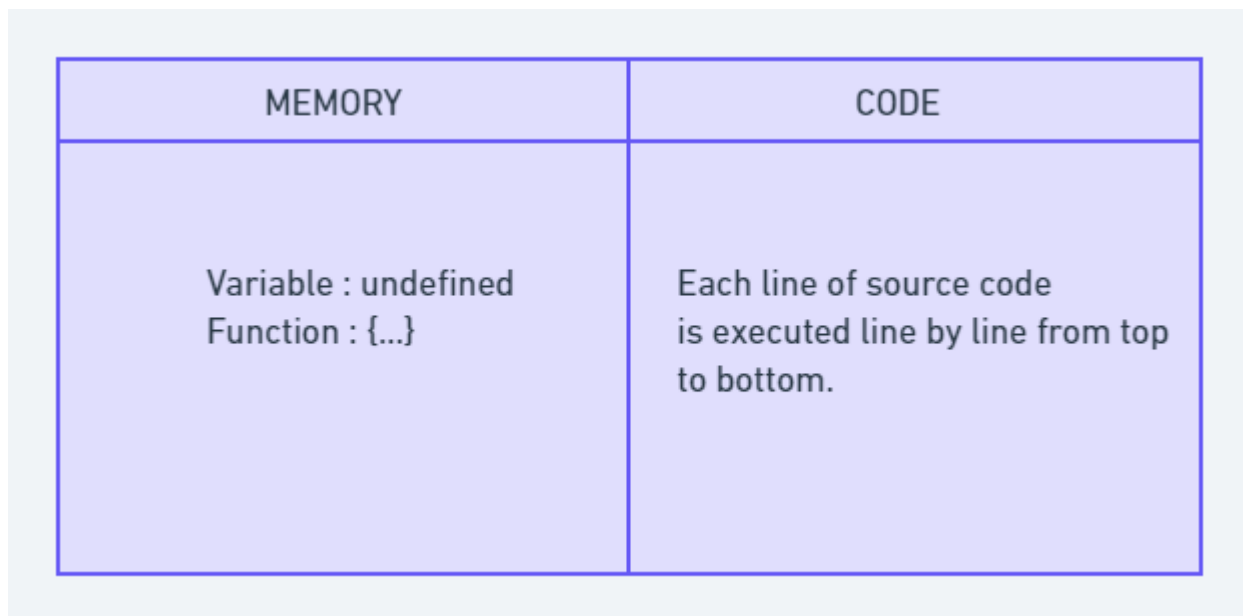**Explain what is execution context in detail with diagram?**

When the JavaScript engine scans a script file, it makes an environment called the **Execution Context** that handles the entire transformation and execution of the code.
During the context runtime, the parser parses the source code and allocates memory for the variables and functions. The source code is generated and gets executed.

There are two types of execution contexts: **global** and **function**. The global execution context is created when a JavaScript script first starts to run, and it represents the global scope in JavaScript. A function execution context is created whenever a function is called, representing the function's local scope.

Creation Phase

MEMORY	CODE
Variable : undefined Function : {...}	Each line of source code is executed line by line from top to bottom.

Let's take this simple example once again:

```
var n = 5;
function square(n) {
    var ans = n * n;
    return ans;
}
var square1 = square(n);
var square2 = square(8);

console.log(square1)
console.log(square2)
```

This is called the creation phase. Here's a diagram to help explain it:

MEMORY	CODE
n : undefined square : {...} square1 : undefined square2 : undefined	