

variables

funcⁿ

Scopes

visibility

variables

functions

↳ Every thing inside your code, will be used in one of the following 2 ways →

1) Either it will be getting some value assigned to it

2) Or some value will be retrieved from it.

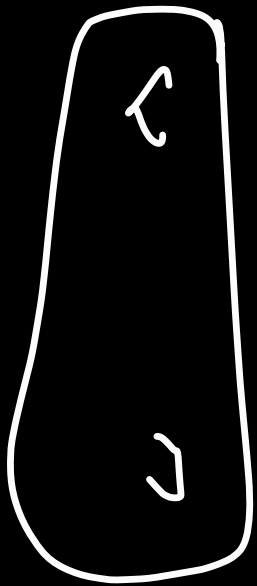
→ Do you think JS is compiled? Do you think JS is interpreted?

Types of Scope

1) Global

2) function

3) block



```

1  var teacher = "Sanket";
2  function fun() {
3      var teacher = "Pulkit";
4      console.log(teacher);
5  }
6  function gun() {
7      var student = "Sarthak";
8      console.log(student);
9  }
10
11  fun();
12  gun();
13  console.log(teacher);

```

Handwritten annotations on the code:

- Line 1: `var teacher = "Sanket";` → global
- Line 2: `function fun() {` → scope of fun
- Line 3: `var teacher = "Pulkit";` → global
- Line 6: `function gun() {` → scope of gun
- Line 7: `var student = "Sarthak";` → scope of gun

Sanket → teacher

every time we see a formal declaration, we think about the scope in parsing phase.

- 1) parsing → scope resolution
- 2) Execution

```
1 var teacher = "Sanket";
2 function fun() {
3   var teacher = "Pulkit";
4   content = "JS";
5   console.log(teacher);
6 }
7 function gun() {
8   var student = "Sarthak";
9   console.log(student);
10 }
11
12 fun();
13 gun();
14 console.log(teacher);
15 console.log(content);
16
17
18
```

global → [Sanket]
fun → [Pulkit]
global → [Sarthak]
[JS]
content

autoglobals

→ parsing
execution

Pulkit
Sarthak
Sanket
JS

```
1 var teacher = "Sanket";
2 function fun() {
3     var teacher = "Pulkit";
4     content = "JS";
5     console.log(teacher);
6 }
7 function gun() {
8     var student = "Sarthak";
9     console.log(student);
10 }
11 console.log(content);
12 fun();
13 gun();
14 console.log(teacher);
15
```

global

Sanket
+ env

fun

global

gun

