

Web Engineering Front-end PT. 3

# 4. JavaScript: Comparison and Logical Operators





# Revision



## Data Types



Data types are important when operating on variables. The main data types we will be focusing on today are strings, numbers and Booleans.

## Contents



1. Assignment Operators
2. Data Type Conversion
3. Comparison Operators
4. Logical Operators

## 4.1 Assignment Operators



## What you know so far



You can assign values to variables by using the = assignment operator.

e.g.

```
var x = 10;
```

## More assignment operators



Aside from `=`, there are other assignment operators that you can use. They allow you to perform arithmetic operations at the same time as assigning the variable.

## More assignment operators



Operator	is the same as	If $x = 10, y = 5$
$x += y$	$x = x + y$	$x = 15$
$x -= y$	$x = x - y$	$x = 5$
$x *= y$	$x = x * y$	$x = 50$
$x /= y$	$x = x / y$	$x = 2$
$x %= y$	$x = x \% y$	$x = 0$

## More assignment operators



e.g.

```
var x = 10;
```

```
x += 5;
```

```
x; // 15
```

## Assignment operator exercise



```
var x = 100;
```

```
x -= 5;
```

```
x %= 10;
```

```
x += 10 + 30;
```

What does x equal?

## Practice: Assignment operators



Spend some time practicing assignment operators. If you have time, you can check out the [full list](#) of assignment operators and see which ones are useful to you.

## 4.2 Data Type Conversions



## Type conversion methods



You can convert variables into different data types while keeping the same value using the String(), Number() and Boolean() methods.

## Type conversion methods



e.g.

```
var x = 10
```

```
typeof(x); // "Number"
```

```
x = String(x);
```

```
typeof(x); // "String"
```

## Implicit conversions



JavaScript will also sometimes do this automatically.

e.g.

`x = 5 + "7"`

`x; // 12 (Not NaN)`

This is called implicit conversion. Can you guess what the previous kind is called?

## More type conversion examples



Do you know what this will output?

Number("0x30");

## Practice: Type conversions



Spend some time practicing type conversions. Try using the same values with different data types and see how the answer changes.

e.g.

$x = 2000 + 21$  vs  $x = 2000 + "21"$

## 4.3 Comparison Operators



## What are comparison operators?



Comparison operators are used to compare values and either return True or False (a Boolean).

## Comparison operators



Operator	Description	If $x = 10$
<code>==</code>	Equal value	<code>x == 10; // True</code> <code>x == 5; // False</code> <code>x == "10"; // True</code>
<code>===</code>	Equal value and equal type	<code>x === 10; // True</code> <code>x === 5; // False</code> <code>x === "10"; // False</code>
<code>!=</code>	Not equal value	<code>x != 10; // False</code> <code>x != 5; // True</code> <code>x != "10"; // False</code>
<code>!==</code>	Not equal value or equal type	<code>x !== 10; // False</code> <code>x !== 5; // True</code> <code>x !== "10"; // True</code>

# Comparison operators



Operator	Description	If $x = 10$
>	Greater than	$x > 5$ ; // True $x > 10$ ; // False $x > 15$ ; // False
$\geq$	Greater or equal to	$x \geq 5$ ; // True $x \geq 10$ ; // True $x \geq 15$ ; // False
<	Less than	$x < 5$ ; // False $x < 10$ ; // False $x < 15$ ; // True
$\leq$	Less than or equal to	$x \leq 5$ ; // False $x \leq 10$ ; // True $x \leq 15$ ; // True

## Comparing strings



Have you ever noticed how the words in a dictionary are ordered?

## Dictionary order



quiet; repose; facility: *v.t.* to free from pain, anxiety, or trouble; give rest or relief.

**easel** (ē'z'l), *n.* a wooden frame or tripod for supporting a canvas, blackboard, &c.

**easement** (ēz'ment), *n.* that which gives ease or relief; a right of accommodation in another's land, or a right of passage.

**east** (ēst), *n.* that part of the heavens where the sun is seen to rise; one of the four cardinal points; the eastern part of the earth: *adj.* com-

## Dictionary order



1. Compare the first character of both words
2. If the first character of the first word is less than the first character of the second word, then the first word is lesser.
3. If the first characters are equal, then move on to the second characters.
4. Repeat until the end of either word.
5. If both strings are the same length, then they are equal. Otherwise, the shorter word is lesser.

## Dictionary order



```
"easel" < "easement"; // True
```

e = e

a = a

s = s

e = e

l < m

quiet; repose; facility: *v.t.* to free from pain, anxiety, or trouble; give rest or relief.

**easel** (ē'z'l), *n.* a wooden frame or tripod for supporting a canvas, blackboard, &c.

**easement** (ēz'ment), *n.* that which gives ease or relief; a right of accommodation in another's land, or a right of passage.

**east** (ēst), *n.* that part of the heavens where the sun is seen to rise; one of the four cardinal points; the eastern part of the earth: *adj.* com-



KRYSTAL INSTITUTE  
DIGITAL ECONOMY SKILL TECHNOLOGY

## Comparing strings



```
"a" < "b" // True
```

```
"A" < "a"; // True
```

```
"Apple" < "Apply"; // True
```

```
"App" < "Apple"; //True
```

## Comparing different data types



When comparing a string to a number, JavaScript will convert the string into a number. A non-numeric string will convert to NaN, which will always return False. An empty string converts to 0.

## Comparing different data types



```
10 < 20; // True  
10 < "20"; // True  
10 < "Twenty"; // False
```

## Practice: Comparison operators



Spend some time practicing comparison operators. Make sure to try different data types.

## 4.4 Logical Operators



## What are logical operators?



Logical operators can be used to connect two or more expressions so that the result depends on the combined expression.

## AND



AND (`&&`) returns True only if all the operators in the combined expression are True.

## AND



```
var x = 5; var y = 10;  
  
x < y && typeof(x) = "number"; // True (True AND True)  
  
x > y && typeof(x) = "number"; // False (False AND True)  
  
x > y && typeof(x) = "string"; // False (False AND False)
```

## AND



```
var x = 5; var y = 10; var z = 20;
```

```
x < y && x < z && y < z; // True (True AND True AND True)
```

```
x < y && x < z && y > z; // False (True AND True AND False)
```

OR (||) returns True if any of the operators in the combined expression is True.

## AND



```
var x = 5; var y = 10;  
  
x < y || typeof(x) = "number"; // True (True OR True)  
  
x > y || typeof(x) = "number"; // True (False OR True)  
  
x > y || typeof(x) = "string"; // False (False OR False)
```

## AND



```
var x = 5; var y = 10; var z = 20;  
  
x > y || x > z || y < z; // True (False OR False OR True)  
  
x > y || x > z || y > z; // False (False OR False OR False)
```

## NOT



NOT(!) reverses the Boolean result of a comparison operator.

## NOT



```
var x = 5; var y = 10;  
  
!(x < y); // False  
  
!(x > y AND typeof(x) = "number"); // True
```

## Truth tables



AND

x	y	$x \&& y$
false	false	false
false	true	false
true	false	false
true	true	true

OR

x	y	$x    y$
false	false	false
false	true	true
true	false	true
true	true	true

NOT

x	$!x$
false	true
true	false

## Combining logical operators



You can combine logical operators using parentheses for more complex logical comparisons.

## Combining logical operators



```
var x = 5; var y = 10; var z = 20;
```

```
(x < y || y > z) && x < z; // True ((True OR False) AND True)
```

## Logical operators use case



For verifying user login details:

(username == correctUsername && password == correctPassword)

JavaScript: Comparison and Logical Operators

## Brainstorm: Logical operator use cases



Get into groups and try to come up with more uses for logical operators in your everyday life.



## Practice: Logical operators



Spend some time practicing logical operators with comparison operators.

As you get more comfortable, try forming more complex expressions by combining them.



# The End





Reference 1: JavaScript.info Comparisons <https://javascript.info/comparison>

Reference 2: Dictionary Page (Slides 23, 25) <https://olddesignshop.com/2015/04/easter-free-vintage-dictionary-page/>