

# EcmaScript 6+

## JavaScript's Golden Age

EcmaScript  $\leq$  5



Dojo  
ExtJS  
jQuery  
MooTools  
Prototype.js  
YUI

# EcmaScript 6

# Top ES6, 7, 8 Features

1. The `let` and `const`
2. Default parameter
3. Template strings
4. Arrow function
5. Enhanced Object Literals
6. Classes
7. Iterator for ... of, in
8. Destructuring Assignment
9. Rest/Spread operator
10. Modules
11. Promise
12. Async/Await
13. Generators
14. etc

# Hoisting

Hoisting is JavaScript's default behavior of moving all declarations to the top of the current scope (to the top of the current script or the current function).

# Hoisting - example 1

```
function sayHi() {  
  console.log('hi!')  
}  
  
sayHi()
```

## Hoisting - example 2

```
sayHi()
```

```
function sayHi() {  
  console.log('hi!')  
}
```



## Hoisting - example 3

```
var sayHi = function() {  
  console.log('hi!')  
}  
  
sayHi()
```

## Hoisting - example 4

```
sayHi()  
  
var sayHi = function() {  
  console.log('hi!')  
}
```

## Hoisting - example 5

```
a = 30
```

```
function sayNumber() {  
    console.log(a)  
}
```

```
a = 20
```

```
sayNumber()
```

## var issue 1

```
var a = 30

function sayNumber() {
  a = a + 55
  console.log(a)
}

sayNumber()

console.log(a) // 85
```

## var issue 2

```
var i = 0  
  
for (var i = 0; i <= 45; i++) {  
  
}  
  
console.log(i) // ?
```

# let scope

```
let i = 0  
  
for (let i = 0; i <= 45; i++) {  
  
}  
  
console.log(i) // ?
```

# Default parameter

```
function limit(text, count = 100) {  
  
}  
  
// Before  
function limit(text, count) {  
    var count = count || 100  
}
```

# Template String - Before

```
var longText = "This is " +  
               "multiple line " +  
               "of text."
```



# Template String - Before

```
let count = 2
let html = "<div class=\"highlight\">" +
  "  <h1>" +
  "" + "Important Stuff" + "(" + count + ")" +
  "  </h1>" +
  "</div>"
```

# Template String

```
let count = 2
let html = `
```

# Arrow function

```
const uppercase = function(text) {  
  return text.toUpperCase()  
}
```

```
const uppercase = text => text.toUpperCase()
```

# Arrow function

```
const repeat = function(text, count = 100) {  
  return text.repeat(count)  
}
```

```
const repeat = (text, count = 100) => text.repeat(count)
```

Arrow function  
is not  
function replacement

Arrow function can't access  
this, arguments, super, or new.target

# Enhanced Object Literals

```
const name = "Beef Burger"  
const price = 14500  
const onSale = true  
  
const item = {  
  name: name,  
  price: price,  
  onSale: onSale,  
}
```

# Enhanced Object Literals

```
const name = "Beef Burger"  
const price = 14500  
const onSale = true  
  
const item = {  
  name,  
  price,  
  onSale,  
}
```

# Classes

```
class BasicAuth extends Auth {  
  constructor(username, password) {  
    super()  
    this.username = username  
    this.password = password  
  }  
  
  login() {  
    console.log(this.username, this.password)  
  }  
}
```



## for .. of

```
const fruits = [  
  { id: 1, name: 'Jagung' },  
  { id: 2, name: 'Ketela' },  
]  
  
for (let f of fruits) {  
  console.log(f.name) // Jagung, Ketela  
}
```

## for .. in

```
const fruits = [  
  { id: 1, name: 'Jagung' },  
  { id: 2, name: 'Ketela' },  
]  
  
for (let i in fruits) {  
  console.log(i)                // 0, 1  
  
  console.log(fruits[i].name) // Jagung, Ketela  
}
```

# Destructuring Assignment

```
const csv = 'Pixel 4 XL,12800000,8'

const [name, price, qty] = csv.split(',')

console.log(name)           // Pixel 4 XL
console.log(price)          // 12800000
console.log(qty)             // 8
```

# Destructuring Assignment

```
const json = {  
  "name": "iPhone 11 Pro",  
  "price": 18900000,  
  "qty": 6,  
}  
  
const { name, price, qty } = json  
  
console.log(name)           // iPhone 11 Pro  
console.log(price)          // 18900000  
console.log(qty)             // 6
```

# Rest operator

```
function sum(...numbers) {  
  }
```

```
sum(1, 3, 5, 1, 3, 4, 9)
```

```
sum(1, 3, 5)
```

```
sum(1, 3, 52, 3, 73, 29, 97, 288, 165, 549)
```

# Spread operator

```
const json = {  
  name: 'Someone',  
  profiles: {  
    gender: 'male',  
    birth: '2019'  
  }  
}
```

```
const account = {  
  name: account.name,  
  ...account.profiles  
}
```

# Modules

```
// a.js  
export const number = 22
```

```
// b.js  
import { number } from './a.js'  
  
console.log(number)    // 22
```

# Modules

```
// a.js  
export default name = 'Something'
```

```
// b.js  
import anything from './a.js'  
  
console.log(anything)    // 'Something'
```



# Promise & Async/Await

Questions?