

# ES6 Features

1. ES6 Syntax
2. Arrow Function
3. Destructuring
4. Array methods
5. String methods
6. ES6 Modules
7. ES6 Classes
8. Promise , Async



index.js ×



javascript-tutorials-code &gt; index.js

```
1  /*
2  ES6 Features
3      1. ES6 Syntax
4          1.1 ES6 Variables -> var, let, const (scope, redeclaration, value assigning)
5          1.2 Template Literals,
6          1.3 Hoisting & strict mode
7          1.4 Default and Rest parameter
8          1.5 spread operator
9          1.6 for...of
10     2. Array functions
11     3. Destructuring
12         3.1 Array and Object Destructuring
13     .
14     .
```

TERMINAL

PROBLEMS

OUTPUT

DEBUG CONSOLE

1: powershell



PS C:\Users\Anisul Islam\Dropbox\full stack\front end\Javascript\JavaScript codes&gt; |



master\*



0 0 0

Ln 5, Col 31

Spaces: 4

UTF-8

CRLF

Babel JavaScript



Go Live



ESLint



Prettier



5:36 PM

12/15/2020



## 1. ES6 Syntax

1.1 ES6 Variables -> var, let, const (scope, redeclaration, value assigning)

1.2 Template Literals,

1.3 Hoisting & strict mode


1.4 Default and Rest parameter

1.5 spread operator

1.6 for...of


# ES6 Features

1. ES6 Syntax
2. Arrow Function
3. Destructuring
4. Array methods
5. String methods
6. ES6 Modules
7. ES6 Classes
8. Promise / Async

- 
- I. Traditional VS Arrow function
  - II. Arrow function with map and filter function

# ES6 Features

1. ES6 Syntax
2. Arrow Function
3. Destructuring
4. Array methods
5. String methods
6. ES6 Modules
7. ES6 Classes
8. Promise / Async

- 
- I. Array destructure
  - II. Object destructure

index.js

javascript-tutorials-code &gt; index.js

```
1 // array destructure
2 // swap variables
3 // object destructure
4 // nested object destructure
5 // destructuring function parameters
```


TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE

1: powershell, powershell

```
PS C:\Users\Anisul Islam\Dropbox\full stack\front end\Javascript\JavaScript code
s\javascript-tutorials-code>
```

# ES6 Features

1. ES6 Syntax
2. Arrow Function
3. Destructuring
4. Array methods
5. String methods
6. ES6 Modules
7. ES6 Classes
8. Promise / Async

- 
- I. `Array.find()`
  - II. `Array.findIndex()`

index.js ×

🔍 ▶ □ ...

javascript-tutorials-code &gt; index.js

```
1 // find(callback, value) return the value of the first element that pass certain condition
2
3
4
5 // findIndex(callback, value) return the index of the first element that pass certain condition
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE

1: powershell, powershell

+ □ 🗑 ⬆ ×

```
PS C:\Users\Anisul Islam\Dropbox\full stack\front end\Javascript\JavaScript code
ascrip\JavaScript codes\javascript-tutorials-code>
```

master\* 0: 11 0 0 0

Ln 2, Col 1 Spaces: 4 UTF-8 CRLF Babel JavaScript

Go Live

ESLint

Prettier

6:43 PM

12/19/2020



# ES6 Features

1. ES6 Syntax
2. Arrow Function
3. Destructuring
4. Array methods
5. String methods
6. ES6 Modules
7. ES6 Classes
8. Promise / Async



- I. `String.startsWith()`
- II. `String.endsWith()`
- III. `String.includes()`

index.js ×

🔍 ▶ □ ...

javascript-tutorials-code &gt; index.js

```
1 // startsWith(searchString, position) -> check a string starts with another string
2
3 // endsWith(searchString, length) -> check a string ends with another string
4
5 // includes(searchString, position) -> check if a string contains another string
6
7 // all these methods are case sensitive
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE

1: powershell, powershell

+ □ ✕

```
PS C:\Users\Anisul Islam\Dropbox\full stack\front end\Javascript\JavaScript code
s\javascript-tutorials-code>
```

# ES6 Features

1. ES6 Syntax
2. Arrow Function
3. Destructuring
4. Array methods
5. String methods
6. ES6 Modules
7. ES6 Classes
8. Promise / Async



- I. Export
- II. import



# ES6 Features

1. ES6 Syntax
2. Arrow Function
3. Destructuring
4. Array methods
5. String methods
6. ES6 Modules
7. ES6 Classes
8. Promise / Async



- I. Class
- II. Constructor
- III. Object
- IV. Set
- V. get



# ES6 Features

1. ES6 Syntax
2. Arrow Function
3. Destructuring
4. Array methods
5. String methods
6. ES6 Modules
7. ES6 Classes
8. Promise , Async



- I. Synchronous vs Asynchronous
- II. Callback, higher-order function
- III. Promise
- IV. Async await

- How to create a promise
- How to use a promise
- How to run multiple promise - `all()`
- `race()`
- Promise chaining

## API Calling in JS

```
1 // 4 ways to call api - XMLHttpRequest, fetch, axios, jquery
2
3 // fetch() has replaced XMLHttpRequest
4 // fetch() - global method for making HTTP Request
5 // 2 ways to call - then, async await
6
7 // + fetch() is easy to use compare to XMLHttpRequest
8 // + fetch() returns a promise
9 // - returned promise can only handle network error
10 // - does not support all the older browser
11
```

## API Calling in JS

5\_API\_CALLING > js-api-calling > scripts > axios.js

```
1 // 4 ways to call api - XMLHttpRequest, fetch, axios, jquery
2 // axios is a js library
3 // it helps to make request from browser (plain js/Vue/React/
  Angular), node.js
```

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21



# API Calling in JS

Elements Console

top Filter

Default levels 1 issue: 1

index.html M jquery.js U X

5\_API\_CALLING > js-api-calling > scripts > jquery.js

```
1 // 4 ways to call api - XMLHttpRequest,
  fetch, axios, jquery
2 // add jquery library cdn
3 // ajax - asynchronous javascript and xml
4 |
```

master\*

Spaces: 4

UTF-8

LF

JavaScript React

Port : 5500

Prettier



EXPLORER



index.html U

local-storage.js U X



JAVASCRIPT-TUTORIALS...

&gt; 1\_BASIC\_JS

&gt; 2\_DATA\_STRUCT...

&gt; 3\_JS HTML DOM

&gt; 4\_Browser\_BOM

&gt; 5\_Error\_Handling

&gt; 6\_ES6 FEATURES

&gt; 7\_API\_CALLING

&gt; 8\_JSON

9\_Storage

index.html U

local-stora... U

&gt; 11\_PROJECT



&gt; OUTLINE

&gt; TIMELINE

9\_Storage &gt; local-storage.js

```
1 // Web storage API -- allows us to store & read  
  data in browser  
2 // Web storage API -- localStorage, sessionStorage  
3  
4 // localStorage -- store, read, update and remove  
  data  
5 // no expiry date: data never gets lost even if  
  you close the browser  
6
```

