

ECMAScript 6

ECMAScript 6 Introduction

ECMA

ECMA?

- ECMA (ECMA International)
 - 정보 통신에 대한 표준을 제정하는 비영리 표준화 기구
- ECMAScript는 ECMA에서 [ECMA-262](#)* 규격에 따라 정의한 언어
 - ECMA-262*: 범용적인 목적의 프로그래밍 언어에 대한 명세
- ECMAScript6는 ECMA에서 제안하는 [6번째 표준 명세](#)를 말함
 - (참고) ECMAScript6의 발표 연도에 따라 ECMAScript2015라고도 불림

세미콜론

세미콜론 (semicolon)

- 자바스크립트는 세미콜론을 선택적으로 사용 가능
- 세미콜론이 없으면 ASI*에 의해 자동으로 세미콜론이 삽입됨
 - [ASI*: 자동 세미콜론 삽입 규칙 \(Automatic Semicolon Insertion\)](#)
- 본 수업에서는 자바스크립트의 문법 및 개념적 측면에 집중하기 위해
세미콜론을 사용하지 않고 진행

세미콜론 (semicolon) 예시

```
// 세미콜론이 있는 경우  
const greeting = 'Hello, world!';  
console.log(greeting);
```

```
// 세미콜론이 없는 경우  
const greeting = 'Hello, world!'  
console.log(greeting)
```

선택적으로 사용 가능

코딩 스타일 가이드

코딩 스타일 가이드

- 코딩 스타일의 핵심은 합의된 원칙과 일관성
 - 절대적인 하나의 정답은 없으며, 상황에 맞게 원칙을 정하고 일관성 있게 사용하는 것이 중요
- 코딩 스타일은 코드의 품질에 직결되는 중요한 요소
 - 코드의 가독성, 유지보수 또는 팀원과의 커뮤니케이션 등 개발 과정 전체에 영향을 끼침
- (참고) 다양한 자바스크립트 코딩 스타일 가이드
 - [Airbnb Javascript Style Guide](#)
 - [Google Javascript Style Guide](#)
 - [standardjs](#)

코딩 스타일 가이드 예시

- 본 수업에서는 [Airbnb Style Guide](#)를 중심으로 진행할 예정
 - 단, 가이드의 일부 항목은 문법 및 개념적 측면에 집중하기 위해 변형해서 사용

- 19.1 Use soft tabs (space character) set to 2 spaces. eslint: `indent`

```
// bad
function foo() {
  ....let name;
}
```

```
// bad
function bar() {
  ·let name;
}
```

```
// good
function baz() {
  ··let name;
}
```

- 13.1 Always use `const` or `let` to declare variables. Not doing so will result in global variables, polluting the global namespace. Captain Planet warned us of that. eslint: `no-undef` `prefer-con`

```
// bad
superPower = new SuperPower();
```

```
// good
const superPower = new SuperPower();
```