **== , !=, <, <=, >, >= ,** (=== : comparer l’égalité du type et du contenu entre deux variables), ( !== : comparer la différence du type et du contenu entre deux variables).

**Remarque** : Il faut s’assurer de la compatibilité des types lors des comparaisons.

**Opérateurs logiques** :(&& : et logique), (|| : ou logique), (! : négation).

**Opérateur de chaîne de caractères (+)** : Permet de **concaténer** deux chaînes de caractères.

**Exemple**: "Java" + "Script" donnera "JavaScript".

**Remarque :** La priorité des opérateurs est comme suite : (1) les parenthèses (2) négation (3) \*,/ ,modulo ,et logique (4) +,-,ou logique (5)> ,< , <=, >= , != , ==.

Pour deux opérateurs de même priorité, on commence par le plus à gauche.

**Exemple** : (8+ (5-3)\*8)/4 \* 3. Le résultat est 18.

**e/** Pour écrire un commentaire, vous pouvez utiliser le **double slash ( //)** au cours d’une ligne, ou bien utiliser /\*…\*/ sur plusieurs lignes.

**Exemple** :   
<script language= "JavaScript">

//voici un commentaire sur une ligne  
/\* Voici un commentaire

sur plusieurs lignes \*/  
</script>

**Remarques** :

**1-** Pour l’HTML : **<! - -** Ceci est un commentaire en HTML sur une ligne -->

**<! - -**Ceci est un commentaire en HTML

sur plusieurs lignes…………..-->

**2-** Le raccourci clavier **Ctrl+/** permet de créer ou d’annuler un commentaire sur une ligne, pour les langages HTML et JavaScript.

**f/** Pour sauter la ligne en JavaScript, on peut utiliser la balise <br/> dans la chaîne de caractère.

**document.write("Bonjour <br/> tout le monde");**

**g/** On peut utiliser une variable non déclarée.

**Exemple** : **var x = 10;**

**y=7 ;**

**document.write(x, y);**

**INSFP OULED FAYET**

**Spécialité :** Développeur web et mobile / S2

**Module :** Les interfaces utilisateurs interactives

**TP n° 1**

Ecrire en JavaScript, les scripts permettant de :

**1-** Faire la multiplication de deux nombres entiers.

**2-** Donner le triple d’un nombre entier.

**3-** Permuter le contenue de deux variables réels.

**4-** Renvoyer la racine carrée d’un entier.

**5-** Donner la moyenne de deux nombres entiers, cette moyenne doit être arrondie.

**6-** Donner la partie entière de la moyenne de deux nombres entiers.

**Solution** :

**1-** Multiplication de deux nombres entiers.

var a=3, b=7;

c=a\*b;

document.write(c);

**2-** Triple d’un nombre entier.

var a=10;

document.write(a\*3);

**3-** Permutation de nombre réels;

var a=3.2, b=7.5;

c=a;

a=b;

b=c;

document.write(a+'<br> '+b) ;

**4-** Racine carrée d’un entier.

var a=27;

b=Math.sqrt(a);

document.write(b);

**5-** Moyenne arrondie de 2 nombres entiers.

var a=8, b=13;

c=Math.round((a+b)/2);

document.write(c);

**6-** Partie entière de la moyenne de 2 entiers.

var a=8, b=13;

c=Math.trunc((a+b)/2);

document.write(c);

**INSFP OULED FAYET**

**Spécialité :** Développeur web et mobile / S2

**Module :** Les interfaces utilisateurs interactives

**Chapitre 2 : Les instructions conditionnelles et répétitives**

JavaScript définit plusieurs structures de contrôle afin de réaliser des conditions et des boucles.

**1- Les conditions :** Il existe deux types de structures de conditions.

**1.1 L’instruction if** : Son rôle est de vérifier une condition et de réaliser des traitements en cas de validité. La syntaxe est la suivante :

**if** (condition)

{

Bloc **1** d’instructions

}

**else**

{

Bloc **2** d’instructions

}

**Remarques** :

**a**- Il est possible d’utiliser if sans else.

**b-** Il est possible d’utiliser autant de if else imbriqués que nécessaire.

**c-** L’**opérateur conditionnel** peut remplacer if-else, selon la syntaxe de l’exemple suivant :

**x = (y > z) ? 5 : 3 ;** Si (**y** > **z**) alors **x**=**5**, sinon **x**=**3**.

**Exemple 1 :** Ecrire un script en JS qui détermine si un nombre entier est supérieur à 10.

**var** x =17;

**if** (x>10)

{

document.write(x+ ' est supérieur à dix' );

}

**Exemple 2 :** Ecrire un script en JS qui détermine si un nombre entier est supérieur ou inférieur à 10.

**var** x =17;

**if** (x>10)

{

document.write (x+ ' est supérieur à dix' );

}

**else**

**{**

document.write (x+ ' est inférieur à dix' );

}

**Exemple 3 :** Ecrire un script en JS qui détermine si un nombre entier est supérieur, inférieur ou égal à 10.

**var** x =17;

**if** (x>10)

**{**

document.write (x+ ' est supérieur à dix' );

**}**

**else**

**{**

**if** (x==10)

**{**

document.write (x+ ' est égal à dix' );

**}**

**else**

{

document.write (x+ ' est inférieur à dix' );

}

**}**

**1.2 Les instructions switch, case et default** : Cela permet de faire des traitements en fonction de la valeur d’une variable (variable = valeur) avec la syntaxe suivante :

**switch** (variable)

{

**case** valeur 1:

Bloc **1** d’instructions

**break**;

**case** valeur 2:

Bloc **2** d’instructions

**break**;

**default**:

Bloc **3** d’instructions

}

**Exemple :** Ecrire un script en JS qui renvoie le semestre de formation d’un stagiaire.

var s=2;

**switch** (s)

{

**case** 1:

document.write ('Vous êtes en semestre '+s);

**break**;

**case** 2:

document.write ('Vous êtes en semestre '+s);

**break**;

**default**:

document.write ('Vous êtes en semestre '+s);

}

**2-Les instructions répétitives :**

**2.1-L’instruction for :** Cette instruction permet d’incrémenter une variable à partir d'une valeur initiale jusqu'à une valeur finale et d'exécuter des instructions entre chaque incrémentation. La syntaxe est la suivante :

**for** (condition de début ; condition de fin ; incrémentation/décrémentation)

**{**

Blocd’instructions

**}**

**Exemple :** Ecrire un script en JS qui calcule le factoriel d’un entier. (Exp : factoriel (3)= 3\*2\*1=6).

**var** n=3; f=1;

**for** ( var i=1; i<=n; i++ )

**{**

f=f\*i;

**}**

**document.write**("Le factoriel de "+n+" est "+f );

**2.2-L’instruction while :** Ce type de boucle, contrairement à la précédente, évalue une condition avant d'exécuter des instructions comme suite :

**while** (condition)

**{**

Blocd’instructions

**}**

**Exemple :** Reprendre l’exercice du factoriel avec la boucle while.

**var** n=3; f=1; i=1;

**while** ( i<=n )

{

f=f\*i;

i=i+1;

}

**document.write**("Le factoriel de "+n+" est "+f);

**2.3-L’instruction do :** Cette boucle effectue les instructions placées entre deux accolades et évalue à chaque fois une condition de type booléenne avec while avant de continuer la boucle pour décider l'arrêt.

La syntaxe est comme suite :

**do**

{

Blocd’instructions

}

**while** (condition)

**Exemple :** Reprendre l’exercice du factoriel avec la boucle do**.**

**var** n=3; f=1; i=1;

do

{

f=f\*i;

i=i+1;

}

**while**(i<=n)

**document.write** ("Le factoriel de "+n+" est "+f);

**Remarques :**

**1-** Pour la boucle while, les instructions peuvent ne pas être exécutées si la condition n’est pas remplit, contrairement à la boucle do où au moins une instruction est exécutée.

**2-** Il est possible d’imbriquer les instructions conditionnelles et répétitives selon le besoin.

**3-** Pour la boucle for, il est possible d’utiliser plusieurs compteurs en même temps.

**Exemple**: **for (i=1, j=3 ; i<=n, j<=m ; i++, j++)**

**4-** Il est possible d’utiliser **break** avec la boucle **for** pour arrêter immédiatement la boucle avec une certaine condition comme le montre l’exemple ci-dessous.

**Exemple**: Afficher les tous les multiples de 3 qui sont inférieurs à 10.

**var i;**

**for (i=1; i<20; i++)**

**{**

**if (3\*i > 10)**

**break;**

**document.write(3\*i);**

**}**

**5-** Il est possible d’utiliser **continue** avec la boucle **for** pour pouvoir continuer la boucle dans certains cas exceptionnels, comme le montre l’exemple ci-dessous.

**Exemple**: Afficher l’inverse des nombres dans l’intervalle [-3, 3] (1 divisé par 0 n’est pas possible).

**var i;**

**for(i=-3; i<=3; i++)**

**{**

**if (i == 0)**

**continue;**

**document.write(1/i+);**

**}**

**NB : NaN** signifie (Not a Number) pas un nombre, message qui peut être affiché en divisant un nombre sur une chaine de caractères.

**INSFP OULED FAYET**

**Spécialité :** Développeur web et mobile / S2

**Module :** Les interfaces utilisateurs interactives

**TP n° 2**

Ecrire en JavaScript, les scripts permettant de :

**1-** Vérifier si une année donnée est bissextile ou non, en utilisant if-else.

L’année bissextile est une année spéciale contenant un jour supplémentaire dans une année(366).

Une année est considérée comme bissextile si elle est divisible par 4 mais non divisible par 100.

**2-** Trouver un maximum entre trois nombres en utilisant if-else imbriquée.

**3-** En utilisant l’instruction switch, afficher en lettre un nombre donné en chiffre de 1 jusqu’à 3 et afficher le message « supérieur à trois » pour les nombres supérieur à 3.

**4**- Calculer la somme de N premier nombre entier avec les trois types de boucles.

**5-** En utilisant la boucle for, calculez la puissance n d’un entier x (**xn**).

**6-** Afficher les nombres impairs compris entre 1 et N, en utilisant la boucle for.

**7-** Afficher les nombres pairs compris entre 1 et N, en utilisant la boucle while.