

Projet de développement Web

QU'EST CE QUE LE JAVASCRIPT

est un langage de programmation orienté objet, faiblement typé qui peut s'exécuter côté client, côté serveur et même être utilisé dans des applications desktop



QU'EST CE QUE LE JAVASCRIPT

est un langage de programmation interprété.

- Moteur javascript V8 est utilisé sur Chrome et Node.js
- Moteur javascript SpiderMonkey utilisé sur Firefox

QU'EST CE QUE LE JAVASCRIPT

Quelques librairies/frameworks connus en Javascript

Côté client

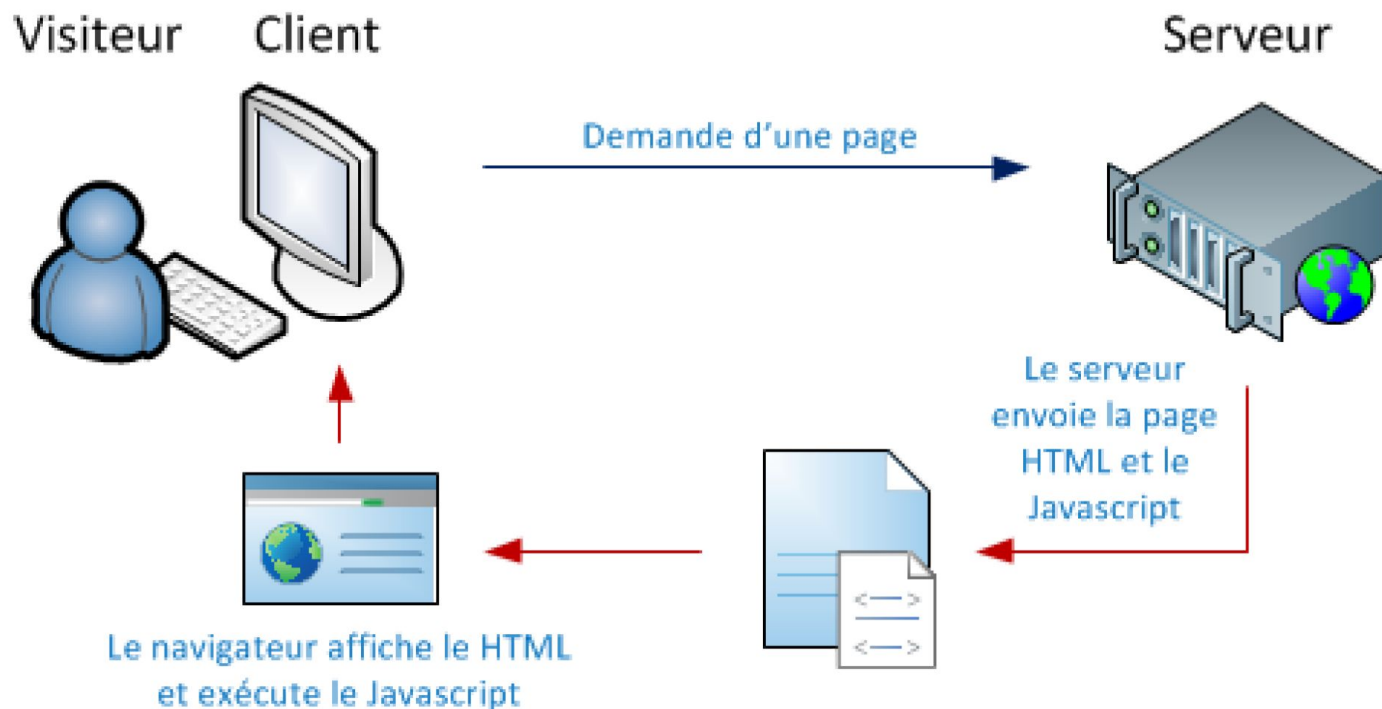
- JQuery
- AngularJS
- ReactJS
- VueJs
- ...

Côté serveur

- Express.js
- Sails.js
- Meteor.js
- ...

Javascript Côté client

Comment cela fonctionne-t-il



Comment insérer du javascript dans une page web

A l'aide de la balise script vous pouvez directement insérer du code javascript

```
<script type="text/javascript">
  function test(){
    console.log('Projet de développement web');
  }
  test();
</script>
```

Où en incluant un fichier externe hébergé sur le serveur ou sur un autre serveur

```
<script src="fichierJs.js"></script>
<script src="http://www.liepsm.be/fichierJs.js"></script>
```

Ou insérer du javascript dans une page web

```
1 <html>
2 <head>
3   <meta charset="utf-8"/>
4   <title>Projet de développement web</title>
5 </head>
6 <body>
7   <script src="fichierJs.js"></script>
8   <script type="text/javascript">
9     function test(){
10       console.log('Projet de développement web');
11     }
12     test();
13   </script>
14 </body>
15 </html>
16
```



Vous pouvez insérer votre code javascript entre les balises **head** et **body**, mais pour des raisons de performance, il est préférable de le placer juste avant la fermeture de la balise **body**

Déclaration des variables

```
function fn() {  
  const foo = "bar"  
  if (true) {  
    const foo // SyntaxError, la variable a besoin d'être assignée  
    const foo = "qux"  
    foo = "norf" // SyntaxError, la variable ne peut pas être réassignée  
    console.log(foo)  
    // "qux", la variable appartient au scope de son bloc (le "if")  
  }  
  console.log(foo)  
  // "bar", la variable appartient au scope de la fonction "fn"  
}
```

const

Variable scopée dans le bloc et qui ne peut être réassignée à nouveau, la constante est faite au niveau référence.

Déclaration des variables

```
1 function fn() {  
2   let foo = "bar";  
3   var foo2 = "bar";  
4   if (true) {  
5     let foo; // pas d'erreur, foo === undefined  
6     var foo2;  
7     // Attention, les déclarations "var" ne sont pas scopées au niveau bloc  
8     // foo2 est en réalité écrasé !  
9     foo = "qux";  
10    foo2 = "qux";  
11    console.log(foo);  
12    // "qux", la variable appartient au scope de son bloc (le "if")  
13    console.log(foo2);  
14    // "qux"  
15  }  
16  console.log(foo);  
17  // "bar", la variable appartient au scope de son bloc (la fonction "fn")  
18  console.log(foo2);  
19  // "qux"  
20 }
```

let

Fonctionnalités identiques que
const mais peut être réassignée

Déclaration des variables

var

L'ancêtre de let, à utiliser le moins souvent, vous pouvez l'utiliser pour les try/catch

Attention compatibilité !


















Avant d'utiliser des variables, fonctions ou objets en javascript, il faut vérifier si elles sont compatibles avec les navigateurs web pour lesquels vous développez.

Pour ceci vous pouvez vous rendre sur le site <https://developer.mozilla.org/>

Exemple pour **let**

🔗 Compatibilité des navigateurs

Les nouveaux tableaux de compatibilité sont disponibles en beta ▼

														
														
Support simple	49	12	44 *	11	17	10	49	49	12	44 *	17	10	4.0	6.0.0

Mots réservés Javascript

- break
- case
- catch
- continue
- debugger
- default
- delete
- do
- else
- finally
- for
- function
- if
- in
- instanceof
- new
- return
- switch
- this
- throw
- try
- typeof
- var
- void
- while
- with

Les types de variables

Le javascript étant un langage typé dynamiquement, vous ne devez pas préciser le type de la variable comme en C, java etc...

```
1  const monTexte = "Texte comment";  
2  
3  const monText = 06;  
4  
5  const monBooleen = false;  
6  |
```






Les types de variables

Vous pouvez savoir le type d'une variable en utilisant l'opérateur `typeof`. De plus, vous pouvez modifier le type d'une variable en la mettant à jour.

```
1  let monTexte = "Texte comment";  
2  
3  typeof monTexte  
4  
5  monTexte = 06;  
6  
7  typeof monTexte
```

Opérateurs arithmétiques

les opérateurs arithmétiques classiques

Opération arithmétique	Symbole en JS
Addition	
Soustraction	
Multiplication	
Division	
Modulo (reste d'une division)	

Opérateurs arithmétiques

```
1  const add = 3 + 7;  
2  console.log(add);  
3  
4  const value1 = 3, value2 = 5;  
5  const mult = value1 * value2;  
6  console.log(mult);  
7  
8  
9  const divisor = 5;  
10 const calcul = (10 * 9) / 5;  
11 const div = calcul / divisor;  
12 const mod = calcul % divisor;  
13 console.log(div);  
14 console.log(mod);  
15
```

Raccourcis pour les opérations

```
1  let number = 20;
2
3  number += 5; // équivalent à number = number + 5;
4  console.log(number); //affiche 25
5
6  number -= 5; // équivalent à number = number - 5;
7  console.log(number); //affiche 20
8
9  number /= 4; // équivalent à number = number / 4;
10 console.log(number); //affiche 5
11
12 number *= 4; // équivalent à number = number * 4;
13 console.log(number); //affiche 20;
14
15 number %= 4; // équivalent à number = number % 4;
16 console.log(number); //affiche 0;
17
```

La concaténation

En javascript vous pouvez concaténer des chaînes, nombre et booléen entre eux

```
1  let concatenation = '';
2
3  const text1 = 'Hello', text2 = ' Tout le monde ', numero = 43, boolean = true;
4
5  concatenation = text1 + text2;
6  console.log(concatenation); // affiche Hello Tout le monde
7
8  concatenation += numero;
9  console.log(concatenation); // affiche Hello Tout le monde 43
10
11 concatenation += boolean;
12 console.log(concatenation); // affiche Hello Tout le monde 43true
13
```

JavaScript Popup Boxes

Il y a trois types de popup javascript, qui sont dans l'objet **window**

- **alert** : Ouvre un popup qui alerte l'utilisateur
- **confirm** : Ouvre une popup qui demande à l'utilisateur une confirmation
- **prompt**: Ouvre une popup et demande une information à l'utilisateur

window est un objet représentant une fenêtre contenant un document **DOM**

DOM (Document Object Model) est une interface de programmation pour les documents HTML, XML et SVG. Il fournit une représentation structurée du document sous forme d'un arbre et définit la façon dont la structure peut être manipulée par les programmes, en termes de style et de contenu

JavaScript Popup Boxes

```
1  alert('Attention !');  
2  
3  confirm('êtes-vous sûr de vouloir supprimer ... ?');  
4  
5  const prenom = prompt('Quel est votre prénom ?');  
6  
7  console.log('Mon prenom est ', prenom);
```

Exercice

A l'aide de ***prompt*** réaliser un petit programme qui va vous demander votre nom, prénom et afficher le résultat dans un ***alert***

Exercice

A l'aide de ***prompt*** réaliser un petit programme qui permet de faire la somme de deux nombres et afficher le résultat dans un ***alert***

Convertir des chaînes en entier

```
1  let number = '1';  
2  
3  console.log(parseInt(number)); // va afficher 1  
4  console.log(typeof number); // va afficher "number"  
5  
6  let number = 'Hello';  
7  
8  console.log(parseInt(number)); // va afficher NaN  
9  console.log(typeof number); // va afficher "number"
```


Structures conditionnelles

```
1  const number = 4;
2  if(number === 1 ) {
3      console.log('le nombre vaut 1');
4  }
5  else if( number === 2 ){
6      console.log('le nombre vaut 2');
7  }
8  else if( number === 3 ) {
9      console.log('le nombre vaut 3');
10 }
11 else {
12     console.log('le nombre vaut ...')
13 }
14
```

Exercice

A l'aide d'un prompt, demandez l'âge d'une personne.

Si + de 18 ans affichez le message ***Vous êtes majeur***

Si - de 18 ans affichez le message ***Vous êtes mineur***

Utilisez une structure conditionnelle.

Structures conditionnelles

```
1  const temperature = 18;
2
3  switch(temperature){
4      case 1:
5      case 2:
6      case 3:
7      console.log('Il fait froid');
8          break;
9      case 4:
10     case 10:
11     case 17:
12         console.log('Il fait bon');
13         break;
14     case 18:
15         console.log('Il fait chaud');
16     default:
17         console.log('...');
18 }
```

Les tableaux

```
1  const motCle = ['voiture', 'maison', 'auto'];
2  const motCle2 = new Array('voiture', 'maison', 'auto');
3
4  const motCle3 = new Array();
5  motCle3.push('voiture');
6  motCle3.push('maison');
7  motCle3.push('auto');
8
```

Les tableaux

Pour lister l'ensemble des fonctions que vous pouvez appliquer sur un tableau, faites un ***console.log(Array.prototype)***

```
> console.log(Array.prototype);
```

```
▼ [constructor: f, concat: f, find: f, findIndex: f, pop: f, ...] ⓘ
```

- ▶ concat: f concat()
- ▶ constructor: f Array()
- ▶ copyWithin: f copyWithin()
- ▶ entries: f entries()
- ▶ every: f every()
- ▶ fill: f fill()
- ▶ filter: f filter()
- ▶ find: f find()
- ▶ findIndex: f findIndex()
- ▶ forEach: f forEach()
- ▶ includes: f includes()
- ▶ indexOf: f indexOf()

Exercice

Depuis ce tableau trouver les méthodes qui permettent de supprimer le premier élément et supprimer le dernier élément

```
1  const jours = [  
2    'lundi',  
3    'mardi',  
4    'mercredi',  
5    'jeudi',  
6    'vendredi',  
7    'samedi',  
8    'dimanche'];
```

Exercice

Depuis ce tableau, trouver la méthode qui permet de supprimer le mardi

```
1  const jours = [  
2    'lundi',  
3    'mardi',  
4    'mercredi',  
5    'jeudi',  
6    'vendredi',  
7    'samedi',  
8    'dimanche'];
```

Exercice

Depuis ce tableau trouver la méthode qui permet de joindre les éléments et les séparer par une virgule.

```
1  const jours = [  
2    'lundi',  
3    'mardi',  
4    'mercredi',  
5    'jeudi',  
6    'vendredi',  
7    'samedi',  
8    'dimanche'];
```


Les objets

Voici comment déclarer des objets en javascript

```
1  const monObjet1 = {}  
2  monObjet1.voiture = 'audi';  
3  
4  const monObjet2 = new Object({voiture: 'audi'});  
5  
6  const monObjet3 = new Object();  
7  ✓ monObjet3['voiture'] = 'audi';  
8  monObjet3.maison = true;  
9
```

Les fonctions

```
1  function maFonction1(params, params2){
2
3  }
4
5  //nouvelle syntaxe
6  function maFonction2(params1, params2) => {
7
8  }
9
10 const maFonction3 = function(params1, params2){
11
12 }
13 |
14 //nouvelle syntaxe
15 const maFonction4 = (params1, params2) => {};
16
```