GlintsAcademy
Powered by BINAR ACADEMY

ES6

Isumi
Batam, Jan 2020

BINAR ACADEMY

GOALS

1. let
2. const
3. Arrow Function
4. Import & export
5. Classes

Javascript vs ECMAscript

Javascript is general purpose scripting language that conforms to the ECMAScript specification.

var, let and const

- Use Let avoid var
- Use const for static and constant variable

	Redeclare	Hoisting	Block Scope	Create global props
var	√	√	X	√
let	X	X	√	X
const	X	X	√	X

BINAR ACADEMY

var vs. let

Using let is more be safer than using var, usually var make everyone confuse

Example:

var

```
var x = 10
function myFuncA(){
  var x = 1
  console.log('Inside myFuncA :' + x)
}
myFuncA();
//var x = 100
console.log(x)
```

let

```
let y = 10
function myFuncB(){
  let y = 1
  console.log('Inside myFuncB :' + y)
}
myFuncB();
//let y = 100
console.log(y)
```

https://playcode.io/445545

const

The value of a constant cannot change through reassignment, and it can't be redeclared.

Example:

```
const KEY=123;
KEY = 456; //=> TypeError: Assignment to constant variable.

const KEY=123;
var KEY = 456; //=> TypeError: Assignment to constant variable.
```

const exception

In Javascript object and array is mutable

Object & Array

```
const obj = {id:1, name:'connor'}
obj.location = 'batam'
console.log(obj) // { id:1, name:'connor',location:'batam'}
obj={} //TypeError: Assignment to constant variable.

const arr = [1,2,3,4]
arr.push(5)
console.log(arr) //[1,2,3,4,5]
arr=[] //TypeError: Assignment to constant variable.
```

Arrow Function

Arrow function is a shorthand for function and more simpler

General

```
// ES5 / Javascript function
var multiplyES5 = function(x, y) {
  return x * y;
};
```

vs

Arrow

```
// ES6 Arrow function
const multiplyES6 = (x, y) => {
  return x * y
};
```

Import

Export

To make objects, functions, classes or variables available to the outside world it’s as simple as exporting them and then importing them where needed in other files.

export

Example:

```
//lib.js
export const x = 2;
export function multiply(a,b){
  return a * b;
}
export function multiplyWithx(y){
  return x * y;
}
```

import

Example:

```
//main.js
import { multiply, multiplyWithx } from './lib';
console.log(multiply(5,5)); // 25
console.log(multiplyWithx(6)); // 12
```

Classes

Classes are in fact "special functions", and just as you can define function expressions and function declarations, the class syntax has two components: class expressions and class declarations.

Classes

```
class Rectangle {
  constructor(height, width) {
    this.height = height;
    this.width = width;
  }
  // Getter
  get area() {
    return this.calcArea();
  }
  // Method
  calcArea() {
    return this.height * this.width;
  }
}

const square = new Rectangle(10, 10);

console.log(square.area); // 100
```

Class declaration

Define Constructor

Getter

Function / Method to run

Call the class

Explore ES6

1. ES6 Features : http://es6-features.org/
2. ECMAScript and Javascript differences : https://medium.freecodecamp.org/whats-the-difference-between-javascript-and-ecmascript-cba48c73a2b5
3. ES6 Video Tutorial : https://www.youtube.com/watch?v=IEf1KAcK6A8

"The world is yours to explore."

Exercises

1. Convert the calculator function (exercise in week 2) to ES6
2. Try import-export using babel (https://idjs.github.io/belajar-nodejs/es6/memakai_es6.html)