

JavaScript For Web

Week 2, Lecture 3 - JavaScript Fundamentals: String

Instructor: Jason Xu

Today's Overview

- Review: Functions and Strings
- Introduction to Objects
- String Methods
- **Tutorial:** Assignment 1, Exercise 5

Functions Review

Functions are reusable blocks of code.

```
let sum = function(a, b) {  
  return a + b  
};  
  
console.log(sum(1, 2)) // 3  
console.log(sum(2, 2)) // 4  
console.log(sum(100, 2)) // 102
```

This arrow function is a shorter form of function expression

```
let sum = (a, b) => a + b
```

and it's the same as:

```
let sum = (a, b) => {  
  return a + b  
}
```


Strings Review

A `string` in JavaScript must be surrounded by quotes. There are three different types of quotes:

```
let username = "Jason" //double quotes
let userId = '56498725' //single quotes
let phrase = `Hello, my name is ${username} and my ID is ${userId}.` //backticks
```

What are Objects

Think about in a real life, a car is an object. A car has **properties** like weight and color, and **methods** like start and stop:

Object	Properties	Methods
	<code>car.name = Fiat</code> <code>car.model = 500</code> <code>car.weight = 850kg</code> <code>car.color = white</code>	<code>car.start()</code> <code>car.drive()</code> <code>car.brake()</code> <code>car.stop()</code>

- All cars have the same **properties**, but the property **values** differ from car to car.
- All cars have the same **methods**, but the methods are performed **at different times**.

What are Objects continued

In JavaScript, objects are variables too. But objects can contain many **properties(values)** and **methods(functions)**. For example:

```
const myCar = {  
  type: "Fiat",  
  model: 500,  
  color: "white",  
  ...  
  start: function() {...},  
  drive: function() {...}  
  ...  
}
```

- We can access the properties or run the methods using `variable name` with a dot notation `.`:

```
console.log(myCar.type)  
myCar.start()
```

- You don't have to know too much details about the object at this moment. You just need to know what are **properties** and **methods**, and how we can access them using dot notation `.`.
- Just like this car object, **strings** also have **properties** and **methods**, today we will talk about those are commonly used.

Strings Properties and Methods

String Length

The length property returns the length of a string

```
let text = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"  
let length = text.length
```


String Method: `charAt()`

The `charAt()` method returns the character at a specified index (position) in a string

```
let text = "HELLO WORLD"  
let char = text.charAt(0)
```

- Index always start from `0`
- What is the last index of the string? `text.length - 1`

String Method: `slice()`

`slice()` extracts a **part** of a string and returns the extracted part in a **new string**.

- The method takes 2 parameters: start position, and end position (end not included).
- Slice out a portion of a string from position 7 to position 13:

```
let text = "Apple, Banana, Kiwi"  
let part = text.slice(7, 13)
```

- If you omit the second parameter, the method will slice out the rest of the string:

```
let text = "Apple, Banana, Kiwi"  
let part = text.slice(7)
```

- If a parameter is negative, the position is counted from the end of the string:

```
let text = "Apple, Banana, Kiwi"  
let part = text.slice(-12)
```

String `substr()`

`substr()` is similar to `slice()`.

- The difference is that the second parameter specifies the **length** of the extracted part.

```
let str = "Apple, Banana, Kiwi"
let part = str.substr(7, 6)
```

- If you omit the second parameter, `substr()` will slice out the rest of the string.

```
let str = "Apple, Banana, Kiwi"
let part = str.substr(7)
```

- If the first parameter is negative, the position counts from the end of the string.

```
let str = "Apple, Banana, Kiwi"
let part = str.substr(-4)
```

String: Converting to Upper and Lower Case

A string is converted to upper case with `toUpperCase()` :

```
let text1 = "Hello World!"  
let text2 = text1.toUpperCase()
```

A string is converted to lower case with `toLowerCase()` :

```
let text1 = "Hello World!"  
let text2 = text1.toLowerCase()
```

String: `trim()`

The `trim()` method removes whitespace from both sides of a string:

```
let text1 = "    Hello World!    "  
let text2 = text1.trim()
```

String `includes()`

The `includes()` method returns true if a string contains a specified value. Otherwise it returns `false`.

- Check if a string includes "world":

```
let text = "Hello world, welcome to the universe."  
text.includes("world")
```

- Check if a string includes "world". Start at position 12:

```
let text = "Hello world, welcome to the universe."  
text.includes("world", 12)
```

- `includes()` is case sensitive.
- `includes()` is an ES6 feature.

Summary

- Objects: properties and methods

String Properties and Methods

- `str.length`
- `str.charAt()`
- `str.slice()` and `str.substr()`
- `str.toUpperCase()` and `str.toLowerCase()`
- `str.trim()`
- `str.includes()`
- [Check more about strings on MDN](#)

Thank you