Introduction to JavaScript

Javascript is programming language that has following features :

- **High - level :** It does memory management automatically using *Garbage collector* unlike C language that is low-level
- **Dynamic :** It executes at *runtime* unlike C language which is static and executes at *compilation time*
- **Dynamically/Loosely typed :** Variables are not assigned to specific datatype.

  ```
  let name = 'peter';
  console.log(name)   // peter
  name = 1;
  console.log(name)   // 1
  ```

  Unlike C language which is statically typed,variables must have proper datatype.

  ```
  int a = 1;
  int a = 'peter';  // wrong
  ```

- **Interpreted :** It doesn't need compilation stage before program runs unlike C language.
- **Multi-Paradigm :** It supports multiple programming paradigms like :
  - *Procedural programming*
  - *Functional programming*
  - *Object-oriented programming*

Cool features of JavaScript :

1. **Single/Multi- threaded :** *Javascript is a single threaded language by default*, because it executes one statement at a time, and each statement must run to completion before the next one can start. *(Read* ***Interesting fact*** *below to understand* **multi-threaded***)*
2. **Imperative/Declarative :** *JavaScript can be both imperative and declarative*, depending on the way it is used.

Imperative programming involves giving step-by-step instructions for how a program should execute. In JavaScript, this might involve using for-loops or if-statements to control the flow of the program.

Declarative programming involves describing what a program should accomplish, rather than specifying how it should do so. In JavaScript, this might involve using higher-order functions like `map`, `filter`, or `reduce` to manipulate data in a more functional style.

Many modern JavaScript frameworks, such as React and Vue, ***emphasize a declarative programming style*** that makes it easier to reason about complex user interfaces.

3. **Synchronous/Asynchronous :** *JavaScript is synchronous by default*.

   Synchronous code is executed in a sequential manner, with each line of code being executed one after the other. This means that the program will wait for each line of code to finish executing before moving on to the next one. *Javascript has only one call stack, therefore it is synchronous language*

   Asynchronous code, on the other hand, allows multiple operations to be performed concurrently. This is achieved through the use of callbacks, promises, and async/await.

Interesting fact :

When you visit a web page, you run a browser to do so (Chrome, Firefox, Safari, Edge). Each browser has its own version of *JavaScript Runtime* with a set of *Web API's*. In a synchronous language, only one thing can be done at a time. Imagine an alert on the page, blocking the user from accessing any part of the page until the OK button is clicked. If everything in JavaScript that took a significant amount of time blocked the browser, then we would have a pretty bad user experience. This is where *concurrency* and the *event loop* come in. Even though JavaScript is a single threaded language, there are *WebWorkers* in the browser and *WorkerThreads* in Node.js that work in the background that don't block the main thread. Just like a browser creates a new thread when you open a new tab.

We will study how JavaScript works

JavaScript Engine,Execution Context,Call stack,Memory Heap,Event loop,Callback queue,Job Queue