

# ECMAScript JavaScript VSCode

Parcours JSA 2023

# Sommaire

## ECMAScript

- Définition
- Caractéristiques de ECMAScript

## JavaScript

- Introduction
- Variables et conditions
- Concaténation
- Conditions et boucles
- Objets
- Méthodes
- Javascript et le web

## VSCode

- Éditeur
- Barre latérale
- Barre d'état
- Barre d'activité
- Panneaux



# ECMAScript

# Définition

Draft ECMA-262 / March 21, 2023

ECMAScript® 2023 Language Specification



- ECMA : European Computer Manufacturers Association.
- Ensemble de normes sur les langages de type script (1<sup>ère</sup> éd, 1997).
- Normes standardisées par Ecma international. <https://tc39.es/ecma262/#sec-ecmascript-language-source-code>
- Standard dont spécifications sont mises en œuvre dans différents langages de script.
- Langage de programmation orientée prototype.

# Caractéristiques d'ECMAScript

- Syntaxe du langage.
- Mécanisme de traitement d'erreurs.
- Types de variables.
- Objet global.
- Mécanisme d'héritage (basé sur le concept de prototype).
- Objets et fonction natifs
- Mode strict.

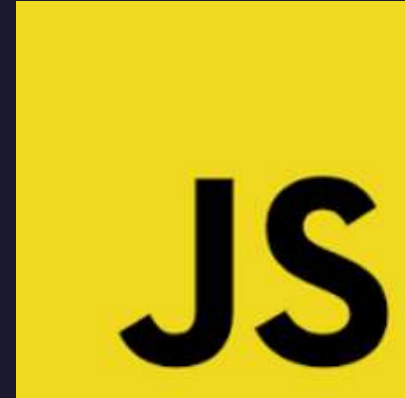


# JavaScript

# JavaScript

## INTRODUCTION

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les langages HTML et CSS, JavaScript est au cœur des langages utilisés par les développeurs web. Il a été créé en 1995 par Brendan Eich et intégré au navigateur web Netscape Navigator 2.0.



# JavaScript

## VARIABLES ET CONDITIONS

- Déclarer une variable

La déclaration se fait avec le mot clé "let"

(ou anciennement "var")

Il n'est pas nécessaire de définir le type de la variable.

Les constantes sont déclarée à l'aide de "const"

Initialisation d'un tableau

```
let compteur = 0;
```

```
let userIsSignedIn = true;  
let userIsAdmin = false;
```

```
let firstName = "Will";  
let lastName = 'Alexander';
```

```
1 const nombrePostParPage = 20;
```

```
let guests = [];
```



# JavaScript

## CONCATÉINATION

- Il est possible d'effectuer la concaténation par plusieurs méthodes.
- La méthode classique avec l'opérateur + :

```
let wholeName = firstName + " " + lastName; // valeur: "Will Alexander"
```

- La string interpolation via l'expression `${maVariable}`:

```
const myName = `Alexander`;  
const salutation = `Bienvenue sur mon site ${myName}!`;  
console.log(salutation); //retournera "Bienvenue sur mon site Alexander!"
```

# JavaScript

## CONDITIONS ET BOUCLES 1/2

- Nous retrouvons les conditions avec la même syntaxe

qu'en JAVA

```
if (userLoggedIn) {
  welcomeMessage = 'Welcome back!'; //
} else {
  welcomeMessage = 'Welcome new user!';
}

console.log(welcomeMessage); // imprime '
```

```
switch (firstUser.accountLevel) {
  case 'normal':
    console.log('You have a normal account!');

    break;
  case 'premium':
    console.log('You have a premium account!');

    break;
  case 'mega-premium':
    console.log('You have a mega premium account!');

    break;
  default:
    console.log('Unknown account type!');
}
```

```
for (let i = 0; i < numberOfPassengers; i++) {
  console.log("Passager embarqué !");
}
```

# JavaScript

## CONDITIONS ET BOUCLES 2/2

- Deux boucles simplifiées pour parcourir les tableaux.
- Boucle for ..in

```
for (let i in passengers) {  
  console.log("Embarquement du passager " + passengers[i]);  
}
```

- Boucle for ..of

```
for (let passenger of passengers) {  
  console.log("Embarquement du passager " + passenger);  
}
```

Initialisation d'un tableau

```
let guests = [];
```

# JavaScript

## LES OBJETS 1/2

- Les objets JavaScript sont écrits en JSON (JavaScript Object Notation). Ce sont des séries de **paires clés-valeurs** séparées par des virgules.
- Une fois l'objet créé, on peut accéder aux attributs via la notation pointée ou la notation bracket

```
let myBook = {  
  title: "L'Histoire de Tao",  
  author: "Will Alexander",  
  numberOfPages: 250,  
  isAvailable: true  
};  
let bookTitle = myBook.title; // "L'Histoire de Tao"  
let bookPages = myBook.numberOfPages // 250
```

```
let bookTitle = myBook["title"]; // "L'Histoire de Tao"  
let bookPages = myBook["numberOfPages"]; // 250
```

# JavaScript

## LES OBJETS 2/2

- Pour la création de nombreux objets il sera utile de passer par la création de classes.
- Création :
- Instanciation et affichage :

```
class Book {  
  constructor(title, author, pages) {  
    this.title = title;  
    this.author = author;  
    this.pages = pages;  
  }  
}
```

```
let myBook = new Book("L'Histoire de Tao", "Will Alexander", 250);  
//Cette ligne crée l'objet suivant :  
{  
  title: "L'Histoire de Tao",  
  author: "Will Alexander",  
  pages: 250  
}
```



# JavaScript

## MÉTHODES

- Fonctions
- Methodes d'instance  
(après les attributs)
- Methodes statiques

Permet l'appel de la fonction sans instance.

```
function afficherDeuxValeurs(valeur1, valeur2) {  
    console.log('Première valeur:' + valeur1);  
    console.log('Deuxième valeur:' + valeur2);  
}
```

```
showBalance() {  
    console.log("Solde: " + this.balance + " EUR");  
}
```

```
class BePolite {  
  
    static sayHello() {  
        console.log("Hello!");  
    }  
}
```

# JavaScript

## JAVASCRIPT ET LE WEB

- Le code JavaScript s'exécute dans le navigateur lors de son appel

via la balise `<script>`.

- Il peut également être utilisé côté serveur notamment via Node.

```
<!DOCTYPE html>
<html lang="en" >
<head>
  <meta charset="UTF-8">
  <title>Page web</title>
  <link rel="stylesheet" href="./style.css">
</head>
<body>
  <p>Contenu de la page</p>
  <script src="./script.js"></script>
</body>
</html>
```

# Visual Studio Code

Développé par Microsoft

Disponible pour Windows, macOS et Linux (♡)

# Présentation de l'interface utilisateur

- **Éditeur** - La zone principale pour éditer vos fichiers. Vous pouvez ouvrir autant d'éditeurs que vous le souhaitez côte à côte verticalement et horizontalement.
- **Barre latérale** - Contient différentes vues comme l'Explorateur pour vous aider à travailler sur votre projet.
- **Barre d'état** - Informations sur le projet ouvert et les fichiers que vous modifiez.
- **Barre d'activité** - Située à l'extrême gauche, elle vous permet de basculer entre les vues et vous donne des indicateurs supplémentaires spécifiques au contexte, comme le nombre de modifications sortantes lorsque Git est activé.
- **Panneaux** - Vous pouvez afficher différents panneaux sous la région de l'éditeur pour les informations de sortie ou de débogage, les erreurs et les avertissements, ou un terminal intégré.

# Présentation de la barre d'activité

La barre d'activité est une surface de navigation centrale dans VS Code. Les extensions peuvent ajouter des conteneurs de vue à la barre d'activité qui apparaissent en tant qu'éléments de la barre d'activité. Voici les éléments qui apparaissent par défaut :

- Explorateur (Ctrl+Maj+E) - Permet d'obtenir rapidement et facilement les fichiers dont vous avez besoin.
- Recherche (Ctrl+Maj+F) - Permet de rechercher un élément présent dans l'éditeur.
- Contrôle de code source (Ctrl+Maj+G) - Permet de travailler avec un dépôt distant directement dans l'éditeur.
- Exécuter et déboguer (Ctrl+Maj+D) - Permet au développeur d'obtenir de l'aide et analyser les bugs de son programme.
- Extensions (Ctrl+Maj+X) - Permet d'ajouter et de gérer les extensions de l'éditeur.



# Sources

## VSCode

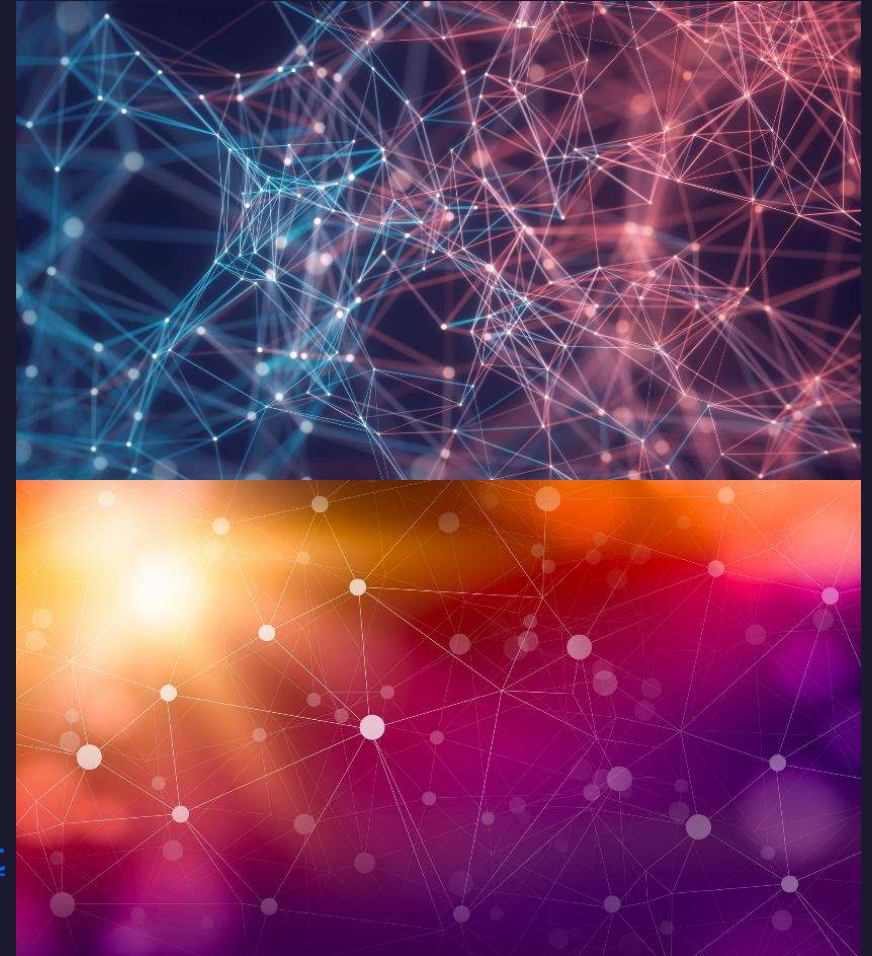
- <https://code.visualstudio.com/>
- <https://code.visualstudio.com/docs/getstarted/userinterface>

## JavaScript

- <https://openclassrooms.com/fr/courses/6175841-apprenez-a-programmer-avec-javascript>
- <https://fr.wikipedia.org/wiki/JavaScript>

## ECMAScript

- [https://developer.mozilla.org/fr/docs/Web/JavaScript/JavaScript\\_technologies\\_overview](https://developer.mozilla.org/fr/docs/Web/JavaScript/JavaScript_technologies_overview)



Pour finir, « Des Mots » de VSC par Camille ;-)