

Data types

There are 8 basic data types in JavaScript.

- `number` for numbers of any kind: integer or floating-point, integers are limited by `±(253 - 1)`.
- `bigint` is for integer numbers of arbitrary length.
- `string` for strings. A string may have zero or more characters, there's no separate single-character type.
- `boolean` for `true` / `false`.
- `null` for unknown values – a standalone type that has a single value `null`.
- `undefined` for unassigned values – a standalone type that has a single value `undefined`.
- `object` for more complex data structures.
- `symbol` for unique identifiers.

The `typeof` operator allows us to see which type is stored in a variable.

- Two forms: `typeof x` or `typeof(x)`.
- Returns a string with the name of the type, like `"string"`.
- For `null` returns `"object"` – this is an error in the language, it's not actually an object.

```
typeof undefined // "undefined"

typeof 0 // "number"

typeof 10n // "bigint"

typeof true // "boolean"

typeof "foo" // "string"

typeof Symbol("id") // "symbol"

typeof Math // "object" (1)

typeof null // "object" (2)

typeof alert // "function" (3)
```

Tasks

String quotes

What is the output of the script?

```
let name = "Ilya";

alert( `hello ${1}` ); // ?

alert( `hello ${"name"}` ); // ?

alert( `hello ${name}` ); // ?
```

Math Object

In JavaScript the Math Object provides a lots of methods to work with numbers.

```
const PI = Math.PI

console.log(PI) // 3.141592653589793

// Rounding to the closest number
// if above .5 up if less 0.5 down rounding

console.log(Math.round(PI)) // 3 to round values to the nearest number

console.log(Math.round(9.81)) // 10

console.log(Math.floor(PI)) // 3 rounding down

console.log(Math.ceil(PI)) // 4 rounding up

console.log(Math.min(-5, 3, 20, 4, 5, 10)) // -5, returns the minimum value

console.log(Math.max(-5, 3, 20, 4, 5, 10)) // 20, returns the maximum value

const randNum = Math.random() // creates random number between 0 to 0.999999
console.log(randNum)

// Let us create random number between 0 to 10
```

```

const num = Math.floor(Math.random() * 11) // creates random number between 0 and 10
console.log(num)

//Absolute value
console.log(Math.abs(-10)) // 10

//Square root
console.log(Math.sqrt(100)) // 10

console.log(Math.sqrt(2)) // 1.4142135623730951

// Power
console.log(Math.pow(3, 2)) // 9

console.log(Math.E) // 2.718

// Logarithm
// Returns the natural logarithm with base E of x, Math.log(x)
console.log(Math.log(2)) // 0.6931471805599453
console.log(Math.log(10)) // 2.302585092994046

// Trigonometry
Math.sin(0)
Math.sin(60)

Math.cos(0)
Math.cos(60)

```

Strings

Strings are texts, which are under **single**, **double**, **back-tick** quote. To declare a string, we need a variable name, assignment operator, a value under a single quote, double quote, or backtick quote. Let's see some examples of strings:

String Concatenation

Connecting two or more strings together is called concatenation. Using the strings declared in the previous String section:

```

let fullName = firstName + space + lastName; // concatenation, merging two string together.
console.log(fullName);

```

String Methods

Everything in JavaScript is an object. A string is a primitive data type that means we can not modify it once it is created. The string object has many string methods. There are different string methods that can help us to work with strings.

1. *length*: The string *length* method returns the number of characters in a string included empty space.

1. *Accessing characters in a string*: We can access each character in a string using its index. In programming, counting starts from 0. The first index of the string is zero, and the last index is the length of the string minus one.

```
let string = 'JavaScript'
let firstLetter = string[0]

console.log(firstLetter)           // J

let secondLetter = string[1]       // a
let thirdLetter = string[2]
let lastLetter = string[9]

console.log(lastLetter)           // t

let lastIndex = string.length - 1

console.log(lastIndex)           // 9
console.log(string[lastIndex])    // t
```

- Length Method
- toLocaleLowerCase() Method
- toLocaleLowerCase() Method
- indexOf() Method
- search() Method

- slice() Method
- substring() Method
- substr() Method
- replace() Method
- includes() Method
- concat() Method
- charAt() Method
- charCodeAt() Method
- lastIndexOf() Method
- trim() Method
- match() Method
- split() Method
- toString() Method
- valueOf() Method

Checking Data Types

To check the data type of a certain variable we use the *typeof* method.

Changing Data Type (Casting)

- Casting: Converting one data type to another data type. We use *parseInt()*, *parseFloat()*, *Number()*, *+* *sign*, *str()* When we do arithmetic operations string numbers should be first converted to integer or float if not it returns an error.

String to Int

We can convert string number to a number. Any number inside a quote is a string number. An example of a string number: '10', '5', etc. We can convert string to number using the following methods:

- `parseInt()`
- `Number()`
- Plus sign(+)

Exercise: Level 1

1. Declare a variable named `challenge` and assign it to an initial value **'Full stack JS track '**.
2. Print the string on the browser console using **`console.log()`**
3. Print the **length** of the string on the browser console using `console.log()`
4. Change all the string characters to capital letters using **`toUpperCase()`** method
5. Change all the string characters to lowercase letters using **`toLowerCase()`** method
6. Cut (slice) out the first word of the string using **`substr()`** or **`substring()`** method
7. Slice out the phrase *Days Of JavaScript* from Full stack JS track .
8. Check if the string contains a word **Script** using **`includes()`** method
9. Split the **string** into an **array** using **`split()`** method
10. Split the string Full stack JS track at the space using **`split()`** method
11. 'Facebook, Google, Microsoft, Apple, IBM, Oracle, Amazon' **split** the string at the comma and change it to an array.
12. Change Full stackJS track to Full stackJS group using **`replace()`** method.
13. What is character at index 15 in 'Full stack JS track ' string?
Use **`charAt()`** method.
14. What is the character code of J in 'Full stack JS track ' string
using **`charCodeAt()`**
15. Use **`indexOf`** to determine the position of the first occurrence of **a** in Full stackJS track

16. Use **lastIndexOf** to determine the position of the last occurrence of **a** in Full stackJS track
17. Use **indexOf** to find the position of the first occurrence of the word **because** in the following sentence: '**You cannot end a sentence with because because because is a conjunction**'
18. Use **lastIndexOf** to find the position of the last occurrence of the word **because** in the following sentence: '**You cannot end a sentence with because because because is a conjunction**'
19. Use **search** to find the position of the first occurrence of the word **because** in the following sentence: '**You cannot end a sentence with because because because is a conjunction**'
20. Use **trim()** to remove any trailing whitespace at the beginning and the end of a string. E.g ' 30 Days Of JavaScript '.
21. Use **startsWith()** method with the string Full stackJS track and make the result true
22. Use **endsWith()** method with the string Full stackJS track and make the result true
23. Use **match()** method to find all the **a**'s in 30 Days Of JavaScript
24. Use **concat()** and merge 'full stack ' and 'JavaScript' to a single string, 'Full stack JS track'
25. Use **repeat()** method to print Full stackJS track 2 times

1. Using console.log() print out the following statement:

The quote 'There is no exercise better for the heart than reaching down and lifting people up.' by John Holmes teaches us to help one another.

2. Using console.log() print out the following quote by Mother Teresa:

"Love is not patronizing and charity isn't about pity, it is about love. Charity and love are the same -- with charity you give love, so don't just give money but reach out your hand instead."

3. Check if typeof '10' is exactly equal to 10. If not make it exactly equal.
4. Check if parseFloat('9.8') is equal to 10 if not make it exactly equal with 10.
5. Check if 'on' is found in both python and jargon
6. *I hope this course is not full of jargon.* Check if *jargon* is in the sentence.

7. Generate a random number between 0 and 100 inclusively.
8. Generate a random number between 50 and 100 inclusively.
9. Generate a random number between 0 and 255 inclusively.
10. Access the 'JavaScript' string characters using a random number.
11. Use `console.log()` and escape characters to print the following pattern.

Exercise