



# JavaScript – ECMA Script 6

Sung-Dong Kim,  
School of Computer Engineering,  
Hansung University

# Contents

- ▶ ECMA Script 6
- ▶ 백틱
- ▶ for of, forEach()
- ▶ 화살표 함수
- ▶ 일급 함수 (일급 객체)
- ▶ Callback Function

# ECMA Script 6

- ▶ ECMA (European Computer Manufacturer Association) 기관에서 표준화 한 범용 프로그래밍 언어
- ▶ JavaScript 표준: 다양한 web browser에서 web page의 상호 동작을 보장
- ▶ WWW 상에서 client-side scripting에 이용됨
- ▶ Node.js를 이용한 server application and services 개발에 이용됨

# ECMA Script 6

- ▶ ECMA Script 6
- ▶ ECMA Script 2015
- ▶ <http://www.ecma-international.org/ecma-262/6.0/>
- ▶ ECMAScript 2015 Language Specification
- ▶ 편리한 기능이 많이 추가되었음

```
console.log('name:%s', 'ksd'); // 템플릿 형식으로 출력하기  
console.log('age:%d', 31); // 숫자는 %d  
console.log('pi:%d', 3.14); // 숫자는 %d  
console.log('math:%d science:%d', 92, 84); // 2가지 같이 쓰기  
console.log('name:%s age:%d', 'ksd', 31); // 여러가지 같이 쓰기  
console.log('name:%s math:%d science:%d', 'ksd', 92, 84); // 3가지 같이 쓰기  
console.log('name:%s math:%s science:%s', 'ksd', 92, 84); // %s로 다 쓰기
```

# 백틱

- ▶ 백틱을 이용한 포매팅: `\${variable}`

```
let num1 = 1;  
let num2 = 2;  
let result = 3;  
const string1 = num1 + ' 더하기 ' + num2 + '는 \'' + result + '\"입니다.';  
const string2 = `${num1} 더하기 ${num2}는 \"${result}\"입니다.`;  
console.log(string1);  
console.log(string2);
```

# for of, forEach()

```
const userList = [  
  { name: 'ksd', age: 31, score: 85 },  
  { name: 'kjh', age: 13, score: 95 },  
  { name: 'pch', age: 35, score: 76 },  
];
```

```
for (const user of userList) {  
  console.log('user:', user);  
}
```

```
userList.forEach(function(user){  
  console.log(user);  
});  
userList.forEach(user => console.log(user));
```

```
const studentList = [  
  { name: 'ksd', age: 31, math: 85, science: 75 },  
  { name: 'kjh', age: 31, math: 95, science: 83 },  
  { name: 'pch', age: 35, math: 76, science: 98 },  
];  
  
const plus = (a, b) => a + b;  
  
studentList.forEach(student => console.log(  
  `${student.name} total: ${plus(student.math, student.science)}`));
```



# 화살표 함수

## ▶ 화살표 함수 (arrow function)

```
const printHello = () => console.log('hello'); // no {}  
const printHello2 = () => 'hello2'; // return 'hello2'  
const printMessage = message => console.log(message);  
const plus = (a, b) => a + b;  
const minus = (a, b) => a - b;
```

```
printHello();  
console.log(printHello2());  
printMessage('message');  
console.log('plus : %d', plus(10, 20));  
console.log('minus : %d', minus(10, 20));
```

```
function printHello() {  
  console.log('hello');  
}  
printHello();  
  
const printBye = () => {  
  console.log('bye');  
}  
printBye();
```

```
const add1 = function(x, y) {  
  return x + y;  
}  
const add2 = (x, y) => {  
  return x + y;  
}  
const add3 = (x, y) => x + y;  
const square = x => x ** 2;  
const square2 = square;
```

# 일급 함수 (일급 객체)

- ▶ 함수를 parameter로 받음
- ▶ 함수를 parameter로 전달

```
const plus = (a, b) => a + b;  
const minus = (a, b) => a - b;  
function calculate(a, b, func){  
    return func(a, b);  
}  
console.log(calculate(10, 20, minus));  
console.log(calculate(10, 20, plus));
```

# Callback Function

- ▶ 함수에 parameter로 전달된 함수

```
const sum = (a, b) => a + b;

const printResult = (result) => {
  console.log(`결과는 ${result} 입니다.`);
};

const calculationAndPrint = (calculationResult, callback) => {
  callback(calculationResult);
};

calculationAndPrint(sum(10, 20), printResult);
```

# 오늘 공부한 것

- ▶ ECMA Script 6
- ▶ 백틱
- ▶ for of, forEach()
- ▶ 화살표 함수
- ▶ 일급 함수 (일급 객체)
- ▶ Callback Function