

420JJAJQ

Programmation mobile

Javascript



CÉGEP DE JONQUIÈRE

Département d'Informatique

MT. Moutacalli

Définition



JavaScript est un langage de programmation de **scripts** principalement employé dans les pages web interactives côté client mais aussi, à partir de 2009, il est aussi utilisé côté serveur (avec l'utilisation, par exemple, de Node.JS ou projet CommonJS).

Histoire



Brendan Eich a initialement développé LiveScript, un langage de script côté serveur pour Netscape.

En 1995, Netscape développe la version orientée client de LiveScript et l'appelle **Javascript** pour profiter de la popularité de la machine virtuelle Java de son partenaire Sun Microsystems.

Javascript != Java

Fonctionnalités

Javascript permet de modifier le comportement des pages web de manière dynamique.

Possibilité de modifier le HTML : chemin source d'images, modification de texte... et le CSS : afficher / masquer des objets, mise à jour de contenu...

Manipulation facile de Strings, d'Arrays, de dates, d'Events, de fonctions, d'opérateurs...

Fonctionnalités



Contrôle des informations saisies dans des formulaires HTML.

Interagir avec le document HTML via l'interface **DOM** (Document Object Model) fournie par le navigateur.

Utilisé au sein de l'architecture Ajax (Asynchronous Javascript And XML), qui permet un dialogue entre le client et le serveur en arrière plan sans perturber l'utilisateur d'une page web.

Stockage d'informations structurées sous le format JSON (JavaScript Object Notation).

Fonctionnalités

Si HTML est utilisé pour le contenu et CSS pour la forme, Javascript est utilisé pour l' **interactivité**.

- Changer le contenu et les attributs HTML : texte, image...

What Can JavaScript Do?

JavaScript can change HTML attributes.

In this case JavaScript changes the src (source) attribute of an image.



Turn on the light

Turn off the light

What Can JavaScript Do?

JavaScript can change HTML attributes.

In this case JavaScript changes the src (source) attribute of an image.



Turn on the light

Turn off the light

Fonctionnalités

Si HTML est utilisé pour le contenu et CSS pour la forme, Javascript est utilisé pour l' **interactivité**.

- Changer le style CSS : couleur de fond de texte...
- Vérifier les informations saisies dans un formulaire
- Stocker des variables saisies par l'utilisateur
- Accéder à une foule d'informations sur le navigateur et la page Web exécutée

Fonctionnalités

Si HTML est utilisé pour le contenu et CSS pour la forme, Javascript est utilisé pour l' **interactivité**.

- Créer des jeux : Ex Cookie et Hex GL



Programmation en Javascript

Il existe deux méthodes pour insérer du code Javascript dans un fichier HTML :

- 1 Directement dans le fichier HTML (méthode déconseillée)
Ex: `<script>console.log("Hello world!");</script>`
- 2 Créer un script (fichier texte avec l'extension .js) puis dans la page HTML indiquer le chemin de ce script grâce à l'attribut **src**
Ex: `<script src="script.js"></script>`

Afficher Hello World dans la console avec les deux méthodes.

Méthode 1

Créer un fichier hellom1.html contenant le code javascript dans Head ou le Body mais, généralement, on l'insère dans le Head:

```
<html>
  <head>
    <script>console.log("Hello world!");</script>
  </head>
  <body></body>
</html>
```

Ou bien :

```
<html>
  <head> </head>
  <body>
    <script>console.log("Hello world!");</script>
  </body>
</html>
```

Méthode 2

Créer un script hello.js contenant :

```
console.log("Hello world!");
```

Créer un fichier hellom2.html contenant :

```
<html>  
  <head> </head>  
  <body>  
    <script src="hello.js"></script>  
  </body>  
</html>
```

Il vaut mieux organiser les scripts dans un dossier "scripts" et dans ce cas le code sera:

```
<script src=".\\scripts\\hello.js"></script>
```

Commentaires

- // pour une ligne
- /* De cette ligne
- Jusqu'à cette ligne */

Types

- Undefined
- Number
- Boolean
- String
- Date
- Array
- Function
- Object
- RegExp
- Null

Variables

En Javascript, une variable peut être déclarée de deux façons:

- Explicite (recommandée) en utilisant:

var nomvariable;

Elle prendra le type Undefined jusqu'à ce qu'on lui affecte une valeur et elle prendra le type de cette valeur.

Déclaration et initialisation: `var a = 4;`

- Implicite:

Ex: `a=7;`

Dans ce cas, la variable est **globale** (pas de `var`) et elle prend directement le type de la valeur.

Attention: Javascript est sensible à la casse:

`variabletest` != `Variabletest`

Test et Conversion de type

- **parseInt()** et **parseFloat()** permettent de convertir une chaîne de caractères en nombre.
- **Number()** transforme un objet en nombre à la manière de **parseFloat()** pour les chaînes de caractères.
- **string()** transforme un objet en chaîne de caractères.
- **isFinite()** permet de tester si la variable est bien un nombre fini
- **isNaN()** teste si le paramètre n'est pas un nombre (NaN = Not a Number)

Opérateur conditionnel

Comme c#

```
if (condition) {  
    instructions  
} else if (condition) {  
    instructions  
} else {  
    instructions  
}
```


Instruction switch

```
switch (choix) {  
  case 1:  
    instructions  
  break;  
  case 2:  
    instructions  
  break;  
  ...  
  case n:  
    instructions  
  break;  
  default:  
    instructions  
}
```

Les boucles

Comme Java:

```
for (var i=0; i<n;i++) {  
    instructions  
}
```

```
var compteur = 1;  
do {  
    instructions  
    compteur++;  
} while (compteur<10);
```

```
var compteur = 1;  
while (compteur<10) {  
    instructions  
    compteur++;  
}
```

Les Fonctions

- En Javascript, les fonctions sont un type de données. Elles peuvent être passées en argument à d'autres fonctions et être mises dans des variables.
- Fonction nommée:
`function ma_fonction(arg1, arg2){ ... }`
- Fonction anonyme:
`function(arg1, arg2) { ... }`

Les Fonctions

```
function times(n,action) {  
    for(var i=0; i<n;i++) {  
        action(i);  
    }  
}  
  
times(10,function(iteration) ) {  
    console.log("Hello world!" + iteration);  
}
```

Les Tableaux

Déclaration d'un tableau de 4 éléments:

```
var tableau1 = new Array(4);  
tableau1[0]="Alice" ;
```

Déclaration d'un tableau dont le nombre d'éléments est inconnu au moment de la déclaration :

```
var tableau2 = new Array();
```

Déclaration avec initialisation

```
var tableau3 = ["David","Jean","Audrey"];
```

Ou:

```
var tableau4 = new Array("David","Jean","Audrey");
```

Tableaux multidimensionnels

```
var tableau1 = new Array(4);  
for (i=0;i<tableau1.length;i++) {  
    tableau1[i] = new Array(2) ;  
}
```