**Comparaison entre JavaScript et Python**

Mohameden Debagh 23069

Ahmed Salim El khay 23090

Mariem Bacar 23049

Mouna El mokhtar 22089

Institut Supérieur du Numérique

# Abstrait

La comparaison entre JavaScript et Python, deux langages de programmation polyvalents et populaires, révèle des différences significatives ainsi que des similitudes. Ce résumé se penche sur les aspects clés de ces langages, y compris leur syntaxe, leur utilisation, et leurs caractéristiques distinctives.

JavaScript, souvent utilisé pour le développement web, offre une syntaxe dynamique et orientée objet. Python, reconnu pour sa lisibilité, présente une syntaxe claire et concise. La comparaison des deux souligne des différences dans la déclaration de variables, les structures de contrôle, et les fonctions, chaque langage ayant ses propres conventions.

**Contents**

[Abstract 2](#_Toc160494097)

[Contenus 3](#_Toc160494098)

[Introduction 4](#_Toc160494099)

[Histoire et Origine: 5](#_Toc160494100)

[Différences et similitudes: 6](#_Toc160494101)

[Syntaxe et structure: 6](#_Toc160494102)

[Performance: 8](#_Toc160494104)

[Domaines d'application: 9](#_Toc160494106)

[Bibliothèques standards riches : 10](#_Toc160494108)

[Conclusion: 11](#_Toc160494111)

[References 13](#_Toc160494112)

# 

# Introduction

Dans le domaine du développement logiciel, le choix du langage de programmation est crucial pour le succès d'un projet. Deux langages de programmation populaires, Python et JavaScript, sont largement utilisés pour différentes applications. Ce projet professionnel se concentre sur une analyse comparative approfondie entre Python et JavaScript, mettant en évidence leurs forces, faiblesses, et domaines d'application spécifiques.

Le choix d'un langage de programmation est crucial dans le développement logiciel. Ce projet vise à effectuer une analyse approfondie des différences entre Python et JavaScript, deux langages de programmation populaires. L'objectif est de fournir des informations détaillées pour aider les développeurs et les organisations à prendre des décisions éclairées.

JavaScript né dans les débuts du web, JavaScript a évolué pour devenir un acteur central dans la création d'interfaces interactives pour les navigateurs . Sa polyvalence s'étend du développement front-end avec des frameworks tels que React et Vue, jusqu'au côté serveur avec Node.js. JavaScript offre une exécution asynchrone, permettant des expériences utilisateur réactives et dynamiques.

Python Conçu pour la lisibilité et la simplicité, Python a conquis divers domaines, de l'automatisation des tâches quotidiennes à des applications complexes en science des données et intelligence artificielle. Son approche syntaxique claire et sa grande communauté en font un choix privilégié pour les débutants et les développeurs expérimentés.

# Histoire et Origine:

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 C'est un langage orienté objet à prototype : les bases du langage et ses principales interfaces sont fournies par des objets. Cependant, à la différence d'un langage orienté objets à classes, les objets de base ne sont pas des instances de classes. En outre, les fonctions sont des objets de première classe. Le langage supporte le paradigme objet, impératif et fonctionnel.  
Une grande majorité des sites web l'utilisent, et la majorité des navigateurs web disposent d'un moteur JavaScript pour l'interpréter.

JavaScript, créé par Brendan Eich en 1995, a été initialement conçu pour rendre les pages web interactives du côté client. Son évolution rapide l'a propulsé au-delà du navigateur pour devenir un langage de programmation à part entière, alimentant des frameworks front-end populaires tels que React et Angular. JavaScript est devenu incontournable pour le développement web, apportant dynamisme et interactivité aux sites modernes.

Python est un langage de programmation objet interprété. Son origine est le langage de script du système d’exploitation Amoeba (1990). Il a été développé par Guido Von Rossum au CWI, à l’Université d’Amsterdam et nommé par rapport au Monthy Python’s Flying Circus.  
Depuis, Python est devenu un langage de programmation généraliste (comp.lang.python est créé en 1994). Python est un langage open source supporté, développé et utilisé par une large communauté .  
Il offre un environnement complet de développement comprenant un interpréteur performant et de nombreux modules. Un atout indéniable est sa disponibilité sur la grande majorité des plates-formes courantes (BeOS, Mac OS X, Unix, Windows).

# Le langage Python a évolué au cours de son histoire. En raison de ses bibliothèques spécialisées, Python peut désormais être utilisé dans de nombreux contextes et s’adapter à tout type d’utilisation. Il demeure largement utilisé comme langage de script pour l’automatisation de tâches simples mais fastidieuses. Pour autant, il est également utilisé comme langage de développement de prototypes afin de concevoir une application fonctionnelle. Ensuite optimisée avec un langage de plus bas niveau.

# Differences and similitudes:

### Syntaxe et Structure

La première différence qui saute aux yeux (littéralement) c'est bien entendu les différences de syntaxe qui existent entre les deux langages. Bien qu'ils restent assez proches, la syntaxe de Python est connue pour être plus minimaliste (et donc plus facile à apprendre).

**Délimitation des blocs** : Avec Python on utilisera l’indentation pour délimiter les blocs de code. Ce qui facilite la compréhension de la structure du programme.

Avec JavaScript ce sont les accolades qui vont délimiter les blocs. De ce fait la syntaxe offre une plus grande flexibilité quant à la façon d’écrire notre code.

**Déclaration des variables**: Lors de la déclaration d’une variable, il n’y a pas besoin de spécifier son type. Il suffit de lui attribuer la valeur voulue, et Python déterminera le type de la variable en fonction de la valeur affectée.

À savoir que Python est un langage dynamique, ce qui signifie que le type de la variable peut être modifié au cours de l’exécution du programme. Aussi, c’est un langage fortement typé. C’est-à-dire que les variables ont un type défini, et que ce dernier ne peut pas être modifié implicitement. Ainsi les variables sont contrôlées en fonction de leur type.

**Les boucles et conditions** : Que ce soit Python ou JavaScript le fonctionnement de base reste le même. Nous pouvons faire le rapprochement avec ce que nous avons vu plus tôt sur la syntaxe. Ci-dessous nous allons itérer sur une suite de données allant de 0 à 4 et afficher la valeur à chaque itération. On utilisera la boucle for. D’abord avec Python où nous déclarons une variable i et une suite de données avec range.

Avec JavaScript nous déclarons une variable avec let i, puis une condition qui sera vérifiée avant chaque itération de la boucle, ici i < 5. Si la condition est évaluée comme false la boucle se termine. Pour que la condition puisse être évaluée à false on modifie la valeur de i en incrémentant 1 à chaque itération avec i++ (sans incrémenter i, la condition serait toujours vraie et la boucle ne pourrait pas s'arrêter).

**Les conditions :** En ce qui concerne les conditions Python et JavaScript utilisent le mot if suivi d’une expression booléenne pour évaluer une condition. Avec Python l’opérateur d’égalité est (==) , et si on vérifie plusieurs conditions on utilisera elif.

Quelques différences à noter avec JavaScript tel que l’utilisation de === comme opérateur d’égalité, et l’utilisation de else if lorsqu’on vérifie plusieurs conditions.

### Performance:

### La performance de JavaScript et Python dépend de plusieurs facteurs, notamment leur utilisation spécifique, les implémentations sous-jacentes, et les cas d'utilisation particuliers. Examinons brièvement la performance relative de ces deux langages dans différents contextes : JavaScript : Côté Client (Navigateur) : JavaScript est le langage principal pour le développement côté client dans les navigateurs web. Il est optimisé pour l'exécution rapide dans ces environnements, avec des moteurs tels que V8 (utilisé dans Google Chrome) et SpiderMonkey (utilisé dans Mozilla Firefox). Node.js (Côté Serveur) : Node.js permet d'exécuter du code JavaScript côté serveur. Bien qu'il soit performant pour les applications basées sur des opérations d'entrée/sortie non bloquantes, il peut ne pas être aussi efficace que certains langages orientés serveur dans des cas spécifiques. Python : Polyvalence : Python est reconnu pour sa simplicité et sa lisibilité, mais il peut être moins performant que certains langages compilés dans certaines situations. Cependant, ses performances ont considérablement augmenté au fil du temps, et des améliorations continues sont apportées par les implémentations telles que CPython, PyPy, et d'autres. Domaines Spécifiques : Python excelle dans des domaines tels que la science des données et l'intelligence artificielle grâce à des bibliothèques comme NumPy et TensorFlow. Bien que ces bibliothèques soient souvent écrites en langages compilés sous-jacents pour des performances élevées, Python offre une interface conviviale.

**Domaines d'application :**

Les domaines d'application de JavaScript et Python varient en fonction de leurs forces et de leurs fonctionnalités spécifiques. Voici une vue d'ensemble des principaux domaines dans lesquels ces langages sont couramment utilisés :

* **JavaScript :**

**Développement Web Frontend** : JavaScript est incontournable pour le développement côté client, permettant d'ajouter une interactivité dynamique aux sites web. Il est souvent utilisé avec des frameworks tels que React, Vue.js, et Angular.

**Développement Web Backend** : Node.js, basé sur JavaScript, est utilisé pour exécuter du code côté serveur, facilitant le développement complet d'applications web avec un seul langage. **Applications Mobiles** : Des frameworks tels que React Native permettent d'utiliser JavaScript pour le développement d'applications mobiles, partageant du code entre les plateformes.

**Jeux Vidéo** : Des moteurs de jeu tels que Phaser et Three.js utilisent JavaScript pour la création de jeux en ligne et de graphismes 3D dans les navigateurs.

* **Python** :

**Science des Données et Analyse** : Python est largement utilisé dans la science des données avec des bibliothèques comme NumPy, Pandas, Matplotlib et Scikit-Learn pour l'analyse de données, la modélisation statistique, et le machine learning.

**Intelligence Artificielle et Apprentissage Machine** : TensorFlow, PyTorch et Keras sont des frameworks populaires en Python pour le développement d'applications d'intelligence artificielle et de modèles d'apprentissage machine.

**Développement Web Backend** : Des frameworks comme Django et Flask font de Python un choix puissant pour le développement côté serveur, offrant une facilité de développement et une extensibilité.

**Automatisation et Scripting** : Python est souvent utilisé pour l'automatisation de tâches, la création de scripts système, et le développement d'applications de ligne de commande. **Développement de Jeux Vidéo** : Des moteurs de jeu comme Pygame permettent de créer des jeux vidéo en utilisant Python. Applications Scientifiques et Ingénierie : Python est utilisé dans des domaines tels que la modélisation mathématique, la simulation, et le traitement du signal.

# Bibliothèques standards riches :

# JavaScript : JavaScript possède une bibliothèque standard intégrée appelée "ECMAScript". Toutefois, JavaScript tire une grande partie de sa puissance de ses bibliothèques tierces et frameworks. Parmi les plus notables, on trouve :

# Node.js : Une plateforme basée sur JavaScript qui permet d'exécuter du code JavaScript côté serveur. Node.js offre une multitude de modules intégrés pour faciliter le développement de serveurs et d'applications.

# React : Une bibliothèque JavaScript développée par Facebook, utilisée pour la construction d'interfaces utilisateur interactives, en particulier pour les applications web à page unique (SPA).

# Express.js : Un framework web minimaliste pour Node.js qui simplifie la création d'applications web et d'API.

# Python : Python possède une bibliothèque standard exhaustive qui couvre un large éventail de fonctionnalités. Quelques exemples incluent :

# NumPy : Une bibliothèque pour le calcul scientifique qui prend en charge les tableaux multidimensionnels et les fonctions mathématiques de haut niveau. Pandas : Une bibliothèque pour la manipulation et l'analyse de données, fournissant des structures de données performantes et des outils de manipulation.

# Django : Un framework web puissant pour le développement rapide d'applications web sécurisées.

# Tkinter : Une bibliothèque graphique pour la création d'interfaces graphiques (GUI) de base.

# Conclusion:

Ce projet professionnel vise à fournir une analyse approfondie et impartiale de Python et JavaScript, aidant ainsi les développeurs et les organisations à prendre des décisions éclairées lors du choix d'un langage pour leurs projets. La documentation technique et la présentation permettront une diffusion efficace des résultats obtenus.

En résumé, Python et JavaScript sont des langages polyvalents, mais ils sont souvent utilisés dans des contextes différents. Python excelle dans la science des données, l'apprentissage automatique, et le développement d'applications de bureau, tandis que JavaScript est essentiel pour le développement web côté client. La compréhension des différences entre ces langages permet de choisir le plus approprié en fonction des besoins du projet.

# References

1. <https://www.turing.com/>
2. <https://dev.to/kachiic/learning-javascript-as-a-python-developer>
3. <https://www.docstring.fr/blog/python-vs-javascript/>
4. <https://www.lemondeinformatique.fr/actualites/lire-les-employeurs-veulent-javascript-mais-les-developpeurs-preferent-python-70700.html>
5. <https://javascript.info/>