



Javascript has 3 types of scopes:

- \* Global ✓
- \* Function ✓
- \* Block ✓

Function

Global

Execution  
Context  
Also called  
as Global scope  
& fn scope.

→ G  
var apple = "apple"

if (true) {

→ G  
var apple = "mango";

console.log(apple); ←

}

apple

mango

```
console.log(apple);
```

← Mango

What is the o/p & why?

```
var apple = "apple";
```

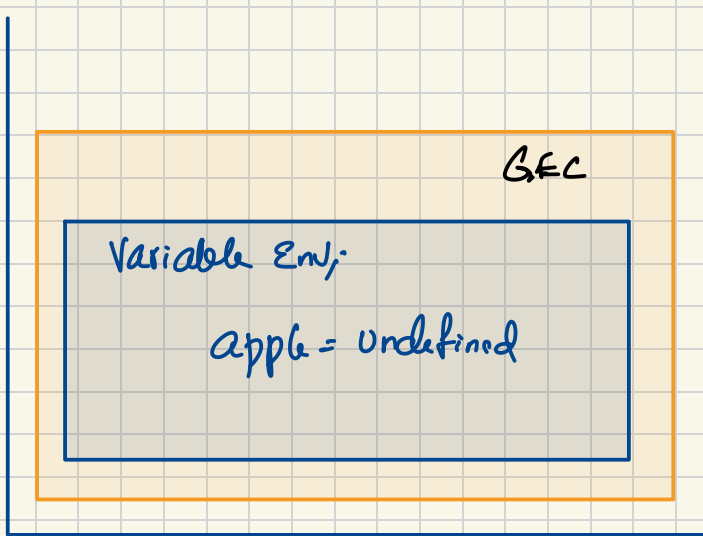
```
if (true) {
```

```
  let apple = "mango";
```

```
  console.log(apple);    mango
```

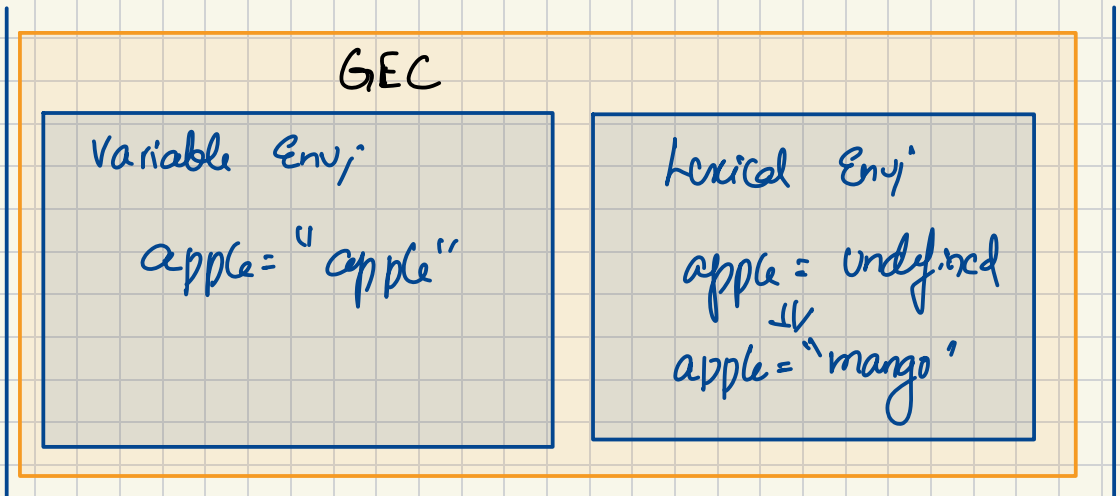
```
}
```

```
console.log(apple);    apple
```



The second *apple* is not declared.

Nested compilation:



```
var apple = "Global apple";
```

```
let banana = "Global banana";
```

```
{
```

```
  let banana = "block banana";
```

```
  var grape = "Global Grape";
```

```
  let orange = "block orange";
```

```
  console.log(apple); // Global apple
```

```
  console.log(banana); // block banana
```

```
}
```

```
console.log(apple); // Global apple
```

```
console.log(grape); // Global grape
```

```
console.log(orange); // not defined.
```

Lets understand the working of the above code:-

CALL STACK

Stack of variables

