JavaScript 😶

writecode

Définition

Le JavaScript est un langage de programmation interprété

JavaScript peut être intégré directement au sein des pages Web, **pour y être exécuté sur le poste client**. C'est le navigateur qui prend en charge l'exécution de ces programmes appelés "scripts"

JavaScript sert à contrôler les données saisies dans les formulaires HTML, interagir avec le document HTML via l'interface **DOM (Document Object Model)** et réaliser des services dynamiques, strictement cosmétiques ou à des fins ergonomiques.

Attention

Se prononce JavaScript et surtout pas Java !

Java est un autre langage de programmation qui n’a rien a voir avec le JavaScript.

Création

JavaScript a été créé en 1995 par Brendan Eich

Il a été standardisé pour le nom "ECMAScript" en juin 1997. JavaScript n'est depuis qu'une implémentation d'ECMAScript, celle mise en place par la fondation Mozilla.

L'implémentation d'ECMAScript par Microsoft se nomme Script, tandis que celle d'Adobe se nomme ActionScript.

Langage interprété

Un langage interprété est un langage qui n'a pas besoin d'être converti (ou compilé) pour être compris par les logiciels qui l'utilisent.

Il est basé sur le fonctionnement des langages de programmation et sur l'algorithmique

Bases JavaScript 

writecode

Variables

Une variable est un objet repéré par un nom, pouvant contenir des données, qui pourront être modifiées lors de l'exécution du programme.

On distinguera trois type des variables :

- Globales, disponible à tout moment
- Locales, disponibles uniquement dans un contexte donné
- Bloc, disponible uniquement dans la partie de code dans laquelle elles ont été créées

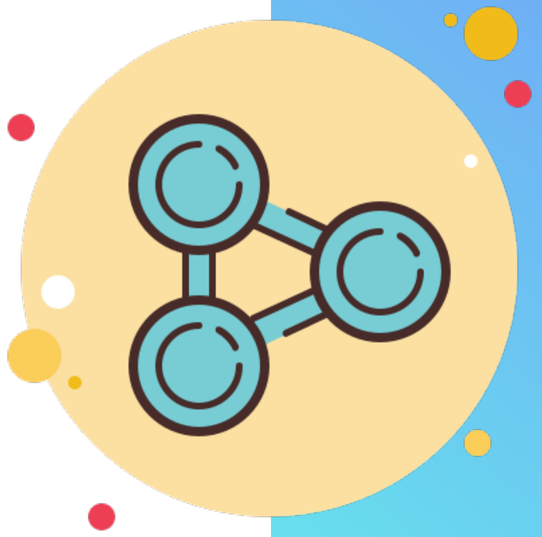
Type de données

var a = 15;	Entier (integer)
var a = 15.12;	Décimal (float)
var a = "15";	Chaîne de caractère (string)
var a = false;	Booléen (boolean)
var a = [15, "14", 32];	Tableau (array)

Opérateurs

Ils existent différents type d'opérateurs

On distinguera les opérateurs de calcul, de comparaison et de logiques



Opérateur de calcul

Addition : +

Soustraction : -

Multiplication : *

Division : /

Modulo : %

Opérateur de comparaison

Supérieur à : >

Supérieur ou égal à : >=

Inférieur à : <

Inférieur ou égal à : <=

Egal à : ==

Strictement égal à : ===

Différent de : !=

Strictement différent de : !==

writecode

Opérateur logiques

Pas : !

Et : &&

Ou : ||

Conditions

Une condition est une instruction qui permet de tester si une affirmation est vraie ou non, ce qui permet notamment de donner de l'interactivité aux scripts

Par exemple, je souhaite savoir si un élément de formulaire à correctement été rempli :

- Si oui, je valide le formulaire
- Si non, j'affiche un message d'erreur

```
if (condition) {
  // La condition est vraie, j'exécute le code présent ici
}
else {
  // La condition est fausse, j'exécute le code présent ici
}
```

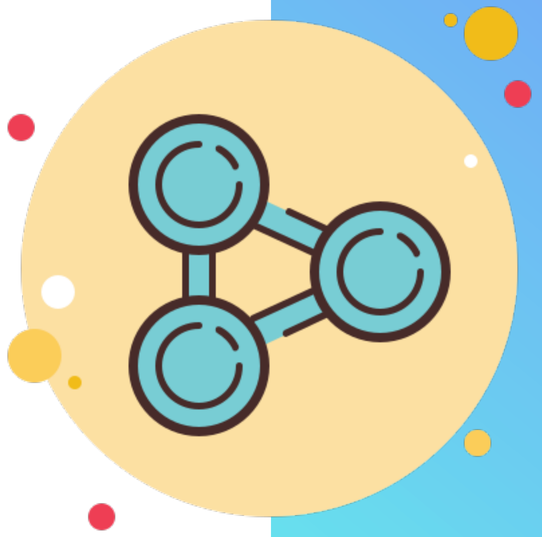
Boucles

Dans certains cas, il est nécessaire de répéter le même code plusieurs fois, c'est alors qu'interviennent les boucles

On distinguera deux types de boucles :

- Lorsqu'on connaît à l'avance le nombre de fois que la boucle doit "tourner"

- Lorsqu'on ne connaît pas à l'avance le nombre de fois que la boucle doit "tourner"



Boucle for()

La boucle for() à besoin de trois arguments pour fonctionner

```
1 for(var essai = 1; essai <= 6; essai++) {
2   // Code à exécuter
3 }
```

writecode

Boucle while()

La boucle while() demande une condition. Tant que c'est condition est vraie, le code à l'intérieur de la boucle est exécuter

```
1 while(mot de passe faux) {
2   // Code à exécuter
3 }
```

Boucle do...while()

La boucle do...while() exécute d'abord le code contenu dans le "do" et ensuite vérifie si la condition est vraie. Si celle-ci s'avère être fausse, le code contenu dans "do" est de nouveau exécuté et ensuite de suite...

```
1 do {
2   // Code à exécuter
3 } while(mot de passe faux)
```