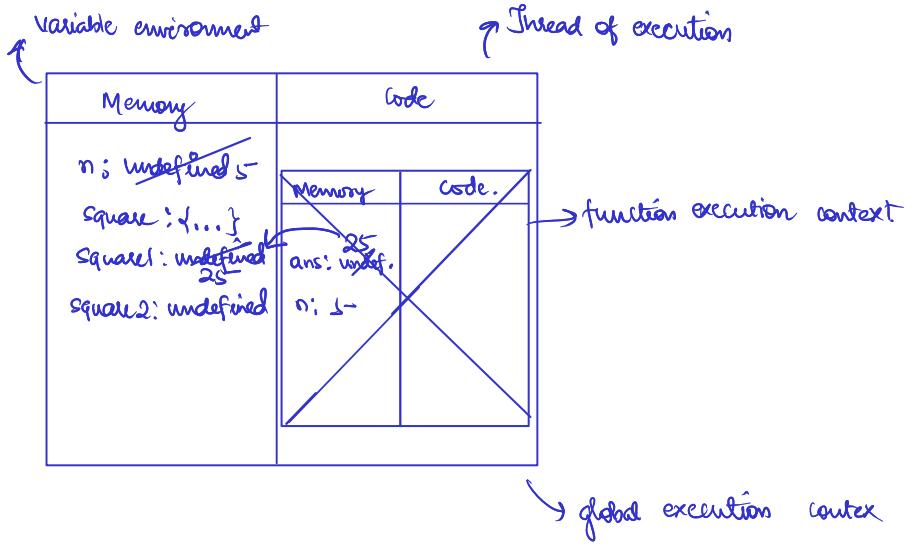


Execution Context:-

```
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)
```



```
let n = 5;
function square(n) {
  let ans = n * n;
  return ans;
}

let square1 = square(n);
let square2 = square(8);

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

→ `let` & `const` are not hoisted. They lie in the **temporal dead zone**.

Hoisting in JS:-

- Hoistable declarations include `function`, `function*`, `async function`, `async function*` & `var`
- `let`, `const` & `classes` are also hoisted but they don't have default initialization.

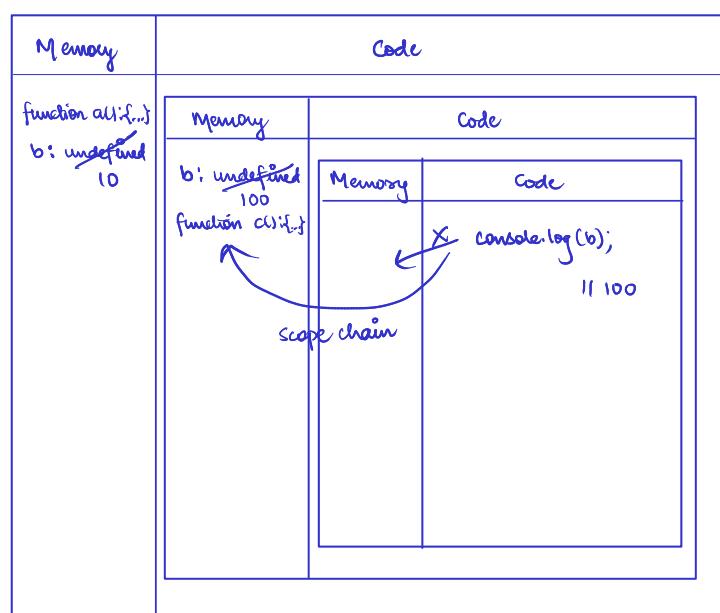
'undefined' vs 'not defined'

↓ ↓

Declared but not initialized (Memory is allocated but no value)	Not declared. (Memory is not allocated)
--	--

Scope chain, Scope & Lexical Environment

```
function a() {
  var b = 100;
  c();
  function c() {
    console.log(b); // 100
  }
  var b = 10;
  a();
}
```



Types of Error: Syntax, Reference, and Type.

Uncaught ReferenceError: x is not defined at ...

This Error signifies that x has never been in the scope of the program. This literally means that x was never defined/declared and is being tried to be accessed.

Uncaught ReferenceError: cannot access 'a' before initialization

This Error signifies that 'a' cannot be accessed because it is declared as 'let' and since it is not assigned a value, it is in its Temporal Dead Zone. Thus, this error occurs.

Uncaught SyntaxError: Identifier 'a' has already been declared

This Error signifies that we are redeclaring a variable that is 'let' declared. No execution will take place.

Uncaught SyntaxError: Missing initializer in const declaration

This Error signifies that we haven't initialized or assigned value to a const declaration.

Uncaught TypeError: Assignment to constant variable

This Error signifies that we are reassigning to a const variable.

let & const are block scoped & var is function scoped.

Closures :-

```
function a() {
  var c = 10;
  function b() {
    console.log(c);
  }
  b();
}
a();
```

```
function a() {
  var c = 10;
  function b() {
    console.log(c);
  }
  b();
}
a();
```

Garbage collector has removed
d. since it was of no
use.

→ Advantages:-

- Module Design patterns
- Caching
- Data Hiding
- SetTimeouts

→ Disadvantages:-

- Over consumption of memory.
- Memory leak
- Freeze browser.

Caching

```
function add(a) {
  return function test(b) {
    return function rest(c) {
      return a + b + c;
    };
  };
}

console.log(add(6)(7)(5));
```

```
function x() {
  function close(i) {
    setTimeout(function () {
      console.log(i);
    }, i * 1000);
  }
  for (var i = 1; i ≤ 5; i++) {
    close(i);
  }
  console.log("Namaste Javascript");
}

x();
```

```

function a() {
  console.log("Function Statement / Function Declaration");
}
a();

var b = function () {
  console.log("Function Expression");
}
b();

// function () {
// anonymous function
// used when functions are used as values
// }

var b = function xyz() {
  console.log("Named Function");
}
b();

var b = function(param1, param2) {
  console.log(param1, param2);
}
var arg1 = 10, arg2 = 10;
b(arg1, arg2);

var b = function(param1) {
  console.log(param1, "First Class Function / Citizens");
}

b(function(){}

```

```

const button = document.getElementById("clickMe");

function attachEventListener() {
  let count = 0;
  button.addEventListener("click", function xyz() {
    console.log("Button clicked", ++count);
  });
}

attachEventListener();

```

html body.vsc-initialized button#clickMe

Styles Computed Layout Event Listeners DOM Breakpoints Properties Accessibility

Ancestors All Framework listeners

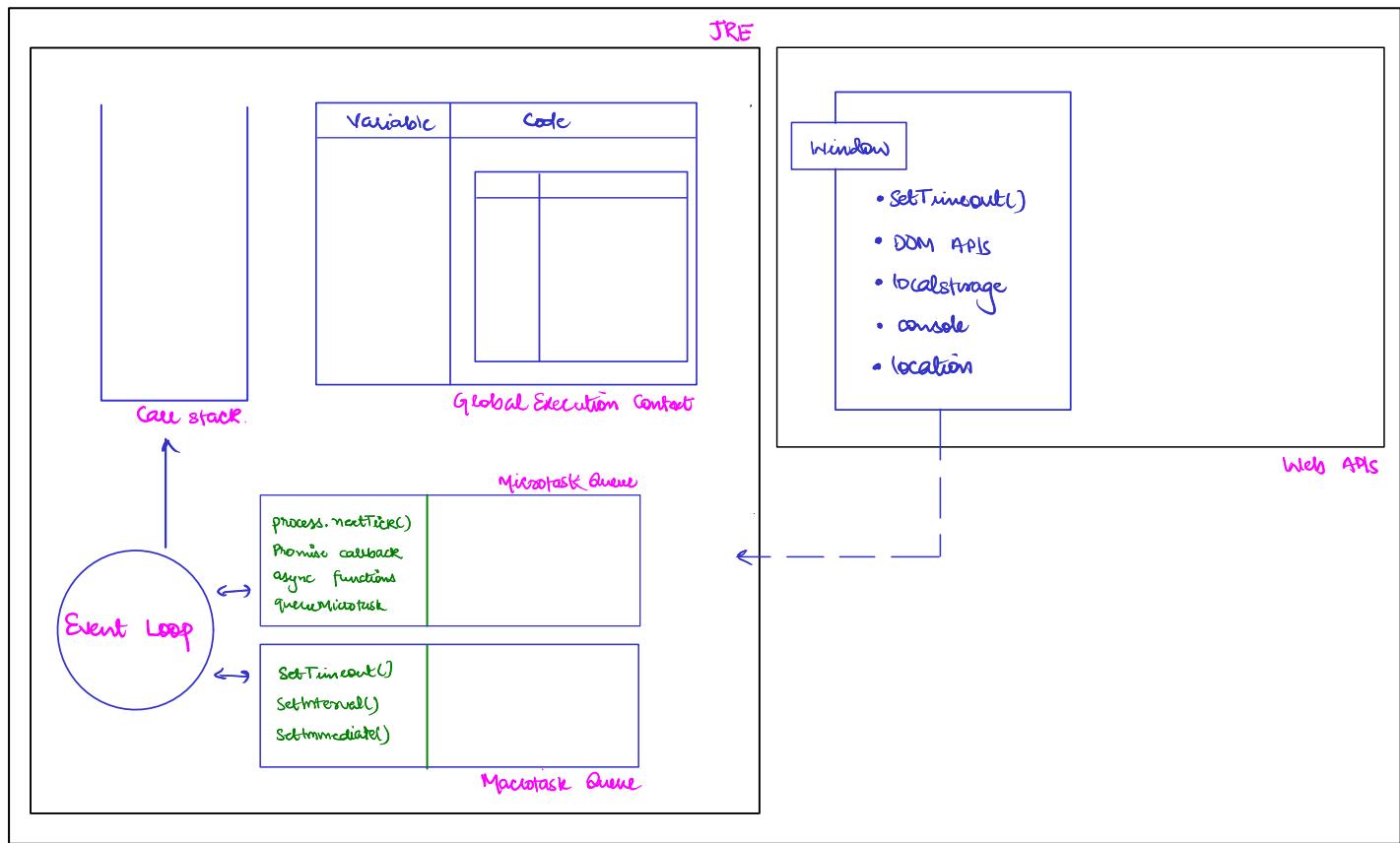
▼ click

▼ button#clickMe Remove index.js:5

- useCapture: false
- passive: false
- once: false

▼ handler: f xyz()

- ▼ [[Scopes]]: Scopes[3]
 - 2: Global {window: Window, self: Window, document: document, name: "", location: Location}
 - 1: Script {button: button#clickMe}
 - 0: Closure (attachEventListener)
- [[Prototype]]: Object
- count: 0
- [[Prototype]]: f ()
- [[FunctionLocation]]: <unknown>
- prototype: {}
- name: "xyz"
- length: 0
- caller: null
- arguments: null



Browser