**Memory Management in JavaScript:**

* It's handled automatically by the runtime environment, typically the JavaScript engine in web browsers or Node.js
* JavaScript uses a garbage collector to manage memory and ensure that developers do not need to manually allocate or deallocate memory.

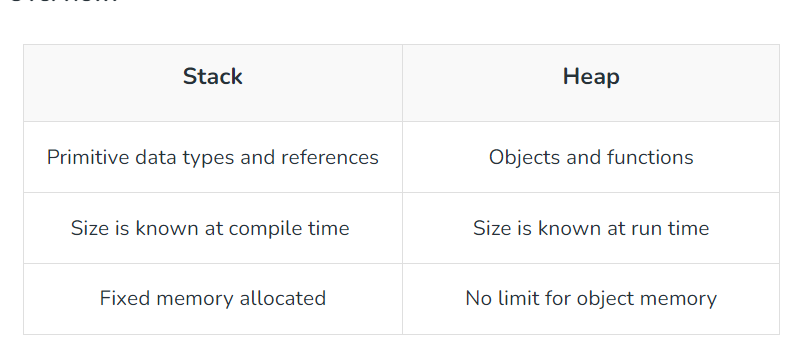
**Memory Life Cycle:**

* **Allocates the memory we need:** JavaScript allocates memory to the object created.
* **Use the allocated memory.**
* **Release the memory when not in use:** Once the allocated memory is released, it is used for other purposes. It is handled by a JavaScript engine.

**JavaScript engines have two places to store data:**

**Stack:** It is a data structure used to store static data. Static data refers to data whose size is known by the engine during compile time. In JavaScript, static data includes primitive values like strings, numbers, Boolean, null, and undefined. References that point to objects and functions are also included. A fixed amount of memory is allocated for static data. process is known as static memory allocation.

**Heap:** It is used to store objects and functions in JavaScript. The engine doesn’t allocate a fixed amount of memory. Instead, it allocates more space as required.



Example:

const employee = {

  name: 'Rajesh',

  age: 30,

};

const name="Ram"

// Allocates memory for object in heap.Values

// in object are primitive,which is why they

// are stored in stack.

**function** getname(name) {

**return** name;

}

// The function return value is given to stack after

// being evaluated in the heap

const newEmployee = employee;

// The newEmployee object will be stored in the stack and

// it will refer to the employee object in heap

