

## Chap 1 : Introduction

### Sources :

**JavaScript, Des fondamentaux aux concepts avancés - Emmanuel Gutierrez - Editions Eni**  
**Algorithmique - Techniques fondamentales de programmation - Franck EBEL - Sébastien ROHAUT – Editions Eni**  
**Apprendre à développer avec JavaScript - Christian VIGOUROUX - Editions ENI**

Pour obtenir un résultat donné, il faut généralement suivre une méthode, une certaine logique.

Il en est de même de la programmation. Il existe plusieurs langages de programmation parfois assez simples, qui peuvent donner un temps l'illusion que vous savez programmer. En entreprise même, certains développeurs ont quelques connaissances confuses de Visual Basic, de Delphi ou de Windev. Le résultat peut être catastrophique. Les publicités sont alléchantes mais trompeuses.

Les bons programmeurs, y compris les autodidactes, ont tous à un moment ou un autre eu affaire avec les algorithmes, car il existe en programmation une multitude de moyens d'arriver à un résultat, mais très peu pour obtenir le meilleur résultat possible.

De plus apprendre la programmation, ce n'est pas seulement savoir écrire un programme, c'est aussi comprendre de quoi il est fait, comment il est fait et ce qu'il fait. Un programme est essentiellement constitué d'expressions et d'instructions.

Un programmeur doit donc acquérir des notions d'algorithmique.

### Algorithme :

**Un procédé systématique qui permet de traiter des informations s'appelle un *algorithme*.**

Ainsi, on peut parler d'algorithmes de recherche d'un mot dans un dictionnaire, d'algorithmes de chiffrement et de déchiffrement, d'algorithmes pour faire des additions et des multiplications, etc. De manière plus générale, un algorithme est un procédé systématique qui permet de faire quelque chose. Par exemple une recette de cuisine est un algorithme

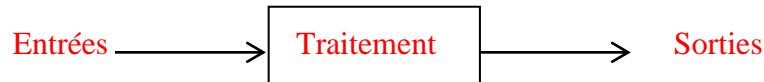
### Programmation :

**Traduction d'un algorithme dans un langage informatique.**

## 1) L'algorithme

Un algorithme est une suite d'instructions qui, quand elles sont exécutées correctement aboutissent au résultat attendu. C'est un énoncé dans un langage clair, bien défini et ordonné qui permet de résoudre un problème, le plus souvent par calcul

Quel est le principe d'un algorithme ? Il utilise des données fournies par l'utilisateur (*les entrées*), réalise diverses opérations dessus (*traitement*) et produit un résultat (*la sortie*).



Exemple 1 : Calcul d'un montant de TVA

Etape	Mise en œuvre	Exemple
<b>Entrées</b>	Prix HT, TauxTVA	100, 0.20
<b>Traitements</b>	MontantTVA = PrixHT x TauxTVA	100*0.20
<b>Résultat</b>	Affichage du montant TVA	20.0

Principe de base :

Plusieurs exécutions d'un algorithme avec les mêmes données d'entrée fourniront toujours le même résultat (l'algorithme est *déterministe*).

Les algorithmes sont utilisés par des humains, ce qui permet une formulation assez libre.

En pratique, l'algorithme n'est pas écrit en français, mais dans une forme codifiée.

A partir d'un algorithme, nous pouvons le traduire simplement dans le langage que nous souhaitons. Nous sommes donc dans une phase de conception qui reste théorique. C'est ce pseudo code que vous devrez utiliser à l'étude de cas lors de votre examen.

Exemple d'un algorithme :

Algo Bienvenue

DEBUT

Afficher ("Hello World")

FIN

Commentaires :

- Il faut donner un nom à l'algorithme
- Il faut indiquer le début et la fin du traitement
- L'instruction Afficher permet d'afficher un message.

## 2) Le programme

Nous avons donc vu qu'un algorithme programme était une suite d'opérations. Ces opérations seront effectuées par votre ordinateur.

En informatique, nous employons les termes « instruction » plutôt qu'opération et de programme plutôt qu'algorithme.

Ces instructions sont exécutées par le microprocesseur de l'ordinateur. Celui-ci étant un composant électronique il ne reconnaît que des suites de 0 et de 1 qui correspondent à l'absence ou non de courant électrique.

Nous nous retrouvons donc devant un problème :

Ce qui est intelligible pour l'humain (l'algorithme écrit plus ou moins en français) est incompréhensible pour l'ordinateur ; et les programmes informatiques (constitués de 0 et de 1) sont illisibles pour le programmeur.

Des langages intermédiaires entre l'algorithme et le langage machine ont donc été développés. Ces langages, appelés **langages de haut niveau**, ont une grammaire et une syntaxe plus proches du programmeur ; ils utilisent des termes anglais. Des outils spéciaux, appelés compilateurs, sont donc chargés de transcrire automatiquement le langage de haut niveau en langage assembleur.

Les compilateurs sont en fait des programmes. Ils prennent en entrée un programme écrit en langage de haut niveau (le programme source) et produisent comme résultat le même programme exprimé en langage machine (l'exécutable).

Il existe 3 classifications de langage :

langages **compilés**: le code source est analysé par un programme appelé compilateur qui va générer du code binaire que l'ordinateur sera capable d'exécuter (attention! ce code binaire sera différent d'un ordinateur à l'autre et nécessite d'être recompilé pour chaque ordinateur).

**Les langages comme le C ou le C++ sont des langages compilés**

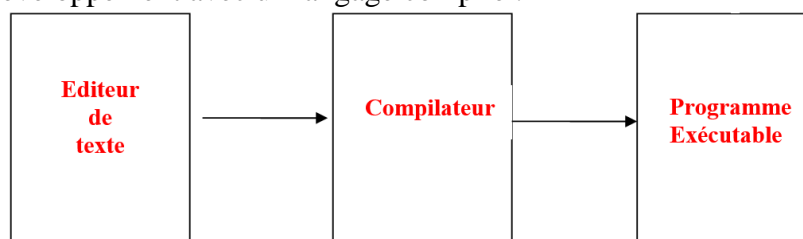
langages **précompilés**: le code source est compilé partiellement, dans un code plus simple mais qui n'est pas du code binaire (ce code est valable pour tous les types de machines). Ce code intermédiaire sera ensuite interprété et exécuté par une machine virtuelle (propre à chaque ordinateur).

**Les langages comme le C# ou le Java sont des langages précompilés**

langages **interprétés**: il n'y a pas de compilation ni de précompilation au préalable. Le code source reste tel quel. Si on veut l'exécuter, il faut faire appel à un interpréteur qui se chargera de l'analyser et de réaliser les actions qu'il contient.

**Le langage Javascript est un langage interprété**

Processus de développement avec un langage compilé :

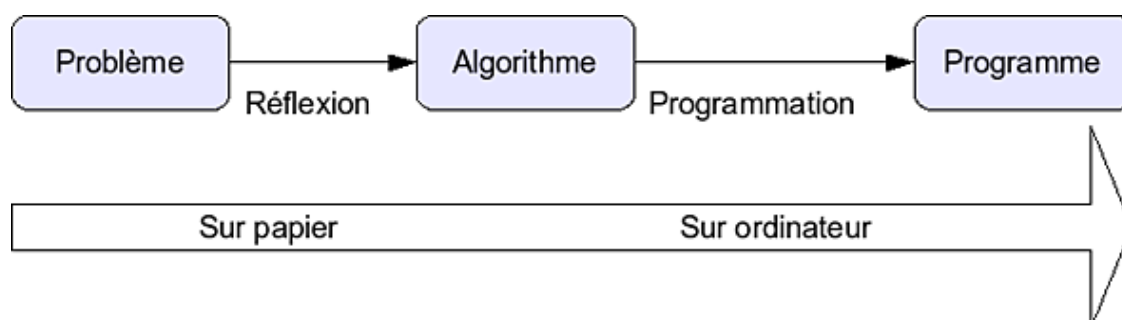


Généralement, l'éditeur de texte et le compilateur sont confondus car les outils de développement intégrés modernes vous fournissent le nécessaire pour programmer. Les langages les plus connus actuellement sont le C++, le Java, le C#, le Visual Basic, le Delphi, le C.

Extraits de code dans différents langages

<b>Algorithme</b> Algo Bienvenue DEBUT Afficher("Hello World") FIN	<b>Programme C#</b> <pre>namespace MonPremierProgramme {     class Program     {         static void Main(string[] args)         {             Console.WriteLine("Hello World");         }     } }</pre>
<b>En assembleur x86 sous DOS</b> <pre>Cseg segment assume cs:cseg, ds:cseg org 100h main proc jmp debut mess db 'Hello world!\$' debut: mov dx, offset mess mov ah, 9 int 21h</pre>	<b>En COBOL</b> <pre>IDENTIFICATION DIVISION. PROGRAM-ID. HELLO-WORLD. ENVIRONMENT DIVISION. DATA DIVISION. PROCEDURE DIVISION. DISPLAY "Hello world!". STOP RUN.</pre>
<b>En Basic originel</b> <pre>10 PRINT "Hello world!" 20 END</pre>	<b>En shell Unix</b> <pre>echo "Hello world!"</pre>
<b>En langage C</b> <pre>#include &lt;stdio.h&gt;  int main(int argc, char **argv) {     printf("Hello world!\n");     return 0; }</pre>	<b>En langage C++</b> <pre>#include &lt;iostream&gt;  int main() {     std::cout &lt; "Hello world!" &lt; std::endl;     return 0; }</pre>
<b>En PHP</b> <pre>&lt;?php print ("Hello world!"); ?&gt;</pre>	<b>En Java</b> <pre>public class HelloWorld {     public static void main(String[] args) {         System.out.println("Hello world!");     } }</pre>
<b>En Pascal</b> <pre>program Bonjour; begin     WriteLn('Hello world!'); end.</pre>	

Synthèse :



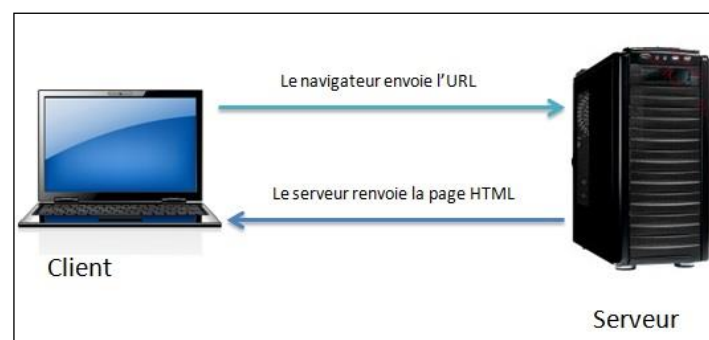
### 3) Le javascript

#### a) Définition et rapide historique

JavaScript, que l'on peut abréger par le sigle JS, est un langage inventé par Brendan Eich en 1995 qui sert essentiellement à programmer des traitements côté "client" dans les développements web.

JavaScript est un langage universel qui se retrouve dans de nombreuses pages HTML, en complément de ce code. Grâce à JavaScript, les pages HTML sont plus riches et disposent de nombreuses fonctionnalités supplémentaires. En particulier les pages pourront réagir à des sollicitations de l'utilisateur comme le choix dans une liste déroulante, une sélection d'une case à cocher, un clic sur un bouton dans un formulaire.

Le code JavaScript, stocké sous forme de fichiers sur un serveur web, est rapatrié, via le protocole HTTP, dans le navigateur du poste "client" sur appel de l'utilisateur par la saisie d'une URL ou par un clic sur un lien hypertexte. Le code JavaScript est décrypté (interprété) par un moteur dédié embarqué dans le navigateur.



Le langage JavaScript a subi de nombreuses évolutions au fil du temps et est pris en compte à peu près complètement par les principaux navigateurs (Microsoft Internet Explorer, Mozilla Firefox, Opera, Safari, Google Chrome...) équipant les ordinateurs et ceci quel que soit leur système d'exploitation (différentes versions de Microsoft Windows, Unix, Linux, Mac OS X...).

Le code JavaScript est décrypté (interprété) par un moteur dédié embarqué dans le navigateur. Chaque navigateur possède son interpréteur Javascript (Chakra chez Microsoft, SpiderMonkey - et ses dérivés TraceMonkey, JägerMonkey... - chez Mozilla, JavaScriptCore - et ses dérivés SquirrelFish, Nitro... - chez Apple, V8 chez Chrome...)

JavaScript a longtemps été décrié car il avait la réputation d'être difficile à déboguer, d'avoir des comportements très différents d'un navigateur à l'autre. De nombreuses améliorations ont été apportées au fil des versions (JavaScript 2.0 actuellement).

Vous pourrez trouver un descriptif complet des nouveautés de JavaScript sur le site Mozilla (<https://developer.mozilla.org/fr/docs/JavaScript>).

Enfin JavaScript a retrouvé ses lettres de noblesse avec l'émergence de nombreuses bibliothèques de qualité (cf. <http://JavaScript.developpez.com/cours/librairies-JavaScript-vraiment-utiles/jQuery>) et avec l'apparition de nombreux frameworks (Ajax, jQuery, Dojo...).

## b) Avantages/Inconvénients

## Avantages

Savoir rédiger des scripts en JavaScript, c'est permettre aux visiteurs de vos pages HTML d'accéder à d'autres fonctionnalités, à d'autres services et d'améliorer ainsi la professionnalisation d'un site.

De plus, ce langage évite des dialogues client-serveur inutiles.

Une autre particularité de JavaScript réside dans le fait qu'il ne nécessite pas d'éditeur particulier ni de compilateur.

## Inconvénients

Comme c'est un langage de script qui s'exécute côté client, il lui est impossible de s'interfacer avec une base de données de type Mysql ou SQL.

Autre point important, JavaScript n'est pas en mesure d'écrire ou de lire sur le disque dur du poste client (hormis les cookies qui ne sont que de petits fichiers texte)

## c) Comparatif java/javascript

Parlons aussi de la confusion chez les développeurs débutants entre JavaScript et Java. Il ne faut tout simplement pas confondre Java et JavaScript. JavaScript est un langage de programmation de scripts développé par Netscape Communications. Java quant à lui est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems. La syntaxe des deux langages est proche car elles dérivent toutes les deux du C++.

	Java	JavaScript
<b>Auteur</b>	Développé par Sun Microsystems	Développé par Netscape Communications
<b>Format</b>	Compilé sous forme de byte-code	Interprété
<b>Stockage</b>	Applet téléchargé comme un élément de la page web	Source inséré dans la page web
<b>Utilisation</b>	Utilisable pour développer tous les types d'applications	Utilisable uniquement pour "dynamiser" les pages web
<b>Exécution</b>	Exécuté dans la JVM ( <i>Java Virtual Machine</i> ) du navigateur	Exécuté par le navigateur
<b>Programmation Orientée Objet (POO)</b>	Orienté objet	Manipule des objets mais ne permet pas d'en définir
<b>Typage</b>	Fortement typé	Pas de contrôle de type

#### d) Outils

##### Outillage nécessaire

Au minimum, il faut :

- Un éditeur de texte
- Un navigateur

Il est fréquent de s'entendre dire "le développement web requiert peu d'outillage". En première approche, il est vrai que pour développer des pages web par l'intermédiaire des langages HTML et JavaScript il est possible de se contenter d'outils minimalistes (et gratuits).

Une connexion Internet n'est même pas obligatoire.

Par contre, pour des développements conséquents, il peut être utile de disposer de logiciels plus évolués (gratuits ou payants). Pour l'élaboration de vos scripts, vous pouvez utiliser des éditeurs de texte (gratuits) offrant des prestations plus complètes (coloration syntaxique, gestion multifichier, complétion automatique, accès FTP intégré, prise en compte de multiples langages web...)

Il existe aussi de nombreux générateurs de code ou IDE gratuits ou payants, le plus connu étant bien évidemment Adobe Dreamweaver

#### e) Premier script

Positionnement du JavaScript face à d'autres technologies de développement web (HTML, CSS, PHP...)

Le JavaScript n'est que très rarement utilisé seul. Les séquences de code JavaScript sont la plupart du temps imbriquées dans des scripts HTML (HyperText Markup Language). Pour dire les choses simplement, dans un affichage de pages web, le langage HTML décrit principalement le contenu alors que le JavaScript, comme indiqué précédemment, gère l'interactivité.

Très souvent la mise en forme des informations affichées (séquences de texte, tableaux, images...) n'est pas assurée directement par des directives (balises) du langage HTML mais par une technologie complémentaire (il est sans doute exagéré de parler de langage), le CSS (Cascading Style Sheets) encore appelé "feuilles de style en cascade".

Il est aussi très important de comprendre que le langage JavaScript est essentiellement orienté "client". Il est donc inutile de lui demander d'agir du côté "serveur". Il n'est pas en mesure de solliciter des bases de données hébergées sur le serveur pour aller y chercher des données stockées dans des tables. Il existe justement des langages supplémentaires, très largement employés dans les projets web, comme le PHP, Microsoft ASP.NET ou encore Java, dont ce sera justement le rôle. Ces langages, via des requêtes SQL, interrogent les bases de données et un interpréteur (logiciel côté "serveur") génère à la volée du code HTML, contenant les informations lues dans les tables, transmis ensuite via le protocole HTTP au navigateur du poste "client". Il restera alors au navigateur à décoder (décrypter) le code HTML (et JavaScript le cas échéant) pour un affichage aisément consultable par un opérateur humain.

## Insérer du code JavaScript

### *Javascript interne*

Un script JavaScript peut se trouver dans le corps ou dans l'en-tête d'un document html.

Cependant, par convention, les scripts se retrouvent plus fréquemment dans la partie head de la page. Leur exécution peut alors être immédiate (au chargement de la page) ou différée (clic sur un bouton, par exemple). Il faudra, dans ce cas, utiliser la programmation événementielle et les fonctions pour que le code s'exécute.

Il suffit simplement d'une page HTML à l'intérieur de laquelle vous ajoutez des lignes rédigées en JavaScript entre deux balises. La première balise indique au navigateur le début du script JavaScript et la seconde en indique la fin. Les deux balises à utiliser sont les suivantes :

Au début du script :

```
<script>
```

et à la fin du script

```
</script>
```

Si, malgré tout, vous souhaitez faire passer des informations aux visiteurs ayant un navigateur ne supportant pas JavaScript, il faut le faire entre les balises `<noscript>` et `</noscript>` que l'on place juste après

### Exemple :

```
<script>
document.write ("bonjour");
</script>
<noscript> votre navigateur ne peut pas lire le code JavaScript</noscript>
```

### *Javascript Externe*

les blocs de script JavaScript peuvent être écrits dans un fichier JavaScript au format .js, totalement dissocié du code HTML de la page.

Concrètement, le script JavaScript est rédigé dans un fichier particulier à l'aide d'un éditeur de texte, par exemple, et doit être enregistré sans formatage au format .js. Si l'éditeur de texte ne propose pas par défaut cette extension, il suffit de l'ajouter lors de la sauvegarde du fichier. Afin de pouvoir retrouver facilement le rôle de votre script, il convient de le nommer avec un nom explicite.

Par la suite, il est très simple d'appeler le fichier JavaScript dans la page HTML en respectant la syntaxe suivante :

```
<script src="fichier_javascript.js"></script>
```

Bien sûr, le fichier devra être présent dans le même répertoire de votre disque dur ou du serveur que le fichier HTML appelant.

Ecrire votre premier programme affichant le message Hello Word.