Scopes

What do we mean by the word Scopes ??

→ In simple words, scope is simply where to look for things. What are we looking for ?? We are looking for variables & functions

→ We have more or less 3 type of scopes in JS →

1. Global
2. function
3. block

Global scope

→ if a variable is present in a global scope, then it is accessible everywhere in the JS file.

But how do we define a variable in global scope ?? there are many ways → one of the way is to declare / define variable outside any function or a block.

function scope

↳ In a function, the visibility of a variable/func is just inside the outer function.

```
function fun() {
        var x =10;
}
```

} here x is just accessible inside fun or we can say it is local to fun.

Block Scope

In JS we can use a pair of curly braces to declare a block.

if else block
while block
for block
same block.

Now if a variable/func^n is only accessible/visible inside a block then it will be having block scope.

let, var, const

var → whenever we use a var anywhere inside a function, the variable gets function scope. if we use it outside a function, no matter if it is enclosed in a block or not, it will give the variable global scope.

let → whenever we initialize a variable with let, it always gets the scope of enclosing Block.

if we declare, let outside any block, then it doesnot get complete global scope

let doesn't allow redeclaration, but var does allow it.

if you initialise a variable with let outside any block, then also it will not become accessible completely in the global scope, but if you do with var, it will be.

const → it also has block scope, (same as let) the only difference is it doesn't allow reassignment.

```
const y = 10
y = 22;     ✗ error
```

How JS, parses the code ??

So we know that JS is not interpreted, it definetely is hybrid i.e compiled + interpreted.

So whenever we try to execute a JS code, JS first parses the whole code, in this parsing phase it assigns scopes to variables/funcⁿ. Once done, then it reads the code & executes it.

Every variable in your code will be used in one of the following ways →

1) either it will be getting a value assigned i.e. it is used as a target

$x = 10$

2) or it will be used to retrieve a value i.e it will be used as source.

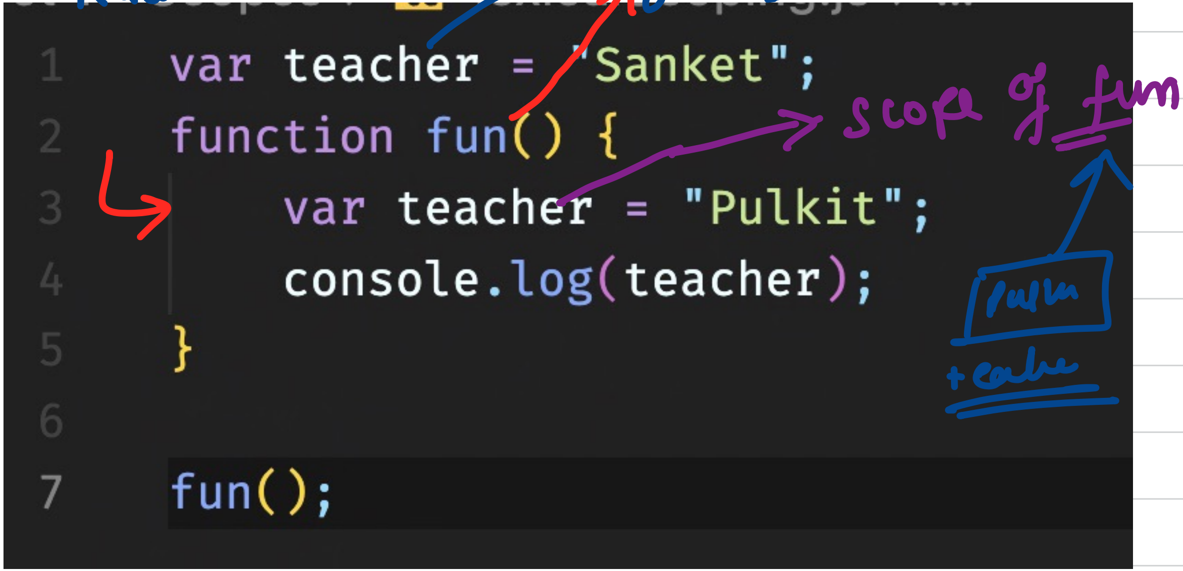
$y = 10 + x$

```
console.log(x)
```

what JS does is, it will start the parsing phase. Outside every thing it maintains global Scope,

but the moment it goes inside a function it starts maintaing Scope of that func^n also.

Sanket
teacher → global scope



```
var teacher = "Sanket";
function fun() {
    var teacher = "Pulkit";
    console.log(teacher);
}

fun();
```

→ scope of fun =

Pulkit
teacher

Phase 1 → parsing

↓

we will just do scope resolution

phase 2 → execution

whenever we declare a variable using var/let/const it is a formal declaration.

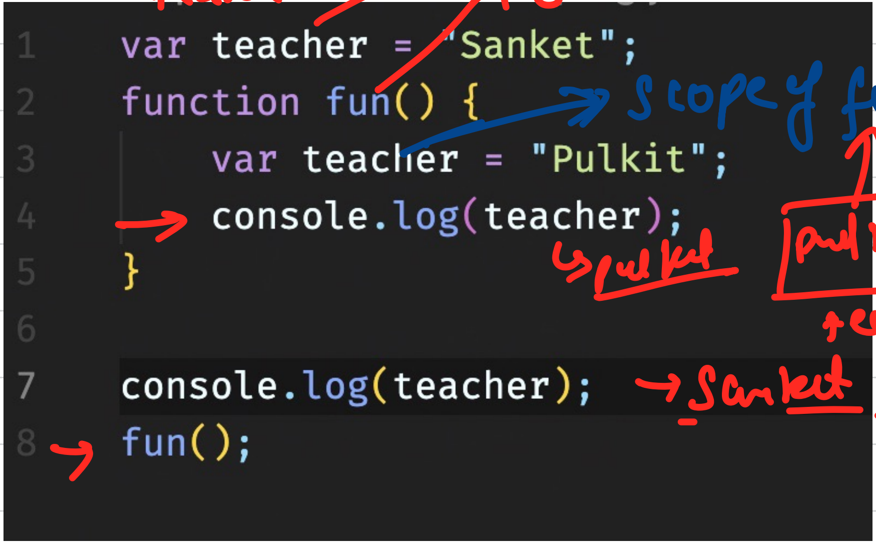
or initialising a func^n is also formal declaration.

In the parsing phase JS looks for formal declarations only.

In this parsing phase we only allocate scope, not values of the variable.

The moment we go in a func^n, we maintain a new scope as well i.e. func^n scope.

$\boxed{\text{Sanket}}$ → global
teacher



```
var teacher = "Sanket";
function fun() {
    var teacher = "Pulkit";
    console.log(teacher);
}

console.log(teacher);
fun();
```

→ scope of fun

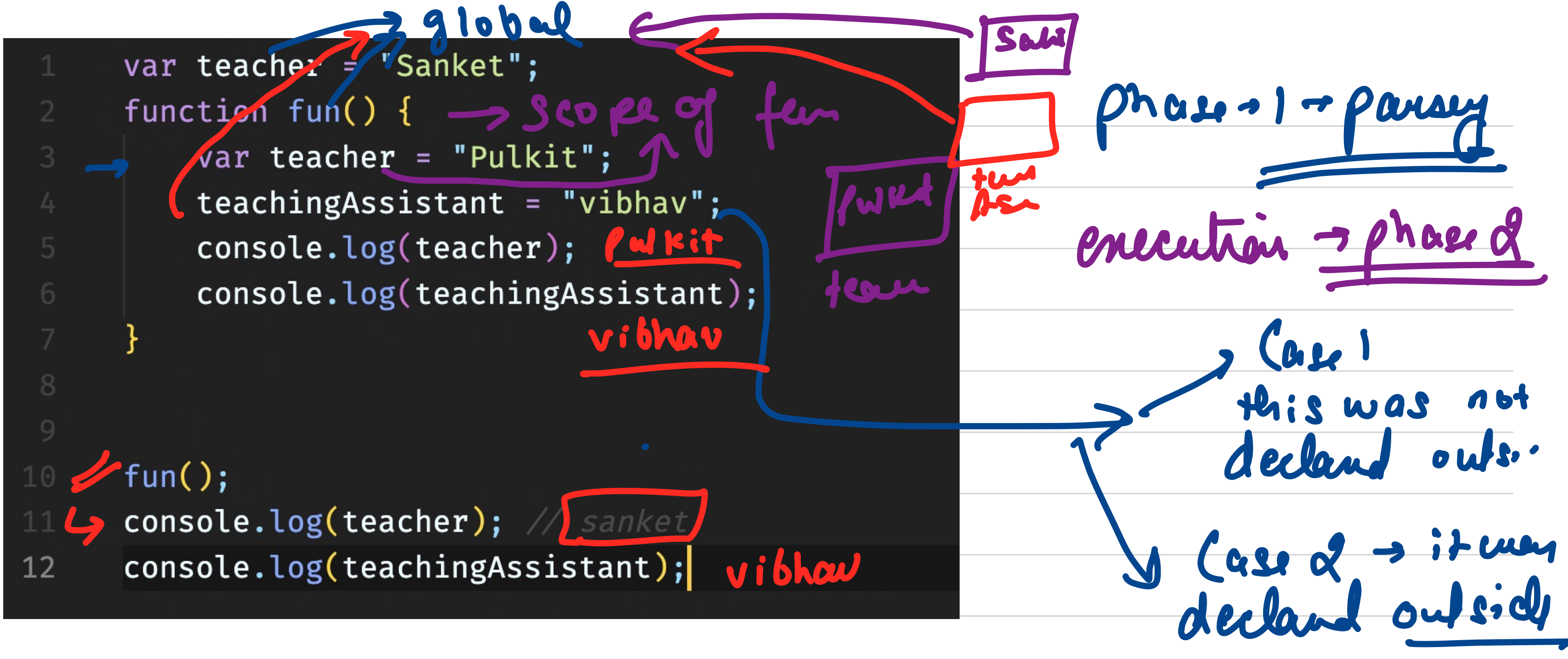
→ pulkit

$\boxed{\text{pulkit}}$
teacher

→ Sanket

phase-1 → Parsing

phase-2 → execution



```
var teacher = "Sanket";
function fun() {
    var teacher = "Pulkit";
    teachingAssistant = "vibhav";
    console.log(teacher);
    console.log(teachingAssistant);
}


fun();
console.log(teacher); // sanket
console.log(teachingAssistant);
```

Once inside a scope, we don't know about scope of a variable we check the outer scopes one by one.

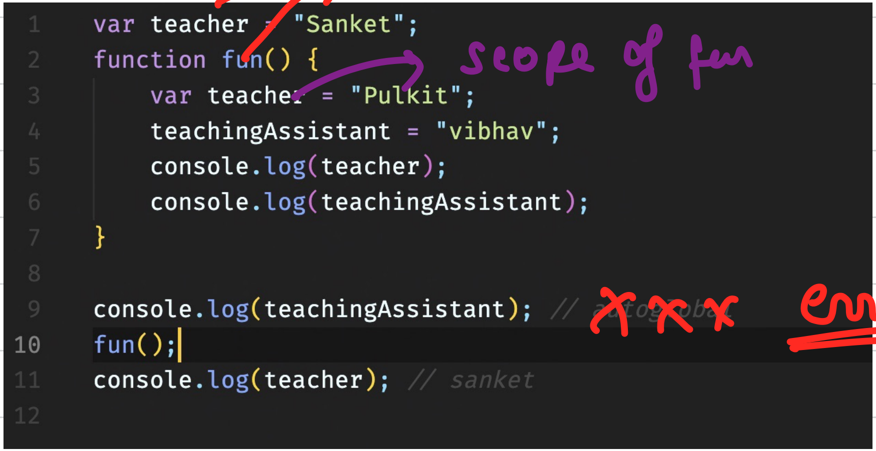
Auto globals → In JS if we keep on searching scope of a variable in outer scopes & no where find it, we automatically consider it in global **scope**. This happens during execution **phase**.

→ autoglobal only works with target reference & not **source**.

global

Sanket

teacher



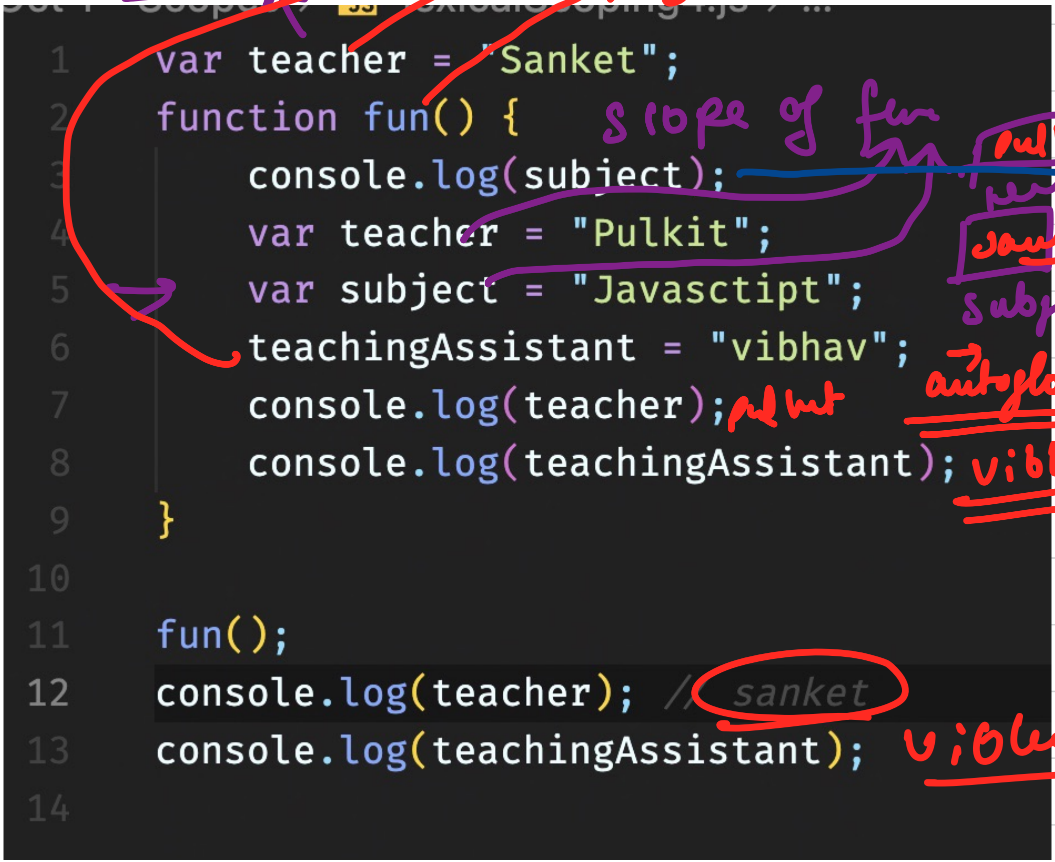
```
var teacher = "Sanket";
function fun() {
    var teacher = "Pulkit";
    teachingAssistant = "vibhav";
    console.log(teacher);
    console.log(teachingAssistant);
}

console.log(teachingAssistant); // autoglobal
fun();
console.log(teacher); // sanket
```

scope of fun

xxx

error



```
var teacher = "Sanket";
function fun() {
    console.log(subject);
    var teacher = "Pulkit";
    var subject = "Javasctipt";
    teachingAssistant = "vibhav";
    console.log(teacher);
    console.log(teachingAssistant);
}

fun();
console.log(teacher); // sanket
console.log(teachingAssistant);
```

Sanket

global

scope of fun

Pulkit

Pulkit

Javas

subject

autoglobal

Pulkit

vibhav

vibhav

phase → 1

phase - 2

undefined

var x;

undefined