



JavaScript Variables

Javascript Session-2



Table of Contents



- ▶ JavaScript Variables
- ▶ The Assignment Operator
- ▶ Naming Rules
- ▶ Reserved Words in JavaScript
- ▶ Let and Const



**Play
Kahoot!**



1 What is Variable?





What is Variables?

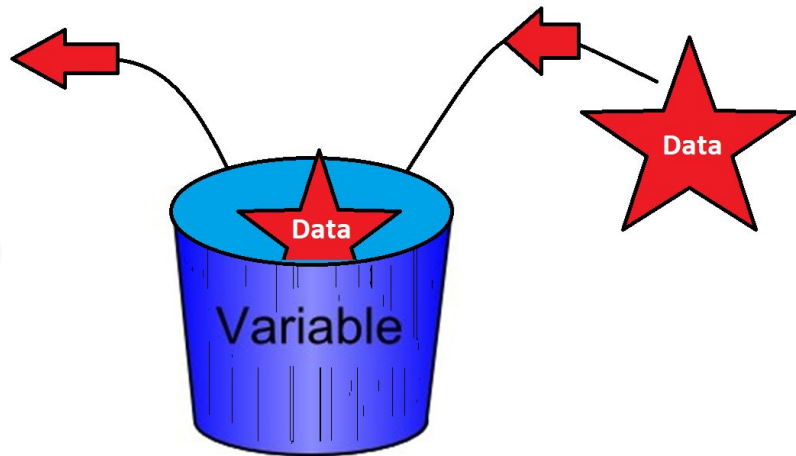


Students, write your response!



What is Variable?

Variables are used to store for data values





What is Variable?

```
var myNumber = 3;
```

In this example,

- * var is a variable keyword.
- * myNumber is a variable.
- * 3 is a value.

```
var quantity;
```

Diagram illustrating the components of the variable declaration `var quantity;`:

- `var` is labeled as the **VARIABLE KEYWORD**.
- `quantity` is labeled as the **VARIABLE NAME**.

```
quantity = 3;
```

Diagram illustrating the components of the assignment statement `quantity = 3;`:

- `quantity` is labeled as the **VARIABLE NAME**.
- `=` is labeled as the **ASSIGNMENT OPERATOR**.
- `3` is labeled as the **VARIABLE 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.



2

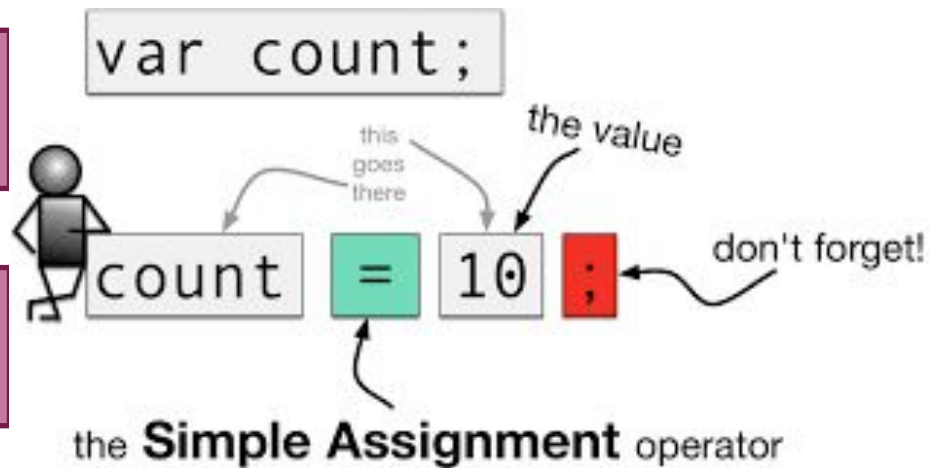
The Assignment Operator

The Assignment Operator



In JavaScript, the equal sign (=) is used for the assignment operator

We can store a value in a variable with the assignment operator.





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





3

Naming Rules

KNOW THE
RULES!





Naming Rules

Names can be composed of letters, digits, underscores, and dollar signs

Numbers are not allowed as the first character

The first character must be;
a letter, an underscore (_), a dollar sign (\$)

JavaScript names must not contain spaces, mathematical or logical operators

Reserved words cannot be used as names

4TheCats



Your name



their-surname



first_colors



value9A



endval!



sh#!@w+



\$11variable





Naming Rules

Reserved words cannot be used as names

JavaScript Reserved Words				
abstract	Arguments	await	boolean	break
byte	Case	catch	char	class
const	continue	debugger	default	delete
do	double	else	enum	eval
export	extends	false	final	finally
float	for	function	goto	if
implements	import	in	instanceof	int
interface	let	long	native	new
null	package	private	protected	public
return	short	static	super	switch
synchronized	this	throw	throws	transient
true	try	typeof	var	void
volatile	while	with	yield	



4 *let and const*

JavaScript

ES6



JS



Let vs Var vs Const



let and const

Before ES6 we used to define a variable using the **var** keyword.

let and **const** keywords are added to JavaScript with ES6.



const & let



let and const

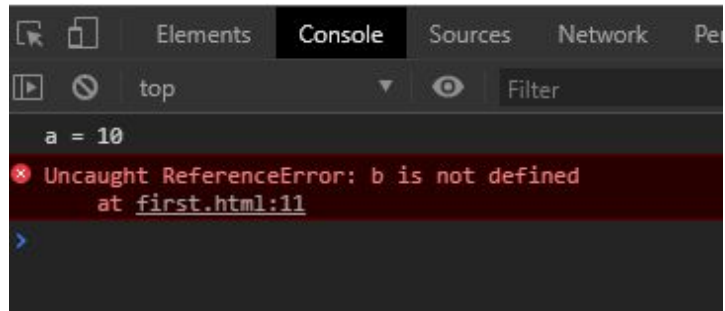
let

The let statement enables you to declare a variable with block scope.

let

Scope is the fundamental concept that defines a variable's visibility in all programming languages.

```
<script>
  var a = 10;
  {
    let b = 3;
  }
  console.log("a = " + a);
  console.log("b = " + b); // generates an error
</script>
```





let and const

const

Const variables are similar to let variables, except that const variables are immutable.

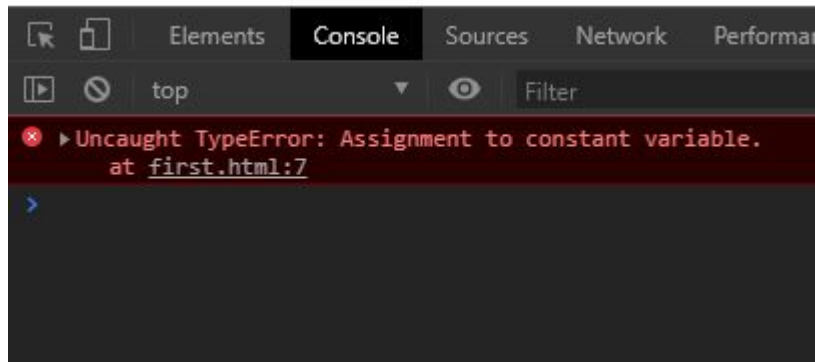
const

They are not allowed to be reassigned.

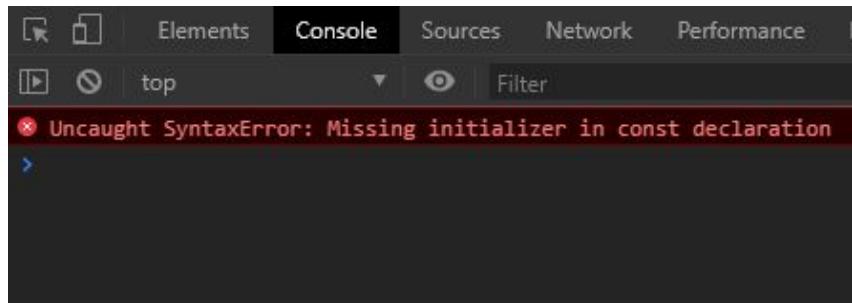
let and const



```
<script>
  const x = 5;
  x = 7;      // generates an error
</script>
```



```
<script>
  const x; // generates an error
  x = 7;
</script>
```





THANKS!

Any questions?

