

LEARNING SPOONS ONLINE

자바스크립트 기초

자바스크립트 시작하기



+jay.ko (고재도)
haibane84@gmail.com

- (현) [“Endowus”](#) - 트레이딩 팀 개발 팀장
- (전) [“카카오뱅크”](#) - 시니어 소프트웨어 엔지니어
- (전) [“Awair”](#) - 시니어 소프트웨어 엔지니어
- (전) [“KT”](#) - 전임연구원
- (전) [“Google Developer Expert”](#) - 웹 기술 분야
- [“자바스크립트 200제”](#) 저자
- [“시작하세요 AngularJS 프로그래밍”](#) 저자

- Section 1.** 자바스크립트 시작하기
2. 자바스크립트 기초 다지기
 3. 자바스크립트 객체지향 프로그래밍
 4. 모듈과 기타 주요 기능
 5. DOM 프로그래밍

Section 1.

2.

나의 하루

3.

10월 28일

4.

☐ 공부하기

5.

☐ 놀기☐ 밥먹기☒ 할 일 추가

Unit

1. 자바스크립트 개발환경
2. 자바스크립트 정의 살펴보기
3. 값과 변수 이해하기
4. 표현식과 명령문
5. 자바스크립트 자료형
6. 기본적인 컨트롤 플로우 처리하기

Unit 1-1

자바스크립트 개발환경

로컬 개발환경 셋업

1. 비주얼 스튜디오 코드 설치 <https://code.visualstudio.com/>
2. 노드제이에스 설치 <https://nodejs.org/ko/>

Visual Studio Code 프로젝트 생성

1. 원하는 위치에 workspace 폴더를 생성해주세요.
2. 각 주차별 프로젝트 폴더를 workspace 폴더에 만들어주세요.
(예, c:\workspace\js)
3. 터미널에서 프로젝트에 위치한 다음 다음과 같이 입력하세요.

```
$ code .
```


Unit 1-2

자바스크립트 정의 살펴보기

자바스크립트 정의

JS 약어로 불리는 자바스크립트 JavaScript는 동적 [dynamic](#), 약타입 [weakly typed](#), 프로토타입 기반 [prototype-based](#), 멀티 패러다임 [multi-paradigm](#) 특징들을 가지는 고급 [high-level](#) 인터프리터 프로그래밍 언어 [interpreted programming language](#)이다. (출처: <https://en.wikipedia.org/wiki/JavaScript>)

인터프리터 언어

인터프리터 언어^{interpreted language}는 컴파일러를 거쳐서 기계어로 변환되지 않고 바로 실행되는 프로그래밍 언어를 말한다. 반대로 반드시 기계어로 컴파일되어야만 실행시킬 수 있는 언어를 컴파일 언어라 한다. (출처: https://ko.wikipedia.org/wiki/인터프리터_언어)

자바스크립트 엔진

- 자바스크립트 엔진은 자바스크립트로 작성한 코드를 해석하고 실행하는 인터프리터
- 주로 웹 브라우저에 이용되지만 Node.js와 같이 JavaScript Engine을 장착한 서버 사이드에서 사용되고 있다.
- 아래와 같은 엔진들이 대표적인 자바스크립트 엔진이다.
 - Google의 V8 (<https://developers.google.com/v8/>)
 - Mozilla의 SpiderMonkey (<https://developer.mozilla.org/ko/docs/SpiderMonkey>)
 - Microsoft의 chakra (<https://github.com/Microsoft/ChakraCore>)

그외 특징

- C 언어나 자바와 닮은 구문
 - if나 while 등의 키워드가 같고 제어 구조도 비슷한 형태
- 동적 타입 언어
 - 변수나 함수의 반환값에 타입을 지정하지 않음
- 프로토타입 기반 객체지향
 - 클래스 기반 객체지향과 다름
- 리터럴 표기의 표현력
 - {}, [] 와 같이 쉽게 객체나 배열을 표현
- 함수형 프로그래밍
 - [1, 2, 3].forEach(alert);

자바스크립트 애플리케이션 실행 환경



브라우저
(클라이언트)



Node.js
(서버)

Unit 1-3

값과 변수 이해하기

값(value)과 변수(variable)

- 값(value)은 코드상에서 데이터를 나타내는 방법이고 여러 종류의 값들을 표현하는 방식을 리터럴이라고 한다.
- 변수는 값을 넣는 공간이라 할 수 있고 자바스크립트에서는 var, let 키워드를 사용하여 선언할 수 있다.

```
var v;  
var v1, v2;  
var v = 2;           // 숫자 리터럴  
var v = "2";         // 문자 리터럴  
var v = '2';         // 문자 리터럴  
var v = true;        // 불리언 리터럴  
var v = { p1: 2, p2: '2' }; // 객체 리터럴  
var v = [1, 2, 3, 4]   // 배열 객체 리터럴
```


Unit 1-4

표현식과 명령문

표현식과 명령문

- 표현식(Expression)은 값을 생성합니다.
- 명령문(Statements)은 일종의 행동 또는 행위를 수행하게 하는 코드입니다.

```
(3 + 12) / 5  
  
function greeting() {  
    "안녕하세요";  
    hi();  
}  
  
greeting();  
greeting(if (true) { console.log("It is not acceptable"); }));
```

Unit 1-5

데이터 자료형

데이터 자료형

- 6개의 원시 데이터 타입과 객체로 이루어져 있음

원시 데이터 타입

- 숫자
- 문자열
- 불린
- null
- undefined
- symbol (ECMAScript 6)

객체

배열

함수

날짜, 정규식 (내장 객체)

Unit 1-6

기본적인 컨트롤 플로우 처리하기

조건문

- 조건문이란, 어떤 조건이 참인지 거짓인지에 따라 실행 여부를 결정하는 구분을 말합니다.
- 대표적으로 if, switch가 있습니다.

```
if (표현식1) {  
    명령문1  
} else if (표현식2) {  
    명령문2  
} else {  
    명령문4  
}
```

```
switch (표현식) {  
    case 값1:  
        명령문1  
        break;  
    default:  
        명령문3  
}
```

반복문

- 반복문은 특정 코드를 반복하는 흐름을 제어합니다. 특정 코드를 특정 조건에 맞게 반복해서 실행하기를 원할 때 유용하게 사용할 수 있습니다.
- 대표적으로 for, while 문이 있습니다.

```
for (초기값; 조건식; 어떤 간격으로) {  
    실행할 문장  
}
```

```
for (속성명 in 반복할 대상 객체) {  
    실행할 문장  
}
```

```
while (조건식) {  
    반복하게 될 문장  
}
```

```
do {  
    반복하게 될 문장  
} while (조건식)
```