

Technologies du web

5 - JavaScript

Previously on HMIN115M

HTML : structuration de la page

CSS : présentation de la page

PHP : génération de la page

...

JavaScript : interactions dans la page

Coté client

Le JavaScript est **interprété côté client**, dans le navigateur, en local (comme le CSS)

Il est **généré côté serveur** et est intégré à la page HTML via un lien (comme CSS) ou directement dans la sa balise `<script></script>` (comme CSS avec sa balise `<style>`)

Page dynamique

- JavaScript permet de **rendre dynamique le contenu d'une page web**, pas besoin de changer de page ou de recharger la page courante pour voir les modifications
- JavaScript permet de **concevoir des applications web**, des applications riches réalisant des traitements de données en local, avant d'envoyer les résultats à un serveur

Généralités

JavaScript n'a **aucun lien avec Java** (mais genre vraiment aucun)

Version 1 développée par Brendan Eich et Netscape en 1996

Évolue selon le standard **ECMAScript**

Langage orienté objet

Versions

2009 : ECMAScript 5^e édition (ES5)

2015 : ES6

2016 : ES7

...

2019 : ES10

Le standard évolue maintenant d'**une version par an** pour éviter les changements trop importants occasionnés à la sortie d'ES6 presque 6 ans après ES5

Usages notables

- Valider les entrées d'un formulaire et afficher les erreurs et l'aide adéquate
- Envoyer et recevoir des données à un serveur sans rechargement de page
 - Suggestions de recherche
 - Flux d'informations type Twitter ou Facebook
- Animer des éléments de la page
 - Jeux
 - Carousel

Inclure du JavaScript

index.html

```
<html>
  <head>
    <script type="text/javascript" src="javascript.js"></script>
  </head>
</html>
```

javascript.js

```
console.log('coucou');
```


Ouvrir la console JavaScript du navigateur

- Clic droit → Examiner / Inspecter → (Onglet) Console
- F12 → (Onglet) Console

Afficher un message

Afficher un message, le contenu d'un variable, etc. dans la console

```
console.log( ' coucou' );
```

Afficher une boîte de dialogue

```
alert( ' coucou' );
```

Commentaires

```
// commentaire sur une ligne
```

```
/* bloc de  
   commentaire  
   multiligne  
*/
```

Syntaxe

```
// une instruction par ligne
```

```
// chaque ligne se termine par un ;
```

```
var pouet;           // déclaration d'une variable indéfinie
```

```
var nom = "Javascript"; // déclaration d'une variable
```

```
pouet = 0 + 0;        // addition mathématique
```

```
console.log(pouet);    // exécution d'une fonction
```

Variables

```
// variable avec une portée limitée à la fonction
var pouet;
var nom = 'Javascript';

function test(){
    var maVar = 'valeur';
}
test();
console.log(maVar);
// Retourne 'undefined'
console.log(nom);
// Retourne 'Javascript'
```

Variables ES6

Deux nouveaux types ont été introduit en 2015 : **let** et **const**

- **let** permet de créer une variable à la portée limitée au bloc courant. Un bloc est représenté entre des { }, comme par exemple le contenu d'un if
- **const** permet de créer une variable dont la valeur ne peut être modifiée et dont la portée est également limitée au bloc.

Les types de variables

Chaîne de caractères, nombre, fonction, objet, booléen

```
var nom = 'Javascript';  
typeof(nom);  
  
// affiche "string"
```

Manipulations mathématiques

Voir l'objet Math pour les opérations mathématiques

https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets_globaux/Math

```
// console.log(Math);
```


Chaine de caractères

```
var chaine = 'chaine de caractères';
```

```
var debut = 'Je suis une ';
```

```
console.log(debut + chaine);           // concaténation de chaines
```

```
console.log(chaine.length);           // longueur de la chaine
```

Créer une fonction

```
function ma_fonction_qui_dit_bonjour(prenom) {  
    console.log('Bonjour ' + prenom);  
    return true;  
}  
  
ma_fonction_qui_dit_bonjour('Bobby');
```

Fonction et variable

```
var resultat = ma_fonction();    // exécute la fonction
```

VS

```
var resultat = ma_fonction;    // référence la fonction
```

//! Know the difference !/\

Conditions

```
function abs(x) {  
    if (x < 0) {  
        x = -x;  
        return x;  
    } else {  
        return x; }  
}
```

```
console.log(abs(2), abs(-3));
```

Boucle

```
var liste = [ 'Firefox', 'Safari', 'Chromium' ];  
  
for (var i = 0; i < liste.length; i++) {  
    console.log(liste[i]);  
}
```

Booléens

true et **false**

Implicitement les variables ayant pour valeur 0, null ou undefined sont considérées comme **false**

L'opérateur **!!** permet de convertir une variable en booléen

Données structurées

Ce sont des **objets**

Listes ordonnées : `[]`

```
ex: var liste = [ 'Firefox', 'Safari', 'Chromium' ];
```

Listes clé-valeur : `{}`

```
ex: var ages = {  
    bobby: 23,  
    thierry: 42  
}
```

Exemple d'objet

```
var info = {  
  nom : "perret",  
  prenom : "pierre",  
  age : 1000,  
  telephones : ["0648151623", "0836656565"],  
  adresse : {  
    rue : "rue de la gerbe",  
    code : 69002,  
    ville : "lyon"  
  }  
};
```

```
console.log(info.adresse.ville);
```


Une fonction dans un objet

```
var moyenne = {  
  notes: [ 4, 8, 15, 16, 23 ],  
  get: function () {  
    var somme = this.notes.reduce(function (a, b) {  
      return a + b  
    });  
    return somme / this.notes.length;  
  }  
};
```

```
moyenne.notes.push(42);           // ajoute 42 à la liste  
console.log( moyenne.get() );     // affiche la moyenne
```

JSON

```
// convertit un objet en JSON  
var chaine = JSON.stringify(moyenne);
```

et inversement

```
// on retrouve notre objet  
var obj = JSON.parse(chaine);
```

Et pour manipuler la page web

Les variables globales :

document : le DOM

ex : `console.log(document.URL);`

window : la fenêtre

Exécuter le JavaScript

Le code JavaScript qui manipule des éléments du DOM doit attendre que la page soit complètement chargée pour être lancé

```
document.addEventListener(  
    'DOMContentLoaded',  
    function () {  
        console.log('Kikoo');  
    }  
);
```

Exemple

```
// sélectionne par id
var menu = document.getElementById('menu');

// sélectionne le premier élément du sélecteur
var elem = document.querySelector('.ma-classe');

// sélectionne tous les éléments du selecteur
var elems = document.querySelectorAll('.ma-classe');
for (var i = 0; i < elems.length; i++) {
    console.log(elems);
}
```

Écouter les évènements

Avec JavaScript on peut “écouter” les évènements provenant de la page, par exemple pour exécuter une fonction lors d'un clic sur un élément.

Il y a plein d'évènements possibles :

- click
- hover
- focus
- change
- submit
- <https://developer.mozilla.org/en-US/docs/Web/Events>

Exemple

```
var btn = document.querySelectorAll('.close');  
btn[0].addEventListener('click', function () {  
    alert('Voulez-vous vraiment fermer ?');  
});
```

Classes et attributs

```
var div = document.querySelector('#menu');
```

```
// accès aux classes  
console.log(div.id);  
var classes = div.classList;  
console.log(classes);  
classes.add('green');  
classes.remove('blue');  
classes.toggle('hidden');
```

```
// accès aux attributs  
div.setAttribute('title', 'main menu');  
console.log(div.getAttribute('title'));
```


Contenu des balises

```
var paragraphe = document.querySelector('p');  
// get HTML  
var html = paragraphe.innerHTML;  
console.log('Contenu HTML : ' + html);  
// set HTML  
paragraphe.innerHTML = 'nouveau texte en <i>italique</i>';  
// get text content (without tags)  
var texte = paragraphe.textContent;  
console.log('Contenu texte : ' + texte);
```

Exemple de formulaire

```
<html>
  <head>
    <script type="text/javascript" src="monfichier.js"></script>
  </head>
  <body>
    <form>
      <label for="nom">Nom</label>
      <input id="nom" name="nom" type="text"/><br/>
      <input type="button" id="envoyer" value="Envoyer"/>
    </form>
    <div id="info"></div>
  </body>
</html>
```

Exemple de validation

```
document.addEventListener('DOMContentLoaded', function () {  
    var verif = document.getElementById('envoyer');  
    verif.addEventListener('click', function () {  
        var name =  
            document.getElementById('nom').value || 'Nom vide';  
  
        document.getElementById('info').innerHTML =  
            '<b>' + name + '</b>';  
    });  
});
```