JS_Basics-S1-Variables

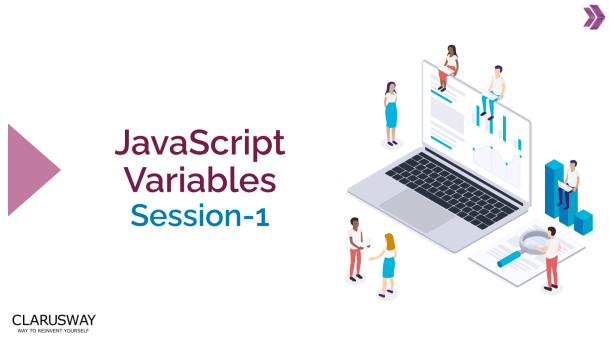
variables
Training Clarusway
Pear Deck - August 6, 2022 at 8:57AM

Part 1 - Summary

Use this space to summarize your thoughts on the lesson

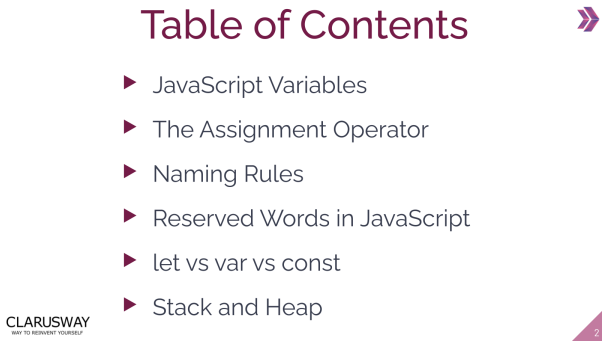
Part 2 - Responses

Slide 1



Use this space to take notes:

Slide 2



Use this space to take notes:

Slide 3

Link(s) on this slide:

- https://create.kahoot.it/details/2-variables/f108964e-1be0-4f3a-a9ae-9ded0a17434f

Use this space to take notes:

Slide 4


Use this space to take notes:

Slide 5	Your Response
	

Use this space to take notes:

Slide 6

What is Variable?

Variables are used to store for data values

CLARUSWAY
WAY TO REINVENT YOURSELF

6

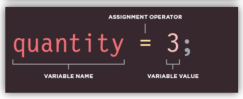
Use this space to take notes:

Slide 7

What is Variable?

```
var myNumber = 3;
```

In this example,
* var is a variable keyword.
* myNumber is a variable.
* 3 is a value.



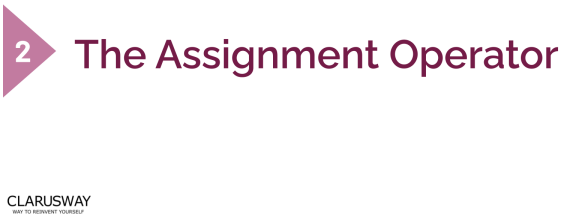
Warning ! :JavaScript is case sensitive. This means that the variables myNumber, mynumber or MYNUMBER are not same variables. All of them are different variables.

CLARUSWAY
WAY TO REINVENT YOURSELF

7

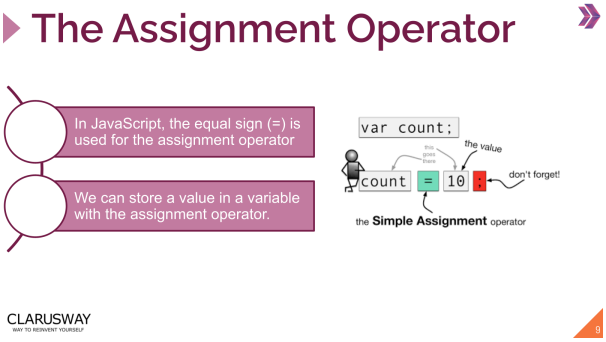
Use this space to take notes:

Slide 8



Use this space to take notes:

Slide 9



Use this space to take notes:

Slide 10

The Assignment Operator

Assignment always goes from right to left

```
var x=5;
var y=7;
y=x;
```

In this example;
x=5
y=5

CLARUSWAY

Use this space to take notes:

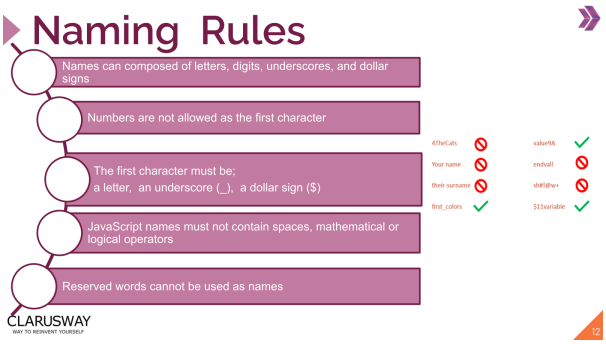
Slide 11

3 Naming Rules

CLARUSWAY

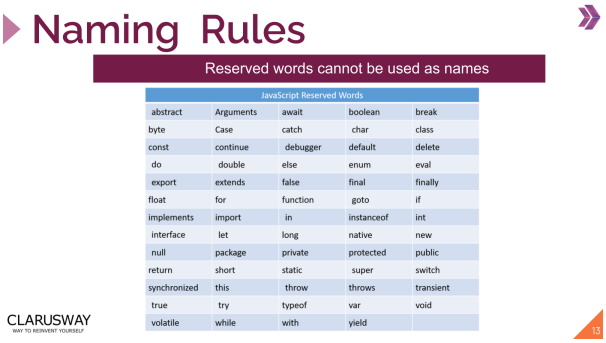
Use this space to take notes:

Slide 12



Use this space to take notes:

Slide 13

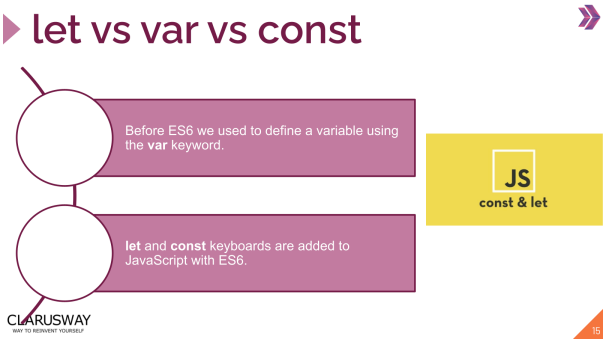


Use this space to take notes:

Slide 14

Use this space to take notes:

Slide 15



Use this space to take notes:

Slide 16

Scope

- Three types of scope:
  - Global scope
  - Function scope
  - Block scope
- Global scope
  Outside any function
  Variables can be accessed from ***anywhere in the program***

CLARUSWAY
WAY TO REINVENT YOURSELF

16

Use this space to take notes:

Slide 17

Scope
Function Scope

- Variables defined anywhere *inside a function* are *local* to that function
- Can be used anywhere inside that function
- Cannot be used outside that function

```
// code here can NOT use myNumber

function myFunction() {
  var myNumber = 42;
  //code here CAN use myNumber
}

// code here can NOT use myNumber
```

CLARUSWAY
WAY TO REINVENT YOURSELF

17

Use this space to take notes:

Slide 18

Scope
Block Scope

- To limit a variable to its block inside the function, use *let*

```
function fn(num){
  if (num > 5){
    var newNum = 5;
  }
  // newNum CAN be accessed here
}
```

```
function fn(num){
  if (num > 5){
    let newNum = 5;
  }
  // newNum can NOT be accessed here
}
```

CLARUSWAY
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 19

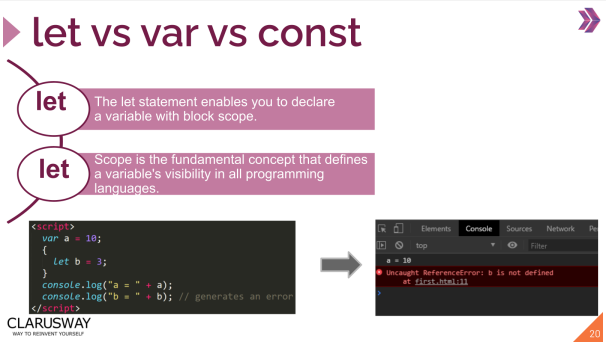
let vs var

- At global and function scopes, *let* and *var* work *almost* the same
- *var* supports redeclaration, while *let* does not
- Both support re-assignment. Use *const* to disallow it
- *let* is more like regular variables in other languages
  Preferred over *var*

CLARUSWAY
WAY TO REINVENT YOURSELF

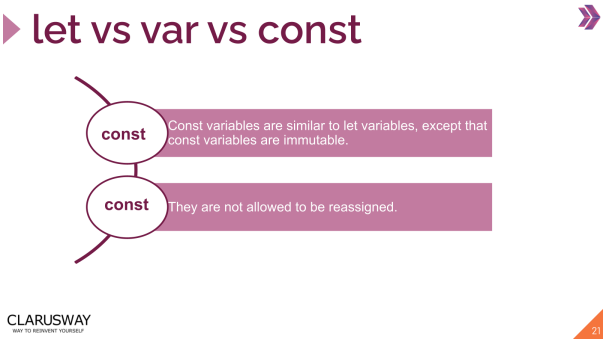
Use this space to take notes:

Slide 20



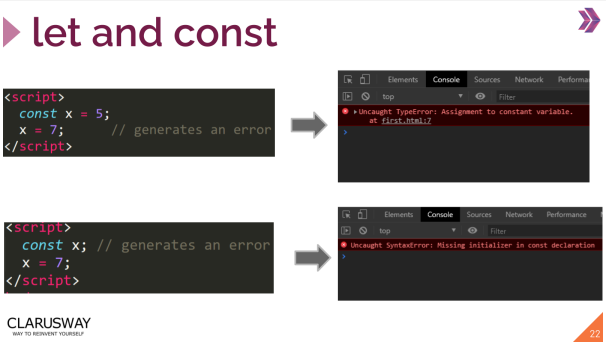
Use this space to take notes:

Slide 21



Use this space to take notes:

Slide 22



Use this space to take notes:

Slide 23

Use this space to take notes:

Slide 24

Stack and Heap

Variables in JavaScript (and most other programming languages) are stored in two places: stack and heap.

- A stack is usually a continuous region of memory allocating local context for each executing function.
- Heap is a much larger region storing everything allocated dynamically.
- This separation is useful

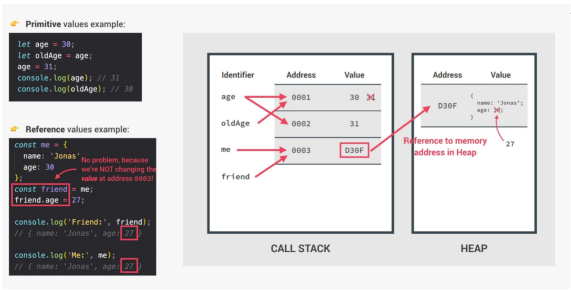
  Stack is more protected and faster, no need for dynamic garbage collection.

CLARUSWAY
WAY TO REINVENT YOURSELF

24

Use this space to take notes:

Slide 25



CLARUSWAY
WAY TO REINVENT YOURSELF

Source: Jonas Schmedtmann, The Complete JavaScript Course 2022: From Zero to Expert!

25

Use this space to take notes:

Slide 26

Use this space to take notes: