

JAVASCRIPT-CORE-S3-Operators

Cohort 10 JS Session - 3
Training Clarusway
Pear Deck - January 22, 2022 at 0:57AM

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

Table of Contents

- ▶ Operators
- ▶ Arithmetic Operators
- ▶ Assignment Operators
- ▶ Comparison Operators
- ▶ Logical Operators
- ▶ Nullish Coalescing Operator

CLARUSWAY
WAY TO REFINEMENT YOURSELF

Use this space to take notes:

Slide 3

Your Response



Did you finish Javascript Core pre-class material?

YES **NO**

CLARUSWAY
Way to Refinement Yourself

Peer Deck Interactive Slide

Use this space to take notes:

Slide 4

Play Kahoot!

CLARUSWAY
WAY TO REINVENT YOURSELF

Link(s) on this slide:

- <https://create.kahoot.it/details/5-operators/18b23ba0-0384-45c3-8ae9-067a960b8e27>

Use this space to take notes:

Slide 5

1 Operators



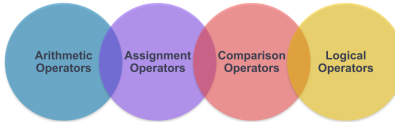
CLARUSWAY
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 6

Operators

- Let's take a simple $3 + 2$ phrase equals 5. Number 3 and 2 are operands and '+' is the operator.
- Expressions rely on operators to create a single value from one or more values
- JavaScript supports the operators of the following types



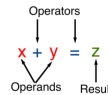
CLARUSWAY
WAY TO REFINEMENT, YOURSELF

8

Use this space to take notes:

Slide 7

2 Arithmetic Operators



CLARUSWAY
WAY TO REFINEMENT, YOURSELF

Use this space to take notes:

Slide 8

► Arithmetic Operators



Arithmetic operators execute arithmetic functions on numbers (literals or variables)

NAME	OPERATOR	EXAMPLE	RESULT
ADDITION	+	10+5	15
SUBTRACTION	-	10-5	5
DIVISION	/	10/5	2
MULTIPLICATION	*	10*5	50
INCREMENT	++	var i=10; i++;	11
DECREMENT	--	var i=10; i--;	9
MODULUS	%	10%3	1

CLARUSWAY
WAY TO REFINEMENT YOURSELF

8

Use this space to take notes:

Slide 9

► Arithmetic Operators

```
<script>
  var a = 20;
  var b = 3;
  var c = a * b;
  console.log("c : " + c);
</script>
```



```
top
c : 60
>
```

CLARUSWAY
WAY TO REFINEMENT YOURSELF

```
<script>
  var d = 25;
  var e = 7;
  var f = d % e;
  console.log("f : " + f);
</script>
```



```
top
f : 4
>
```

9

Use this space to take notes:

Slide 10



3 Assignment Operators

CLARUSWAY
WAY TO REFINEMENT YOURSELF

Use this space to take notes:

Slide 11

► Assignment Operators



Assignment operators assign values to JavaScript variables

OPERATOR	EXAMPLE	MEANING
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y

CLARUSWAY
WAY TO REFINEMENT YOURSELF



Use this space to take notes:

Slide 12

► Assignment Operators

```
<script>
  var a = 7, result = 21;
  console.log("result += a : " + (result += a));
  console.log("result -= a : " + (result -= a));
  console.log("result *= a : " + (result *= a));
  console.log("result /= a : " + (result /= a));
  console.log("result %= a : " + (result %= a));
</script>
```



```
top
result += a : 28
result -= a : 21
result *= a : 147
result /= a : 21
result %= a : 0
```

CLARUSWAY
WAY TO REFINEMENT YOURSELF

12

Use this space to take notes:

Slide 13

4 Comparison Operators

> >= < <=
== != === !==

CLARUSWAY
WAY TO REFINEMENT YOURSELF

Use this space to take notes:

Slide 14

Comparison Operators

Comparison operators are used to determine equality or difference between variables or values in logical statements

All comparison operators return Boolean (true or false)

Operator	Description	Example
==	Equality	3==2 //false
!=	Inequality	3!=2 //true
===	Identity/Strict Equality (equal and of same type)	3===2 //false
!==	Non-identity/Strict Inequality	3!==2 //true
>	Greater than	3>2//true
>=	Greater than or equal	3>=2//true
<	Less than	3<2//false
<=	Less than or equal	3<=2//false

The most notable difference between this operator and the equality (==) operator is that if the operands are of different types, the === operator attempts to convert them to the same type before comparing.

CLARUSWAY
WAY TO REFINEMENT

14

Use this space to take notes:

Slide 15

Comparison Operators

```
<script>
var x = 5;
console.log("(x == 8) : " + (x == 8));
console.log("x == '5' : " + (x == "5"));
console.log("x === '5' : " + (x === "5"));
console.log("x != 8 : " + (x != 8));
console.log("x > 8 : " + (x > 8));
console.log("x < '6' : " + (x < "6"));
console.log("x >= 4 : " + (x >= 4));
console.log("x <= 9 : " + (x <= 9));
</script>
```

```
(x == 8) : false
x == '5' : true
x === '5' : false
x != 8 : true
x > 8 : false
x < '6' : true
x >= 4 : true
x <= 9 : true
```

CLARUSWAY
WAY TO REFINEMENT

15

Use this space to take notes:

Slide 16

5 Logical Operators

NOT(!)
AND(&&)
OR(||)

CLARUSWAY
WAY TO RESEMBLE YOURSELF

Use this space to take notes:

Slide 17

► Logical Operators



Logical operators, also known as Boolean Operators, are used to determine the logic between variables or values and return true or false.

Seeing as $x = 3$ and $y = 2$, logical operators are explained in the table below:

NAME	OPERATOR	DESCRIPTION	EXAMPLE
and	&&	Returns true, if both operands are true	$(x < 5 \ \&\& \ y > 3) // \text{false}$
or		Returns true, if one of the operands are true	$(x == 3 \ \ y == 5) // \text{true}$
not	!	Returns true, if the operand is false, and false, if the operand is true	$!(x == y) // \text{true}$

CLARUSWAY
WAY TO RESEMBLE YOURSELF

17

Use this space to take notes:

Slide 18

► Logical Operators

```
<script>
var x = 6, y = 3;
console.log("1. (x < 10 && y > 1) : " + (x < 10 && y > 1));
console.log("2. (x < 10 && y < 1) : " + (x < 10 && y < 1));
console.log("3. (x == 5 || y == 5) : " + (x == 5 || y == 5));
console.log("4. (x == 6 || y == 0) : " + (x == 6 || y == 0));
console.log("5. (x == 0 || y == 3) : " + (x == 0 || y == 3));
console.log("6. (x == 6 || y == 3) : " + (x == 6 || y == 3));
console.log("7. !(x == y) : " + !(x == y));
console.log("8. !(x > y) : " + !(x > y));
</script>
```

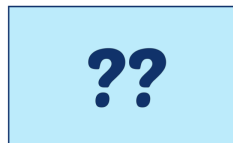
```
1. (x < 10 && y > 1) : true
2. (x < 10 && y < 1) : false
3. (x == 5 || y == 5): false
4. (x == 6 || y == 0): true
5. (x == 0 || y == 3): true
6. (x == 6 || y == 3): true
7. !(x == y) : true
8. !(x > y) : false
```

CLARUSWAY
WAY TO REFINEMENT YOURSELF

Use this space to take notes:

Slide 19

6 ► Nullish Coalescing Operator



CLARUSWAY
WAY TO REFINEMENT YOURSELF

Use this space to take notes:

Slide 20

Nullish Coalescing Operator

The nullish coalescing operator (??) is a logical operator that returns its right-hand side operand when its left-hand side operand is null or undefined, and otherwise returns its left-hand side operand.

Contrary to the logical OR (||) operator, the left operand is returned if it is a falsy value that is not null or undefined.

```
// src/js
const foo = null ?? "default string";
console.log(foo); // expected output: "default string"

const bar = 0 ?? 42;
console.log(bar); // expected output: 0

const nullValue = null;
const emptyStr = ""; // falsy
const number = 42;

const valA = nullValue ?? "default for A";
const valB = emptyStr ?? "default for B";
const valC = number ?? 0;

console.log(valA); // default for A
console.log(valB); // default for B
console.log(valC); // 0
```

```
"default string"
0
"default for A"
<empty string>
42
```

CLARUSWAY
WAY TO REFINEMENT YOURSELF

20

Use this space to take notes:

Slide 21

THANKS!

Any questions?



CLARUSWAY
WAY TO REFINEMENT YOURSELF

21

Use this space to take notes: