Web Framework Tutorial 00

JavaScript ES6

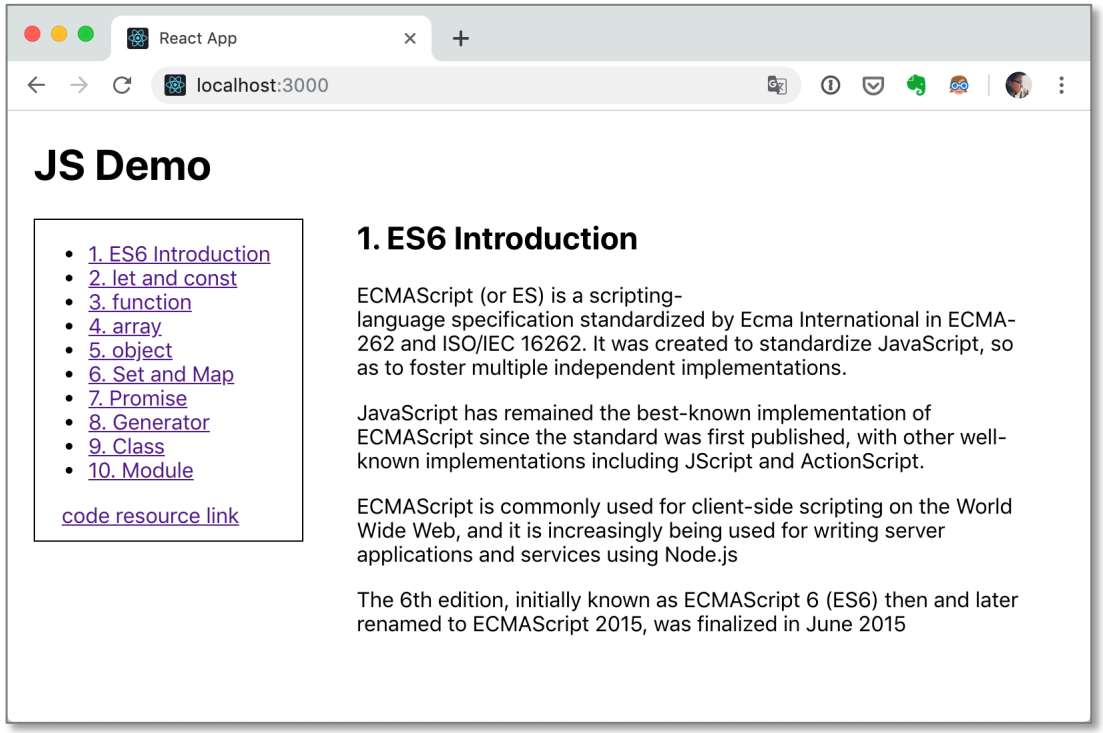
Danqing Shi

Resource: https://github.com/sdq/react-tutorial

CONTENTS

Code Setup

- git clone https://github.com/sdq/react-tutorial.git (or download on sharefolder)
- cd react-tutorial/js-demo/
- yarn (or npm install)
- yarn start (or npm install)



https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

01 Introduction to ECMAScript

ECMAScript (or ES) is a scripting-language specification standardized by Ecma International in ECMA-262 and ISO/IEC 16262. It was created to standardize JavaScript, so as to foster multiple independent implementations.

JavaScript has remained the best-known implementation of ECMAScript since the standard was first published, with other well-known implementations including JScript and ActionScript.

ECMAScript is commonly used for client-side scripting on the World Wide Web, and it is increasingly being used for writing server applications and services using Node.js.

The 6th edition, initially known as ECMAScript 6 (**ES6**) then and later renamed to ECMAScript 2015, was finalized in June 2015.

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

02 let and const

let: The ***let*** statement allows you to declare a variable with block scope.

```
{
  let a = 10;
  var b = 1;
}

a // ReferenceError: a is not defined.
b // 1
```

```
for (let i = 0; i < 10; i++) {
  // ...
}

console.log(i);
// ReferenceError: i is not defined
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

02 let and const

temporal dead zone

```
var tmp = 123;

if (true) {
  tmp = 'abc'; // ReferenceError
  let tmp;
}
```

Sometimes, dead zone is hard to figure out.

```
function bar(x = y, y = 2) {
  return [x, y];
}
```

```
function bar(x = 2, y = x) {
  return [x, y];
}
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

02 let and const

The **const** statement allows you to declare a constant (a JavaScript variable with a constant value).

```
const PI = 3.1415;
PI // 3.1415

PI = 3;
// TypeError: Assignment to constant variable.
```

```
if (true) {
  const MAX = 5;
}

MAX // Uncaught ReferenceError: MAX is not defined
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

03 function

default parameter

```
function log(x, y = 'World') {
  console.log(x, y);
}

log('Hello') // Hello World
log('Hello', 'China') // Hello China
log('Hello', '') // Hello
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

03 function

rest parameter

```
function add(...values) {
  let sum = 0;

  for (var val of values) {
    sum += val;
  }

  return sum;
}

add(2, 5, 3) // 10
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

03 function

Arrow function

```
var f = v => v;

// 等同于
var f = function (v) {
  return v;
};
```

```
var f = () => 5;
// 等同于
var f = function () { return 5 };

var sum = (num1, num2) => num1 + num2;
// 等同于
var sum = function(num1, num2) {
  return num1 + num2;
};
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

04 array

Spread: ...

```
console.log(...[1, 2, 3])
// 1 2 3

console.log(1, ...[2, 3, 4], 5)
// 1 2 3 4 5

[...document.querySelectorAll('div')]
// [<div>, <div>, <div>]
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

04 array

Concat Array

```
const arr1 = ['a', 'b'];
const arr2 = ['c'];
const arr3 = ['d', 'e'];

// ES5 的合并数组
arr1.concat(arr2, arr3);
// [ 'a', 'b', 'c', 'd', 'e' ]

// ES6 的合并数组
[...arr1, ...arr2, ...arr3]
// [ 'a', 'b', 'c', 'd', 'e' ]
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

04 array

Spread String

```
[...'hello']
// [ "h", "e", "l", "l", "o" ]
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

04 array

find() , findIndex()

```
[1, 4, -5, 10].find((n) => n < 0)
// -5
```

```
[1, 5, 10, 15].find(function(value, index, arr) {
  return value > 9;
}) // 10
```

```
[1, 5, 10, 15].findIndex(function(value, index, arr) {
  return value > 9;
}) // 2
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

04 array

fill()

```
['a', 'b', 'c'].fill(7, 1, 2)
// ['a', 7, 'c']
```

flat()

```
[1, 2, [3, 4]].flat()
// [1, 2, 3, 4]
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

05 object

Object.assign()

```
const target = { a: 1 };

const source1 = { b: 2 };
const source2 = { c: 3 };

Object.assign(target, source1, source2);
target // {a:1, b:2, c:3}
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

05 object

Shallow copy

```
const obj1 = {a: {b: 1}};
const obj2 = Object.assign({}, obj1);

obj1.a.b = 2;
obj2.a.b // 2
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

06 Set and Map

Set

```
const s = new Set();

[2, 3, 5, 4, 5, 2, 2].forEach(x => s.add(x));

for (let i of s) {
  console.log(i);
}
// 2 3 5 4
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

06 Set and Map

Map

```
const map = new Map([
  ['name', '张三'],
  ['title', 'Author']
]);

map.size // 2
map.has('name') // true
map.get('name') // "张三"
map.has('title') // true
map.get('title') // "Author"
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

07 Promise

Promise Demo 1

```
function timeout(ms) {
  return new Promise((resolve, reject) => {
    setTimeout(resolve, ms, 'done');
  });
}

timeout(100).then((value) => {
  console.log(value);
});
```

Promise Demo 2

```
let promise = new Promise(function(resolve, reject) {
  console.log('Promise');
  resolve();
});

promise.then(function() {
  console.log('resolved.');
});

console.log('Hi!');

// Promise
// Hi!
// resolved
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

07 Promise

Use Promise to load image async

```
function loadImageAsync(url) {
  return new Promise(function(resolve, reject) {
    const image = new Image();

    image.onload = function() {
      resolve(image);
    };

    image.onerror = function() {
      reject(new Error('Could not load image at ' + url));
    };

    image.src = url;
  });
}
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

07 Promise

Promise catch()

```
// bad
promise
  .then(function(data) {
    // success
  }, function(err) {
    // error
  });

// good
promise
  .then(function(data) { //cb
    // success
  })
  .catch(function(err) {
    // error
  });
```

catch() can also get the error in ***then*** function.

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

07 Promise

Promise.all()

```
const p = Promise.all([p1, p2, p3]);
```

```
const p1 = new Promise((resolve, reject) => {
  resolve('hello');
})
.then(result => result)
.catch(e => e);

const p2 = new Promise((resolve, reject) => {
  throw new Error('报错了');
})
.then(result => result)
.catch(e => e);

Promise.all([p1, p2])
.then(result => console.log(result))
.catch(e => console.log(e));
// ["hello", Error: 报错了]
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

08 Generator

Generator :the Generator object is returned by a generator function and it conforms to both the iterable protocol and the iterator protocol.

```
function* helloWorldGenerator() {
  yield 'hello';
  yield 'world';
  return 'ending';
}

var hw = helloWorldGenerator();

hw.next()
// { value: 'hello', done: false }

hw.next()
// { value: 'world', done: false }

hw.next()
// { value: 'ending', done: true }

hw.next()
// { value: undefined, done: true }
```

The ***function**** declaration defines a generator function

The ***yield*** keyword is used to pause and resume a generator function

Three states:
1. hello,
2. world,
3. return

08 Generator

next() with parameters

```
function* foo(x) {
  var y = 2 * (yield (x + 1));
  var z = yield (y / 3);
  return (x + y + z);
}

var a = foo(5);
a.next() // Object{value:6, done:false}
a.next() // Object{value:NaN, done:false}
a.next() // Object{value:NaN, done:true}

var b = foo(5);
b.next() // { value:6, done:false }
b.next(12) // { value:8, done:false }
b.next(13) // { value:42, done:true }
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

08 Generator

For…of can iterate the Generator function without next()

```
function* foo() {
  yield 1;
  yield 2;
  yield 3;
  yield 4;
  yield 5;
  return 6;
}

for (let v of foo()) {
  console.log(v);
}
// 1 2 3 4 5
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

08 Generator

yield* :Generator inside generator

```
function* bar() {
  yield 'x';
  yield* foo();
  yield 'y';
}

// 等同于
function* bar() {
  yield 'x';
  yield 'a';
  yield 'b';
  yield 'y';
}
```

```
function* foo() {
  yield 'a';
  yield 'b';
}
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

08 Generator

Generator for state machine

Traditional clock function

```
var ticking = true;
var clock = function() {
  if (ticking)
    console.log('Tick!');
  else
    console.log('Tock!');
  ticking = !ticking;
}
```

New clock function with Generator

```
var clock = function* () {
  while (true) {
    console.log('Tick!');
    yield;
    console.log('Tock!');
    yield;
  }
};
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

08 Generator

Pipeline: step1 => step2 => step3 => step4

Traditional callback code

```
step1(function (value1) {
  step2(value1, function(value2) {
    step3(value2, function(value3) {
      step4(value3, function(value4) {
        // Do something with value4
      });
    });
  });
});
```

Generator

Promise

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

08 Generator

Pipeline: step1 => step2 => step3 => step4

Promise code

```
Promise.resolve(step1)
  .then(step2)
  .then(step3)
  .then(step4)
  .then(function (value4) {
    // Do something with value4
  }, function (error) {
    // Handle any error from step1 through step4
  })
  .done();
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

08 Generator

Pipeline: step1 => step2 => step3 => step4

Generator code

```
function* longRunningTask(value1) {
  try {
    var value2 = yield step1(value1);
    var value3 = yield step2(value2);
    var value4 = yield step3(value3);
    var value5 = yield step4(value4);
    // Do something with value4
  } catch (e) {
    // Handle any error from step1 through step4
  }
}
```

For practice…

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

09 Class

Class in JavaScript

```
class Point {
  constructor(x, y) {
    this.x = x;
    this.y = y;
  }

  toString() {
    return '(' + this.x + ', ' + this.y + ')';
  }
}
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

09 Class

Static method (or Class method)

```
class Foo {
  static bar() {
    this.baz();
  }
  static baz() {
    console.log('hello');
  }
  baz() {
    console.log('world');
  }
}

Foo.bar() // hello
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

09 Class

Private method (fake in JS)

```
class Widget {

  // 公有方法
  foo (baz) {
    this._bar(baz);
  }

  // 私有方法
  _bar(baz) {
    return this.snaf = baz;
  }

  // ...
}
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

09 Class

Class extends

```
class ColorPoint extends Point {
  constructor(x, y, color) {
    super(x, y); // 调用父类的constructor(x, y)
    this.color = color;
  }

  toString() {
    return this.color + ' ' + super.toString(); // 调用父类的toString()
  }
}
```

10 Module

Export and Import

```
// profile.js
var firstName = 'Michael';
var lastName = 'Jackson';
var year = 1958;

export { firstName, lastName, year };
```

```
import { firstName, lastName, year } from './profile.js';

function setName(element) {
  element.textContent = firstName + ' ' + lastName;
}
```

https://www.w3schools.com/js/js_es6.asp; http://es6.ruanyifeng.com;

Web Framework Tutorial 00

Danqing Shi

Tongji University