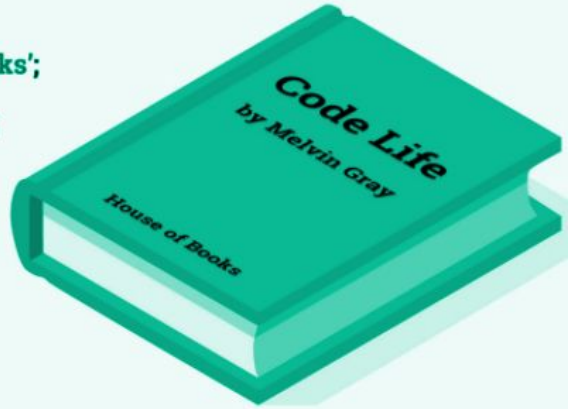
skill academy

Javascript Variables

```
var title = 'Code Life';
let publisher = 'House of Books';
const author = 'Melvin Gray';
```



Javascript Variables

- **Variables** are the data storing containers.
- Values in the variable may vary.
- Javascript variables must have **unique names**.
- The keywords **var**, **let** and **const** are used to declare the variables.
- Same variable should not be redeclared twice.
- Javascript is a untyped language i.e a variable can hold any data type values.

```
var money;
var name;
```

Javascript Variables

- Variables can be declared in multiple lines or even in single line with only one **var** keyword as below
- The variables can be initialized at the time of variable creation or at any point when you need that variable.

```
var money, name;
```

Types of Variables

Global Variable:

- The Global variable can be defined anywhere in the JS code.

```
var data=200;//gloabal variable declaration
function a(){
document.writeln(data);
}
```

Local Variable:

- The Local variable will be visible only within a particular function where it is defined.
- Parameters of a function are always local to that particular function.

```
function b(){
var data=50;//local variable declaration
document.writeln(data);
}
```

Declaring Variables using ‘let’ keyword

- If we try to **Redeclare global** variables with the **let** keyword, we will get a **syntax error**.

```
let x = "John Doe";

let x = 0;

// SyntaxError: 'x' has already been declared
```

- After ES6, Javascript keywords **‘let’** and **‘const’** provide **block scope**.

```
{
  let x = 2;
}
// x can NOT be used here
```

This variable cannot be accessed outside the block

Declaring Variables using ‘let’ keyword

- Variables defined with **let** must be Declared before use.
- If we declare with **‘let’** keyword, the value of the variables can **change**.
- Redeclaring Variables globally and local simultaneously with let keyword won’t impose a problem
- Redeclaring a variable inside a block won’t actually redeclare the variable outside the block

```
let x = 10;
// Here x is 10

{
let x = 2;
// Here x is 2
}

// Here x is 10
```

Declaring Variables using ‘var’ keyword

- We can **Redeclare** variables with the **var** keyword.

```
var x = "John Doe";

var x = 0;
```

- Variables declared with **var** keyword don’t have **block scope**.

```
{
  var x = 2;
}
// x CAN be used here
```

This variable can be accessed outside the block.

Declaring Variables using ‘var’ keyword

- Redeclaring Variables globally and local simultaneously with **var** keyword will impose a problem.
- Redeclaring a variable inside a block will actually redeclare the variable outside the block.

```
var x = 10;
// Here x is 10

{
var x = 2;
// Here x is 2
}

// Here x is 2
```

Declaring Variables using ‘const’ keyword

- We **cannot Redeclare** the variables defined with **const** keyword in the same scope. But, we can redeclare the **const** keyword variables in another block or scope.

```
const x = 2;       // Allowed

{
  const x = 3;   // Allowed
}

{
  const x = 4;   // Allowed
}
```

Declaring Variables using ‘const’ keyword

- We cannot **reassign** the variables defined with **const** keyword. It must be assigned at the time of declaration itself.

```
const PI = 3.141592653589793;
PI = 3.14;      // This will give an error
PI = PI + 10;   // This will also give an error
```

- Always variables that remain unchanged will be declared with the **const** keyword.

- Variables defined with **const** have Block Scope like **let** declaration.

```
const x = 10;
// Here x is 10

{
const x = 2;
// Here x is 2
}

// Here x is 10
```

Declaring Variables using ‘const’ keyword

- **const** keyword is mostly used to declare
  - New Array
  - New Object
  - New Function
  - New RegExp

```
// You can create a const object:
const car = {type:"Fiat", model:"500", color:"white"};

// You can change a property:
car.color = "red";
```

- **const** keyword defines a constant reference to a value but not a constant value.

- We cannot **reassign** a constant value, constant array or constant object but we can **change the elements** of a constant **array** or change the **properties** of a constant **object**.