

JS_Basics-S2-Operators

operators

Training Clarusway


Pear Deck - August 11, 2022 at 7:41PM

Part 1 - Summary


Use this space to summarize your thoughts on the lesson

Part 2 - Responses

Slide 1



Operators
Session-3



CLARUSWAY
HELP TO IMPROVE YOURSELF

»

Use this space to take notes:

Slide 2

Table of Contents

- ▶ Operators
- ▶ Assignment Operators
- ▶ Arithmetic 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
https://www.clarusway.com/peer-deck

Use this space to take notes:

Slide 4



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



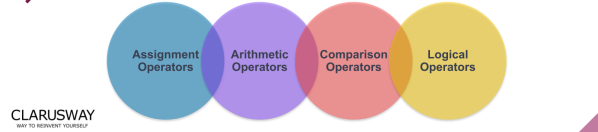
CLARUSWAY
WAY TO REPRESENT YOURSELF

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



Use this space to take notes:

Slide 7

2 ► Assignment Operators

CLARUSWAY
WAY TO REFINEMENT YOURSELF

Use this space to take notes:

Slide 8

► 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
**=	x **= y	x = x ** y

CLARUSWAY
WAY TO MASTER YOURSELF

Use this space to take notes:

Slide 9

► Assignment Operators

```
<script>
let a = 6, 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}');
console.log('result **= a -> ${result **= a}');
</script>
```



Elements Console

top Filter

```
result += a -> 27
result -= a -> 21
result *= a -> 126
result /= a -> 21
result %= a -> 3
result **= a -> 729
```

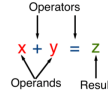
CLARUSWAY
WAY TO MASTER YOURSELF

Use this space to take notes:

Slide 10



3 Arithmetic Operators



CLARUSWAY
WAY TO RESEMBLE YOURSELF

Use this space to take notes:

Slide 11



► 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
EXPONENTIATION	**	10 ** 3	1000
MODULUS	%	10 % 4	2

CLARUSWAY
WAY TO RESEMBLE YOURSELF



Use this space to take notes:

Slide 12

► Postfix Prefix Operators



Postfix prefix operators add or subtract one from their operand

POSTFIX			
NAME	OPERATOR	EXAMPLE	RESULT
INCREMENT	++	let a = 10; let b = a++;	a = 11 b = 10
DECREMENT	--	let a = 10; let b = a--;	a = 9 b = 10

PREFIX			
NAME	OPERATOR	EXAMPLE	RESULT
INCREMENT	++	let a = 10; let b = ++a;	a = 11 b = 11
DECREMENT	--	let a = 10; let b = --a;	a = 9 b = 9

CLARUSWAY
WAY TO MASTER YOURSELF

12

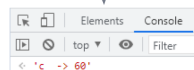
Use this space to take notes:

Slide 13

► Arithmetic Operators



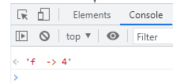
```
<script>
let a = 20, b = 3, c = a * b;
console.log(`c -> ${c}`);
</script>
```



CLARUSWAY
WAY TO MASTER YOURSELF



```
<script>
let d = 25, e = 7, f = d % e;
console.log(`f -> ${f}`);
</script>
```



13

Use this space to take notes:

4 Comparison Operators



Use this space to take notes:

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 == '3' // true
!=	Inequality	3 != '3' // false
===	Strict equality (equal and of same type)	3 === '3' // false
!==	Strict inequality	3 !== '3' // 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

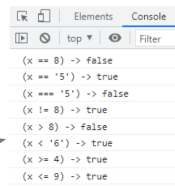
The most notable difference between this operator and the equality (==) operator is that if the operands are of different types, the == operator attempts to convert them to the same type before comparing.

Use this space to take notes:

Slide 16

► Comparison Operators

```
<script>
let 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>
```



Elements Console

top Filter

(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 YOURSELF

Use this space to take notes:

Slide 17

5 ► Logical Operators

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

CLARUSWAY
WAY TO REFINEMENT YOURSELF

Use this space to take notes:

Slide 18

► 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	<code>&&</code>	Returns true, if both operands are true	<code>(x < 5 && y > 3) // false</code>
Or	<code> </code>	Returns true, if either of operands are true	<code>(x == 3 y == 3) // true</code>
Not	<code>!</code>	Simply toggles the operand value true/false	<code>!(x == y) // true</code>

CLARUSWAY
WAY TO MASTER YOURSELF



Use this space to take notes:

Slide 19

► Logical Operators



```
<script>
let 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>
```

CLARUSWAY
WAY TO MASTER YOURSELF

```
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
```



Use this space to take notes:

Slide 20

6 Nullish Coalescing Operator



CLARUSWAY
WAY TO MASTER YOURSELF

Use this space to take notes:

Slide 21

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**.

```
<script>
const nullAndString = null ?? "Hello World!";
console.log( nullAndString => ${nullAndString} );

const result = 0 ?? 42;
console.log( result => ${result} );

const nullValue = null;
const emptyText = ""; // falsy
const randomNumber = 42;

const resultA = nullValue ?? "Hello World!";
const resultB = emptyText ?? "Hello World!";
const resultC = randomNumber ?? 0;

console.log( resultA = ${resultA} );
console.log( resultB = ${resultB} );
console.log( resultC = ${resultC} );
</script>
```

CLARUSWAY
WAY TO MASTER YOURSELF

```
Elements Console Sc
top Filter
nullAndString -> Hello World!
result -> 0
resultA = Hello World!
resultC = 42
```

21

Use this space to take notes:



7 Other Operators

Type checking in JS:
typeof and **instanceof**
operators

typeof

instanceof

CLARUSWAY
WAY TO REFINEMENT YOURSELF

Use this space to take notes:



Other Operators



typeof operator is used to determine the type of given variable's value.



instanceof operator used to determine object type of given object, such as arrays, maps etc.

```
<script>

const arr = [1, 2, 3];
console.log(typeof arr); //object
console.log(arr instanceof Array); //true

</script>
```

CLARUSWAY
WAY TO REFINEMENT YOURSELF



Use this space to take notes:

THANKS!

Any questions?



CLARUSWAY
WAY TO RESIDENT YOURSELF

Use this space to take notes: