SCOPE

- *We declare variable to store different data types. To declare a variable we use the key word var, let and const. A variable can be declared at different scope.*
- *Variables scopes can be*
  - *Glocal*
  - *Local*
- *Variable can be declared globally or locally scope.*
- *Anything declared without let, var or const is scoped at global level.*

WINDOW GLOBAL OBJECT

```
a = 'JavaScript' // declaring a variable without let or const make it available in window object and this found anywhere
b = 10 // this is a global scope variable and found in the window object
 console.log(a, b) // accessible
```

- *Without using console.log() open your browser and check, you will see the value of a and b if you write a or b on the browser. That means a and b are already available in the window.*

AM
DEV

GLOBAL SCOPE

- *A globally declared variable can be accessed every where in the same file.*

```
let a = 'JavaScript' // is a global scope it will be found anywhere in this file
let b = 10 // is a global scope it will be found anywhere in this file
function fn() {
  console.log(a, b) // JavaScript 10
  if (true) {
    let a = 'Python'
    let b = 100
    console.log(a, b) // Python 100
  }
 console.log(a, b) // JavaScript 10
}
fn()
console.log(a, b) // JavaScript 10
```

LOCAL SCOPE

AM DEV

- *A variable declared as local can be accessed only in certain block code.*
  - *Block Scope*
  - *Function Scope*

```
let a = 'JavaScript' // is a global scope it will be found anywhere in this file
let b = 10 // is a global scope it will be found anywhere in this file
// Function scope
function fn() {
    console.log(a, b) // JavaScript 10, accessible
    let value = false // block scope
    if (true) {
        // we can access from the function and outside
        // the function but variables declared inside
        // the  if will not be accessed outside the if block
        let a = 'Python'
        let b = 20
        let c = 30
        let d = 40
        value = !value
        console.log(a, b, c, value) // Python 20 30 true
        }
    // we can not access c,d because c and d's scope is only the if block
    console.log(a, b, value) // JavaScript 10 true
}
fn()
console.log(a, b) // JavaScript 10
```

AM
DEV

VAR

- *A variable declared with var only scoped to function but variable declared with let or const is block scope(function block, if block, loop block, etc). Block in JavaScript is a code in between two curly brackets ({}).*

```
function fn() {
    var a = 2
    console.log(a)  // 2
}
console.log(gravity) // Uncaught ReferenceError:
 // a is not defined
if (true){
  var a = 2
  console.log(a) // 2
}
console.log(a) // 2
for(var i = 0; i < 3; i++){
console.log(i) // 0, 1, 2
}
console.log(i) // 3
```

LET/CONST

AM DEV

- *In ES6 and above there is let and const, so you will not suffer from the sneakiness of var. When we use let our variable is block scoped and it will not infect other parts of our code. The scope let and const are the same. The difference is only reassigning. We can not change or reassign the value of the const variable*

```
function fn() {
    // you can use let or const, but year of birth is
    constant I prefer to use const
    const birthYear = 1995
    console.log(birthYear) // 1995
}
 // console.log(birthYear), Uncaught
//ReferenceError: birthYear is not defined
if (true){
    const birthYear = 1996
    console.log(birthYear) // 1996
}
// console.log(birthYear), Uncaught
// ReferenceError: birthYear is not defined
for(let i = 0; i < 3; i++){
    console.log(i) // 0, 1, 2
}
// console.log(i), Uncaught ReferenceError: i is not
defined
```