

Introduction a javascript

Table des matières

I. Contexte	3
II. Présentation de JavaScript	3
A. Introduction.....	3
B. Langage côté client	4
C. Langage interprété vs langage compilé	5
D. Exercice : Quiz.....	7
III. Histoire et ECMAScript6	8
A. Histoire et ECMAScript6	8
B. Exercice : Quiz.....	9
IV. Essentiel	10
V. Auto-évaluation	11
A. Exercice	11
B. Test.....	11
Solutions des exercices	12

I. Contexte

Durée : 1 h

Pré-Requis : Connaître les bases du HTML/CSS

Environnement de travail : PC

Contexte

Lorsque l'on parle de développement web, les deux standards qui nous viennent peut-être en premier à l'esprit sont les fameux HTML et CSS. Si on veut simplement définir ces deux langages, on peut dire que le HTML est un langage permettant d'organiser via des balises le contenu visuel d'un site internet, tandis que le CSS va permettre de créer et d'affecter des styles visuels aux éléments HTML. Ces deux langages informatiques vont donc de pair. On dit que ce sont des langages côté client.

On distingue le langage côté client du langage côté serveur. Ces deux différents termes identifient l'endroit où vont être exécutées les instructions des codes. Un troisième langage côté client qui va faire l'objet de notre attention est le langage de programmation JavaScript.

JavaScript est une technologie très utilisée dans le web. La plupart du temps, on l'utilise pour générer sur des sites internet des animations visuelles, ajouter de l'interactivité, stocker des cookies, etc. Mais on la retrouve aussi dans des environnements informatiques au niveau du back-end (gestion de serveurs, migrations des données, etc.) comme Node.js, Apache Couch DB, etc.

JavaScript est un langage de programmation très léger et universel. Il permet aussi d'aborder la programmation orientée objet. C'est un langage dit interprété. L'objectif de ce cours est de présenter JavaScript, ses spécificités et son histoire.

II. Présentation de JavaScript

A. Introduction

Commençons par présenter le langage de programmation JavaScript. Voici tout d'abord un exemple de script JavaScript, pour que vous puissiez voir rapidement à quoi peut ressembler sa syntaxe :

```
1 // Fonction pour calculer le périmètre d'un rectangle
2
3 function perimetreR (a, b) {
4     return 2 * a + 2 * b;
5 }
6
7 // Fonction pour calculer l'aire d'un rectangle
8
9 function aireR (a, b) {
10     return a * b
11 }
12
13 // Fonction pour comparer avec un if l'aire de 2 rectangles
14
15 function compareAire(rectangleOne, rectangleTwo) {
16     if (aireR(rectangleOne[0], rectangleOne[1]) < aireR(rectangleTwo[0], rectangleTwo[1])) {
17         alert("Le rectangle 1 est plus petit que le rectangle 2")
18     }
19     else if ((aireR(rectangleOne[0], rectangleOne[1]) > aireR(rectangleTwo[0],
20 rectangleTwo[1]))) {
21         alert("Le rectangle 1 est plus grand que le rectangle 2")
22     }
23     else {
```

```

23     alert("Les rectangles ont la même aire")
24 }
25 }
26
27 var rectangle1 = [10, 4]
28
29 var rectangle2 = [8, 5]
30
31 //Appel de la fonction compareAire en passant en paramètre les 2 tableaux contenant les
    valeurs des côtés des rectangles
32
33 compareAire(rectangle1, rectangle2);
34

```

C'est un exemple très simple de script, mais qui vous donne une idée de ce à quoi peut ressembler un script JavaScript. Maintenant, voyons quelques spécificités de JavaScript.

B. Langage côté client

JavaScript est, au même titre que HTML et CSS, un langage côté client. À la différence de ces derniers qui permettent de gérer le visuel d'une page informatique, JavaScript est un vrai langage de programmation. Que veut dire exactement l'expression « *côté client* » ?

Côté client ou côté serveur

Ces deux termes définissent comme nous l'avons dit le « *lieu* » où sont exécutées les instructions de code.

- Côté serveur : le navigateur web communique avec des serveurs distants (des ordinateurs qui stockent les données) par le biais d'un protocole que vous connaissez très certainement : le protocole HTTP. En réalité, lorsque l'on visite un site réalisé avec un langage côté serveur, le navigateur envoie par le biais de ce protocole des requêtes aux serveurs distants. Il peut ainsi, par exemple, accéder à des bases de données distantes (qui ne sont pas stockées sur son ordinateur). Le navigateur n'accède pas aux bases de données de manière directe, mais par le biais du serveur distant. Les scripts sont exécutés sur le serveur distant.
- Côté client : à la différence d'un langage côté serveur, un langage côté client est exécuté sur la machine du client, grâce au navigateur. Bien évidemment, le navigateur va communiquer avec un serveur distant pour recevoir les informations nécessaires, mais les instructions seront exécutées sur la machine de l'utilisateur et non sur le serveur. C'est le cas de JavaScript.

JavaScript est un langage côté client, mais c'est également le cas de HTML. Quel va être le lien entre ces deux langages ?

Pour le comprendre, il va nous falloir nous intéresser au concept de **Document Object Model** (DOM).

Document Object Model HTML

Comme vous l'avez peut-être noté, nous avons dit que JavaScript est un langage permettant d'aborder la programmation orientée objet. Mais qu'est-ce que c'est ?

La programmation orientée objet est une manière spécifique d'aborder la programmation. En fait, toutes les données sont représentées par le biais d'entités appelées « *objets* ». Les objets sont assimilables à des concepts de la réalité. Ils sont définis par leurs attributs (des valeurs qui les définissent) et des méthodes (des actions qu'ils peuvent utiliser).

Exemple

Prenons un exemple simple de la vie de tous les jours pour que vous puissiez mieux vous représenter ce concept.

Un ordinateur est un objet. Il a des attributs, par exemple :

```
1 marque = "Apple"
2 ram = 8
3 stockage = 512
```

Mais il a aussi des fonctions. Il peut par exemple s'allumer, s'éteindre, allumer certains pixels, etc. On peut comparer ces fonctions à des méthodes de l'objet ordinateur.

JavaScript est donc un langage qui va permettre de programmer en orienté objet. Mais quel est le rapport avec HTML ? C'est que les langages JavaScript et HTML vont pouvoir interagir grâce à cette logique.

Le Document Object Model est une convention qui permet de créer des représentations sous la forme d'objets, notamment dans des documents HTML. Il faut voir le Document Object Model comme un arbre d'objets représentant tous les éléments d'un document HTML, structuré conformément à sa hiérarchie.

Via le DOM HTML, JavaScript va pouvoir accéder aux différents objets du document HTML et interagir avec eux. JavaScript est donc un langage extrêmement lié à HTML. D'ailleurs, son intégration est possible directement dans des documents HTML.

C. Langage interprété vs langage compilé

Langage interprété et langage compilé

En programmation, on distingue langage interprété et langage compilé. En réalité, ces 2 termes sont liés à la manière dont va être traduit un code en langage machine (binaire) pour qu'il soit exécuté :

- Un langage interprété est un langage où le code source (code écrit dans ce langage) est lu puis exécuté dans l'ordre des lignes du script. Un interpréteur lit chaque instruction du code source, et l'exécute.
- Un langage compilé n'est pas traité de la même manière, il nécessite un compilateur. Un compilateur va lire toutes les instructions d'un code source pour créer à partir de l'ensemble un code objet, c'est-à-dire un fichier lisible par la machine. Ce fichier indépendant du code source pourra être stocké dans un fichier exécutable par la machine.

On comprend donc que la stratégie de traitement d'un code source n'est pas la même pour un langage interprété et pour un langage compilé.

JavaScript est classé parmi les langages interprétés. En effet, lorsque l'on exécute un script JavaScript, le code est exécuté de haut en bas, chaque instruction est traduite une par une en langage machine et exécutée.

Attention

Initialement, les moteurs JavaScript (programmes permettant d'exécuter du JavaScript) étaient des interpréteurs. Aujourd'hui, de nombreux navigateurs web utilisent des moteurs JavaScript faisant appel au principe de « *compilation à la volée* », permettant de réduire la perte de performance. Le principe de compilation à la volée consiste à compiler les instructions au moment de l'exécution, ce qui va faire gagner du temps lorsque par exemple, un code est appelé plusieurs fois (étant donné qu'il aura été compilé).

Différence entre interprétation et compilation

Sans trop s'attarder sur cette notion, voyons la différence de comportement entre l'interprétation et la compilation d'un script.

Prenons un exemple d'un script très simple :

```
1 console.log(1);
2
3 console.log(2);
4
5 consolelog(3);
6
7 console.log(4);
8
9 console.log(5);
```

Dans ce script, qui a pour objectif d'afficher dans la console les nombres 1 2 3 4 5, on peut constater une erreur à la ligne **consolelog(3)**. En effet, il manque un point. Donc, si on intègre ce script dans un code HTML, la console va afficher :

```
1 1
2 2
3 ReferenceError: Can't find variable: consolelog
```

Le moteur JavaScript lie chaque instruction et à chaque fois l'exécute, jusqu'à ce qu'il rencontre l'erreur. La console affiche donc les 2 premières valeurs et l'exécution s'arrête dès que l'erreur est rencontrée. On peut voir que le comportement n'aurait pas été le même si JavaScript avait été compilé. En effet, dans ce dernier cas, le compilateur aurait analysé le programme dans son ensemble. Étant donné qu'il rencontre une erreur, il n'aurait même pas compilé le script, et il n'y aurait eu aucune exécution.

Comment est exécuté JavaScript ?

Avec toutes ces informations que nous avons maintenant, on comprend que le JavaScript s'exécute comme suit :

- Le navigateur récupère et charge les codes d'une page web (HTML/CSS/JS).
- Le navigateur transforme toute la page en notre fameux Document Object Model, dans lequel il crée une représentation de chaque élément HTML.
- Le moteur JavaScript du navigateur convertit le code JS en un code intermédiaire entre le code source et la machine d'exécution, le bytecode.
- Les événements comme les clics de souris qui sont reliés à des blocs de code en JS déclenchent leur exécution. Le moteur JavaScript procède à l'interprétation du bytecode généré pour les blocs de code appelés et apporte les modifications visées sur le Document Object Model.
- Le nouveau Document Object Model est alors affiché.

On peut voir que, conformément à ce système, les navigateurs web peuvent utiliser des scripts JavaScript pour modifier le DOM d'une page web, et donc mettre à jour son affichage.

Remarque Modules

Il faut par ailleurs savoir que JavaScript est un langage modulaire. De nombreuses bibliothèques sont facilement utilisables, et permettent d'accroître et d'optimiser l'utilisation de JavaScript.

Définition Bibliothèque

En programmation, une bibliothèque (ou une librairie) est un ensemble de fonctions que l'on peut importer. Ces fonctions sont finalement des ensembles de codes déjà écrits et que l'on pourra appeler et donc utiliser sans avoir à les écrire nous-mêmes.

Voici quelques exemples de bibliothèques que l'on peut utiliser avec JavaScript :

- Ajout de fonctions permettant de manipuler le Document Object Model (DOM) :
 - JQuery¹
 - Umbrella JS²
- Ajout de fonctions de visualisation des données :
 - Chart.js³
 - Algolia places⁴
- Ajout de fonctions de formulaires :
 - wForms⁵
 - Validanguage⁶

Définition Frameworks

Un autre point intéressant est celui des frameworks.

Un framework est comme une grande bibliothèque, mais qui va constituer en quelques sortes un espace de travail pour les projets de code. Un framework va donc jouer un rôle dans l'organisation du projet et impacter toute sa conception.

Il existe plusieurs frameworks JavaScript :

- React Native : pour le développement d'applications JS pour Android et IOS
- Angular JS : pour les applications web plus complètes
- Bootstrap : pour la création de designs en responsive
- Node.js : pour créer des applications côté serveur

Voilà, vous en savez maintenant beaucoup plus sur JavaScript et ses spécificités.

D. Exercice : Quiz

[solution n°1 p.13]

Question 1

L'exécution d'un script JS intégré dans une page HTML se fait sur un serveur distant.

- ☐ Vrai
- ☐ Faux

Question 2

JavaScript est classé parmi les langages interprétés.

- ☐ Vrai
- ☐ Faux

Question 3

1 <https://jquery.com/>

2 <https://umbrellajs.com/>

3 <https://www.chartjs.org/>

4 <https://www.algolia.com/blog/product/introducing-algolia-places/>

5 <https://code.google.com/archive/p/wforms/>

6 <https://github.com/gcao/validanguage>

Comment s'appelle la convention permettant de représenter tous les éléments d'une page HTML par le biais d'objets ?

- ☐ Document Object Model
- ☐ Document Abstract Representation
- ☐ Document All References

Question 4

On peut intégrer un script JavaScript directement dans un document HTML.

- ☐ Vrai
- ☐ Faux

Question 5

Qu'est-ce qu'une bibliothèque ou librairie ?

- ☐ Un ensemble de fonctions déjà rédigées que l'on peut importer
- ☐ Une variable de type int
- ☐ Un espace de travail

III. Histoire et ECMAScript6

A. Histoire et ECMAScript6

Les débuts de JavaScript

Maintenant, parlons de l'histoire de JavaScript avant de nous intéresser à une version spécifique.

JavaScript a vu le jour en 1995 grâce à Brendan Eich, informaticien américain travaillant à ce moment-là pour l'entreprise Netscape Communication Corporation. JavaScript s'inspire du célèbre langage Java, mais l'objectif était de simplifier la syntaxe afin de le rendre plus accessible. Le premier nom de JavaScript est LiveScript, mais c'est très rapidement qu'il est baptisé « *JavaScript* ». L'objectif a été de concevoir un langage de programmation côté client qui soit intégrable dans le navigateur web de Netscape.

En 1996, Netscape intègre JavaScript dans le navigateur web de sa conception : NetScape Navigator 2.0 (cela ne nous parle plus trop aujourd'hui, mais il faut imaginer qu'à l'époque, c'était un navigateur très populaire). La concurrence émerge, Microsoft sort un langage de programmation, JScript, qu'il intègre dans sa version 3.0 d'Internet Explorer.

La standardisation de JavaScript est confiée à l'organisation européenne ECMA international (European association for standardizing information and communication systems). C'est en juin 1997 que naît le premier standard de JavaScript appelé ECMAScript, qui est la première édition du standard appelé ECMA-262. Puis, ce standard sera transmis en 1998 à l'Organisation Internationale de Normalisation, qui publiera un standard international.

En 2012, tous les navigateurs web prennent en charge la version 5.1 d'ECMAScript. En 2015, la sixième version d'ECMAScript est publiée, ECMAScript 2015 (ou ECMAScript 6 voir ES6).

Intérêt de JavaScript

Il faut bien s'imaginer que dans les années 1990, la plupart des pages web étaient statiques. Elles ne comportaient pas vraiment d'animations, ce qui rendait le web moins interactif. JavaScript a vu le jour comme un langage de programmation côté client, permettant d'introduire de l'interactivité et du dynamisme dans la mise en page du web.

JavaScript a donc marqué une véritable révolution dans le monde du développement web. D'ailleurs, il est toujours très utilisé lors de la mise en place d'animations et de solutions d'interactivité. JavaScript présente de nombreux avantages :

- Sa simplicité d'utilisation, JavaScript a été conçu pour être accessible,
- Il peut exécuter plusieurs instructions simultanément,
- Il permet de concentrer des tâches sur l'environnement de l'utilisateur, libérant ainsi les serveurs des tâches qui seront alors exécutées du côté client,
- Il permet de concevoir des sites internet intuitifs et plus accessibles,
- Il peut être exécuté à l'aide de n'importe quel navigateur web,
- Il contient de nombreuses bibliothèques facilement importables et fonctionnelles,
- Il peut être utilisé par le biais de frameworks,
- Il sert au fonctionnement d'API comme Node.js.

ECMAScript 6

Précédemment, nous avons abordé ECMAScript, et vu qu'il s'agit du standard de JavaScript. Il contient entre autres la définition de :

- La syntaxe de JavaScript,
- Les types des variables,
- Les fonctions et les objets dits natifs (de base),
- Les mécanismes assurant le traitement des erreurs,
- Le fonctionnement de l'héritage,
- L'objet dit global, c'est-à-dire l'objet qui va contenir toutes les propriétés, les fonctions et les variables qui seront accessibles à n'importe quel endroit d'un code. Dans un navigateur web, cet objet est l'objet **window**.

ECMAScript est donc un standard qui définit complètement le langage JavaScript et de tout ce qui le compose. Mais qu'apporte en 2015 ECMAScript 6 ?

Précédemment, nous avons abordé la Programmation Orientée Objet (POO). ECMAScript 6 apporte des commandes qui font de JavaScript un langage assimilable visuellement au C++ ou à Java, en termes de syntaxe. Le système de POO devient plus accessible en JavaScript, même si les modifications ne portent globalement que sur l'aspect visuel de la syntaxe. Effectivement, on ne peut pas dire que JavaScript est « *autant* » inscrit dans la logique orientée objet que Java ou C++. Mais ces fonctionnalités s'en rapprochent, au moins visuellement, et permettent d'aborder la POO en JavaScript.

D'autres modifications et ajouts sont opérés, par exemple :

- La possibilité d'importer des modules (bibliothèques, etc.),
- L'ajout de structures de données (tableaux associatifs, etc.),
- Les promesses concernant les systèmes asynchrones.

ECMAScript 6 constitue ainsi une mise à jour considérable de JavaScript, et un événement dans son histoire.

B. Exercice : Quiz

[solution n°2 p.14]

Question 1

Quel est le premier nom de JavaScript ?

- ☐ JScript
- ☐ LiveScript
- ☐ PHP

Question 2

Quel organisme s'est occupé de la standardisation de JavaScript ?

- ☐ L'ECMA international
- ☐ La DREETS
- ☐ Un autre organisme

Question 3

ECMAScript 6 a apporté de nouvelles structures de données.

- ☐ Vrai
- ☐ Faux

Question 4

ECMAScript définit notamment :

- ☐ Les bibliothèques externes
- ☐ La syntaxe de JavaScript
- ☐ La syntaxe de PHP

Question 5

JavaScript n'a jamais eu de concurrence.

- ☐ Vrai
- ☐ Faux

IV. Essentiel

JavaScript est un langage interprété et permettant d'aborder la logique orientée objet. JavaScript va de pair avec le système HTML/CSS, d'ailleurs, des instructions en JavaScript peuvent être directement intégrées dans un document HTML.

Le système de Document Object Model permet de générer une représentation organisée des éléments présents dans une page web. Le DOM va permettre à JavaScript d'accéder aux éléments HTML et de les modifier.

Tous les navigateurs web disposent d'un moteur JavaScript leur permettant d'exécuter du JavaScript. Un navigateur récupère et charge une page HTML, avant de la transformer en Document Object Model. Le moteur JS va convertir les instructions JS en bytecode. Puis, lorsqu'un événement déclenche un appel de code, le moteur JS interprète le bytecode généré correspondant et modifie le DOM en conséquence. Le DOM est alors affiché.

JavaScript possède de nombreux atouts, parmi lesquels figurent sa simplicité d'utilisation, sa possibilité d'exécution sur n'importe quel système, les bibliothèques importables et l'utilisation possible de frameworks. Il permet de rajouter de l'interactivité et des animations sur les sites web.

JavaScript est initialement créé en 1995 dans l'entreprise Netscape Communication Corporation, en s'inspirant de Java. Le premier nom qui lui est donné est LiveScript, mais il est très rapidement renommé JavaScript. JavaScript est standardisé par l'ECMA. ECMAScript est le standard de JavaScript. Une version d'ECMAScript qui a marqué un tournant dans l'histoire de JavaScript est la version 6, qui apporte des changements considérables dans le langage.

V. Auto-évaluation

A. Exercice

Vous êtes développeur et vous vous intéressez aux différentes technologies de développement web. Un ami vous interroge sur la technologie JavaScript.

Question 1

[solution n°3 p.15]

Expliquez-lui en 5 lignes ce qu'est JavaScript.

Question 2

[solution n°4 p.15]

Résumez- lui, en moins de 10 lignes, l'histoire de JavaScript.

B. Test

Exercice 1 : Quiz

[solution n°5 p.15]

Question 1

Un langage interprété est lu en entier avant d'être traduit en langage machine.

- ☐ Vrai
- ☐ Faux

Question 2

Si un script JavaScript contient une erreur :

- ☐ Le script s'exécutera jusqu'à l'erreur
- ☐ Le script ne s'exécutera pas
- ☐ Aucune erreur ne sera signalée

Question 3

Un script d'un langage côté client est :

- ☐ Exécuté sur un serveur distant
- ☐ Exécuté sur la machine de l'utilisateur
- ☐ Exécuté sur un serveur proche

Question 4

Qu'est-ce qu'un framework ?

- ☐ Une bibliothèque prédéfinie en JavaScript
- ☐ Une version de Jscript
- ☐ Une sorte de grande bibliothèque qui définit l'organisation même du projet de code

Question 5

L'objet global dans un navigateur web est :

- ☐ global
- ☐ window
- ☐ root


Solutions des exercices

Exercice p. 7 Solution n°1**Question 1**

L'exécution d'un script JS intégré dans une page HTML se fait sur un serveur distant.

☐ Vrai

☒ Faux


 L'exécution d'un script JS qui est intégré dans une page HTML se fait sur la machine du client et non sur un serveur distant.

Question 2

JavaScript est classé parmi les langages interprétés.

☒ Vrai

☐ Faux

 JavaScript est classé comme langage interprété. Attention toutefois, aujourd'hui, les moteurs JavaScript ne sont pas forcément de simples interpréteurs, mais peuvent compiler à la volée. Le système reste toutefois différent qu'avec un langage compilé.


Question 3

Comment s'appelle la convention permettant de représenter tous les éléments d'une page HTML par le biais d'objets ?

☒ Document Object Model

☐ Document Abstract Representation

☐ Document All References


 C'est le Document Object Model (DOM). C'est une convention représentant tous les éléments d'une page html sous la forme d'un arbre d'objets.

Question 4

On peut intégrer un script JavaScript directement dans un document HTML.

☒ Vrai

☐ Faux

 En effet, il est tout à fait possible d'intégrer un script JavaScript dans un document HTML, directement ou via un lien. C'est certainement la manière la plus courante d'utiliser du JavaScript.


Question 5

Qu'est-ce qu'une bibliothèque ou librairie ?

☒ Un ensemble de fonctions déjà rédigées que l'on peut importer

☐ Une variable de type int

☐ Un espace de travail


-  Une bibliothèque ou librairie est un ensemble de fonctions (des blocs de codes) que l'on peut importer dans un code. On pourra directement appeler ces fonctions dans le script, sans avoir à les écrire soi-même.

Exercice p. 9 Solution n°2

Question 1

Quel est le premier nom de JavaScript ?


- ☐ JScript
- ☒ LiveScript
- ☐ PHP

 Le premier nom de JavaScript est LiveScript, mais il est très rapidement renommé JavaScript.

Question 2

Quel organisme s'est occupé de la standardisation de JavaScript ?

- ☒ L'ECMA international
- ☐ La DREETS
- ☐ Un autre organisme

 L'ECMA (European association for standardizing information and communication systems) a standardisé JavaScript. Le standard de JavaScript est ECMAScript.

Question 3

ECMAScript 6 a apporté de nouvelles structures de données.


- ☒ Vrai
- ☐ Faux

 ECMAScript 6 a apporté de nouvelles structures de données comme les tableaux associatifs.

Question 4

ECMAScript définit notamment :

- ☐ Les bibliothèques externes
- ☒ La syntaxe de JavaScript
- ☐ La syntaxe de PHP

 ECMAScript définit notamment la syntaxe de JavaScript, mais aussi les types de variables, les mécanismes de traitement des erreurs, etc.

Question 5

JavaScript n'a jamais eu de concurrence.

- ☐ Vrai
- ☒ Faux

- Q Une concurrence à JavaScript a été JScript développé par Microsoft et intégré à Microsoft Explorer 3.0, tandis que JavaScript était intégré à NetScape Navigator 2.0.

p. 11 Solution n°3

JavaScript est un langage de programmation côté client qui est classé parmi les langages interprétés. Il fonctionne de pair avec le système HTML/CSS. JavaScript est un langage de programmation accessible, qui est beaucoup utilisé pour ajouter de l'interactivité et des animations sur les sites web. JavaScript permet aussi d'aborder la programmation sous l'angle de la programmation orientée objet. Il existe de nombreuses bibliothèques JavaScript mais aussi des frameworks adaptés.

p. 11 Solution n°4

La première version de JavaScript a été publiée en 1995. Elle a été développée par l'informaticien Brendan Eich dans l'entreprise américaine Netscape. JavaScript est initialement baptisé LiveScript avant de recevoir son nouveau nom. JavaScript a été standardisé par l'ECMA et en 1997 est publié le standard ECMAScript. C'est en 1998 que l'organisation internationale de normalisation publiera le standard international de JavaScript. En 2012, tous les navigateurs web qui existent disposent d'un moteur JavaScript, prenant en charge ECMAScript 5.1. Une version majeure d'ECMAScript est publiée en 2015 : ECMAScript 6.

Exercice p. 11 Solution n°5

Question 1

Un langage interprété est lu en entier avant d'être traduit en langage machine.

- ☐ Vrai
☒ Faux

- Q Un interpréteur lie chaque instruction d'un code source et l'exécute. À la différence d'un script dans un langage interprété, un script dans un langage compilé est lu en entier pour être traduit dans un langage machine, le code objet. Ce dernier est stocké dans un fichier exécutable.

Question 2

Si un script JavaScript contient une erreur :

- ☒ Le script s'exécutera jusqu'à l'erreur
☐ Le script ne s'exécutera pas
☐ Aucune erreur ne sera signalée

- Q Si un script JavaScript contient une erreur, alors il sera exécuté jusqu'à l'erreur.

Question 3

Un script d'un langage côté client est :

- ☐ Exécuté sur un serveur distant
☒ Exécuté sur la machine de l'utilisateur
☐ Exécuté sur un serveur proche

- Q Un langage côté client est exécuté sur la machine de l'utilisateur, grâce au navigateur. Il faut à un navigateur web un moteur JavaScript pour lire et exécuter du JavaScript.

Question 4

Qu'est-ce qu'un framework ?

- ☐ Une bibliothèque prédéfinie en JavaScript
 - ☐ Une version de Jscript
 - ☒ Une sorte de grande bibliothèque qui définit l'organisation même du projet de code
- Q Un framework est en quelques sortes une grande bibliothèque ou un ensemble de bibliothèques qui définit l'organisation même du projet de code. C'est un peu comme un plan qui permet de construire un bâtiment.

Question 5

L'objet global dans un navigateur web est :

- ☐ global
 - ☒ window
 - ☐ root
- Q L'objet global défini par ECMAScript est **window**. Il contient toutes les propriétés ainsi que les variables et fonctions accessibles dans la portée globale, c'est-à-dire dans tout le code JS.