

# Node JS

## Développement d'applications Web

**Intervenant : Jean-Frédéric VINCENT**



An aerial photograph of a vast, dense tropical rainforest. The canopy is a thick, textured layer of green, with various shades of emerald and forest green. The sky above is bright and hazy, with soft, diffused light filtering through the clouds. The overall mood is serene and majestic.

Module 1

# Introduction

# Introduction

Le principe du javaScript

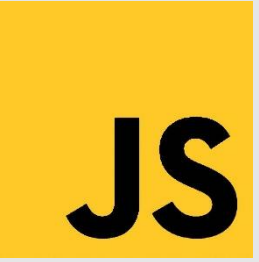
Les Frameworks JavaScript

Présentation du serveur NodeJS

Et JQuery ?

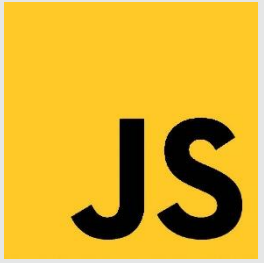
Le DOM

# Présentation



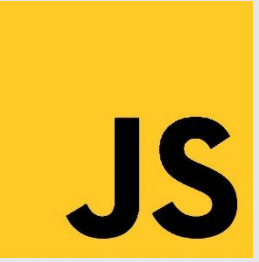
- **JavaScript** est un langage de programmation principalement utilisé pour créer des interactions dynamiques sur des pages web. Il est exécuté côté client (dans le navigateur web), mais peut également être utilisé côté serveur grâce à Node.js.
- Développé en 1995 par **Brendan Eich** sous le nom de "LiveScript", JavaScript est rapidement devenu un standard incontournable dans le domaine du développement web. Le nom "JavaScript" a été choisi pour des raisons marketing, afin de capitaliser sur la popularité du langage Java. Toutefois, ces deux langages partagent peu de similarités et ne doivent pas être confondus.

# Place de JavaScript dans le web



- **JavaScript** occupe une place **centrale** dans le développement des applications web côté client, en complément des langages HTML et CSS. Il permet d'ajouter des **fonctionnalités interactives** aux pages web et de répondre dynamiquement aux actions de l'utilisateur, sans nécessiter le rechargement de la page.
- Grâce à l'environnement **Node.js**, JavaScript a étendu son champ d'application au développement côté serveur, permettant ainsi la création d'**API** et la gestion de bases de données.
- En combinant JavaScript avec des frameworks tels que **React**, **Angular** ou **Vue** ainsi que Node.js, il est désormais possible de développer des applications web complètes (full-stack) en utilisant un seul et même langage de programmation.

# Caractéristiques



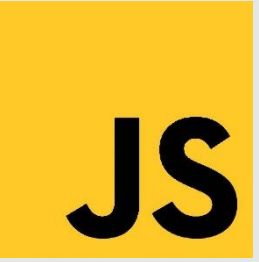
- **Langage interprété**

JavaScript est un langage interprété, ce qui signifie qu'il est exécuté ligne par ligne directement par le navigateur, sans nécessiter de compilation préalable.

- **Typage faible**

JavaScript adopte un typage faible, ce qui implique qu'il n'est pas nécessaire de déclarer explicitement le type des variables, et celles-ci peuvent être modifiées en cours d'exécution. Cette flexibilité peut entraîner des comportements inattendus, d'où l'importance d'une gestion prudente des types de données.

# Caractéristiques



- **Asynchrone et basé sur les événements**

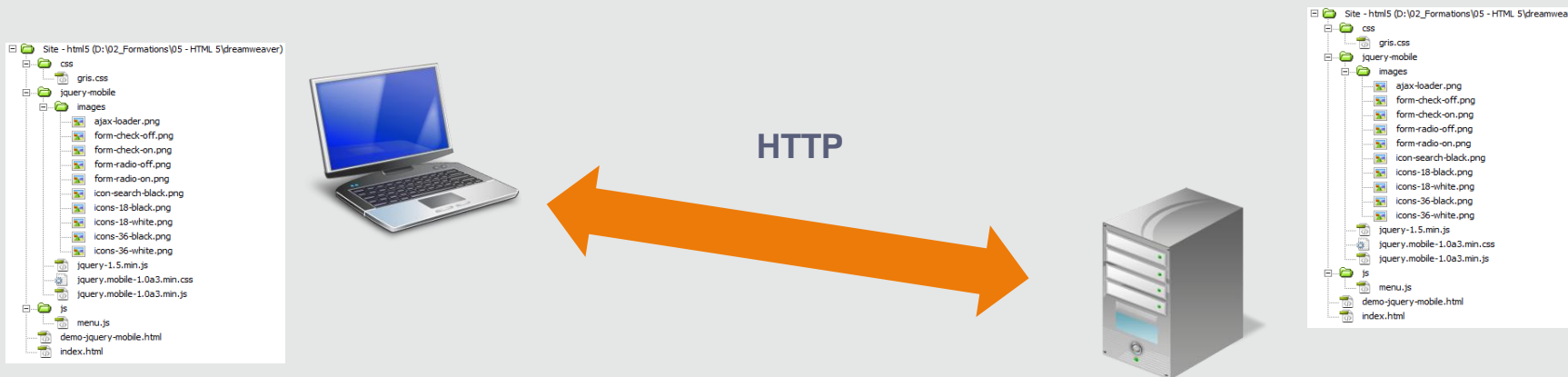
JavaScript repose sur un modèle de programmation asynchrone, qui permet de gérer les événements (tels que les clics utilisateurs ou les requêtes réseau) à l'aide de mécanismes comme les callbacks, les promises et les structures async/await. Cette approche permet d'exécuter des opérations longues (comme des appels API) sans bloquer l'exécution du programme principal.

- **Basé sur les objets**

JavaScript est un langage orienté objet. Tous les types natifs (tels que String, Array, etc.) sont considérés comme des objets, ce qui permet d'appliquer des principes de la programmation orientée objet dans la conception du code.

# Le principe du javaScript

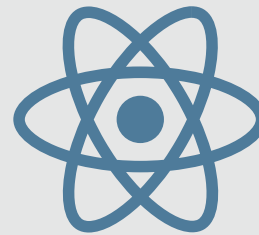
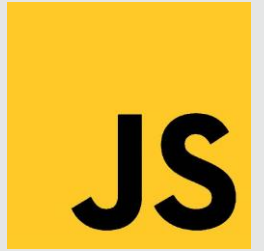
- Les fichiers JavaScript sont téléchargés sur l'ordinateur du client
- C'est le navigateur web qui exécute le JavaScript (et non le serveur)
- Problème le code source est accessible !





# Les Framework JavaScript

- **Vanilla** Javascript **VERSUS** Framework JavaScript
- Moins de codes
- Plus de fonctionnalités
- Code mieux "rangé"
- Maintenance



React





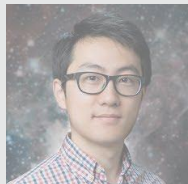
Angular



Vue

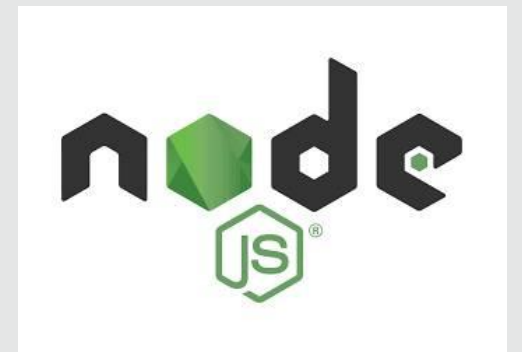
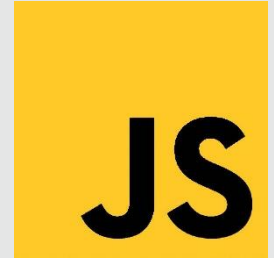
# Les Framework JavaScript

- Les 3 Framework JavaScript les plus utilisés

1		React	
2		Angular	
3		Vue	 Evan You

# Présentation du serveur NodeJS

- NodeJS est un serveur JavaScript
- Il exécute donc du code Javascript
- Il est basé sur le moteur V8 de Chrome
- Il est possible d'installer le serveur NodeJS en local sur sa machine de développement
  - Pour créer des projets JS, React, Angular ou Vue



# Introduction

# Jquery

- La fin d'un règne 2009 - 2019 !
- Vive Vanilla JavaScript ou juste JavaScript

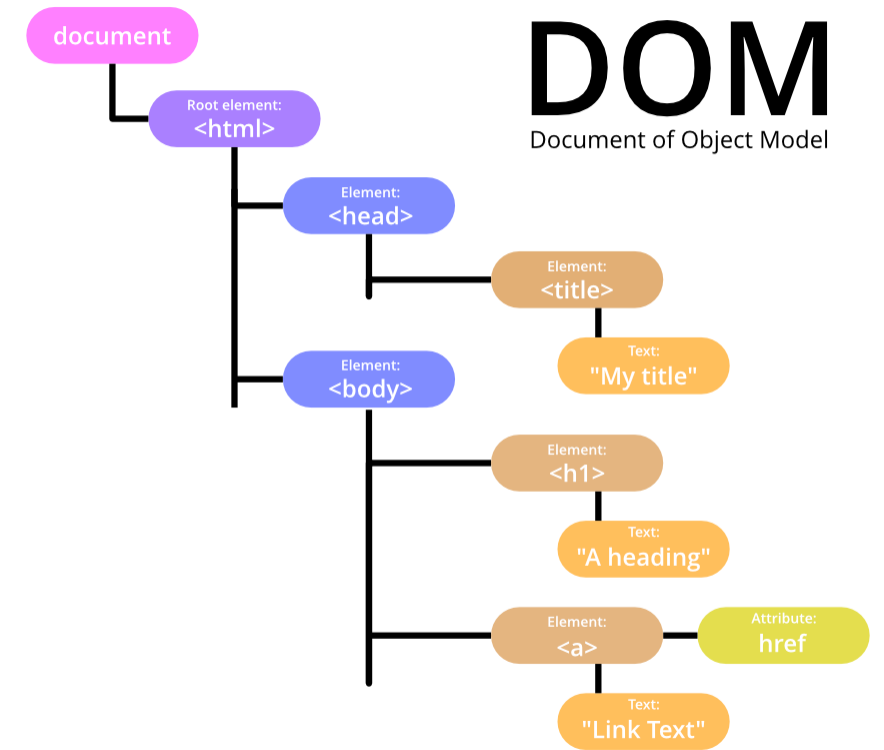


- Mais encore bien présent ....
- Plugin **Select2** et autres ...

# Introduction

## Le DOM

- Le DOM : **D**ocument **O**bject **M**odel
- Le DOM est un modèle de document chargé dans le navigateur.



- La représentation du document est un arbre avec des nœuds
- Chaque nœud représente une partie du document

# ECMA SCRIPT, c'est quoi

- **ECMAScript** est un ensemble de normes concernant les langages de programmation de type script et standardisées par Ecma International dans le cadre de la spécification ECMA-262. Il s'agit donc d'un **standard**, dont les spécifications sont mises en œuvre dans différents langages de script, comme **JavaScript** ou ActionScript.

# ECMA SCRIPT, c'est quoi

- ECMAScript Edition **6 (ES 6/ES 2015)**

**Modules**, **classes**, portée lexicale au niveau des blocs, itérateurs et générateurs, **promesses** pour la programmation **asynchrone**, patrons de destructuration, optimisation des appels terminaux, nouvelles structures de données (tableaux associatifs, ensembles, tableaux binaires), support de caractères Unicode supplémentaires dans les chaînes de caractères et les expressions rationnelles, possibilité d'étendre les structures de données prédéfinies.

- ECMAScript Edition **15 (ES 15 /ES 2024)**

Méthode `groupBy` sur les objets et les maps, méthodes `isWellFormed` et `toWellFormed` sur les chaînes, méthode `withResolvers` sur **Promise**, les **ArrayBuffers** peuvent être redimensionnés et transférés.

# ECMA SCRIPT, c'est quoi

- **ECMAScript** Edition **6** (**ES 6**/ES 2015)
  - Modules
  - Classes
  - Promesses
  - Asynchrone await et async
- **ECMAScript** Edition **15** (**ES 15** /ES 2024)
  - Promesses
  - ArrayBuffers