

Nodejs

2016/11/27

진행

- ▶ nodejs 개발 환경 세팅
- ▶ 실습을 통한 JavaScript 기본 문법 정리
- ▶ 간단한 콘솔 App 만들기
- ▶ 자율 실습 및 스터디 정리

Node.js란?

▶ <https://ko.wikipedia.org/wiki/Node.js> (위키에 의하면...)

- ▶ Node.js는 확장성 있는 네트워크 애플리케이션(특히 서버 사이드) 개발에 사용되는 소프트웨어 플랫폼이다. 작성 언어로 자바스크립트를 활용하며 Non-blocking I/O와 단일 스레드 이벤트 루프를 통한 높은 처리 성능을 가지고 있다.
- ▶ 내장 HTTP 서버 라이브러리를 포함하고 있어 웹 서버에서 아파치 등의 별도의 소프트웨어 없이 동작하는 것이 가능하며 이를 통해 웹 서버의 동작에 있어 더 많은 통제를 가능케 한다.
- ▶ V8 (자바스크립트 엔진) 위에서 동작하는 이벤트 처리 I/O 프레임워크이다. 웹 서버와 같이 확장성 있는 네트워크 프로그램 제작을 위해 고안되었다.

▶ (node.)JavaScript

- ▶ 서버 개발을 하기위한 서버 사이드 언어(javascript)다.

Nodejs 설치 (+npm)

- ▶ Windows에서 설치

- ▶ <https://youtu.be/dbxl7dPebHA>
- ▶ <https://docs.google.com/presentation/d/1TYb3pDd-LfoPtKghVUoKcPtkCBIEsvVVtsbEpsbc16g/edit#slide=id.p>

- ▶ MacOS에서 설치

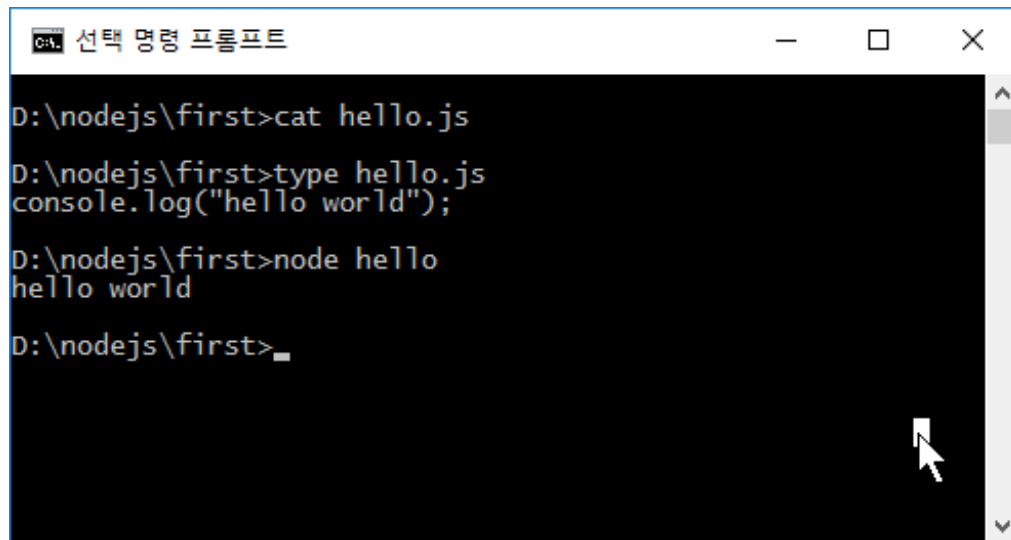
- ▶ https://docs.google.com/presentation/d/16Uc5s8TWS_Sti6wvGXW3flRZAZpvdu8n2f7SrtJXalo/edit#slide=id.p

- ▶ Linux에서 설치

- ▶ Ubuntu에서 : <https://youtu.be/MXTrA9fKWYQ>

Hello world로 실습 방법 익히기

- ▶ 편집기는 자유 (atom, bracket, sublime 등등)
- ▶ `console.log("hello world");`를 좋아하는 편집기로 편집 hello.js로 저장
- ▶ 콘솔(cmd/터미널) 에서 `node hello`로 수행



```
선택 명령 프롬프트
D:\nodejs\first>cat hello.js
D:\nodejs\first>type hello.js
console.log("hello world");
D:\nodejs\first>node hello
hello world
D:\nodejs\first>
```

- ▶ REPL : 실전에서 거의 사용 안 할 것 같아서 패스...

JavaScript 기초 문법

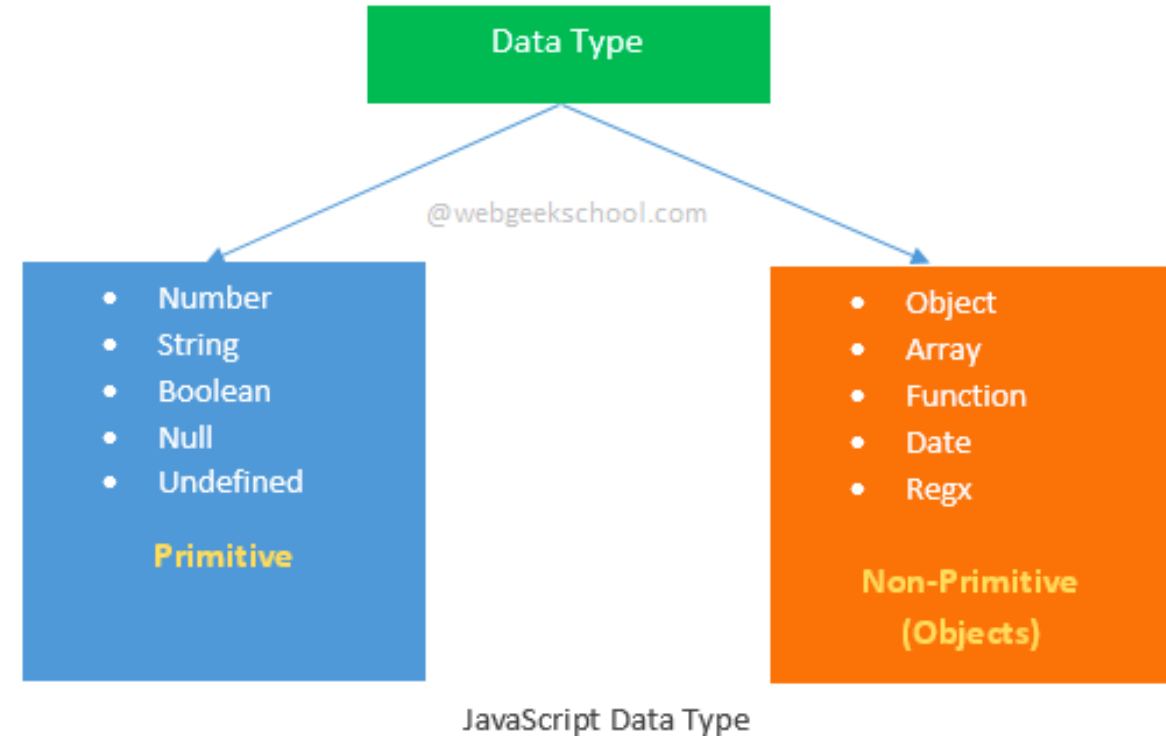
- ▶ 변수와 자료형
- ▶ 연산자
- ▶ 배열
- ▶ 콜렉션
- ▶ 조건문
- ▶ 반복문
- ▶ 모듈화
- ▶ ES6 문법

변수와 자료형

- ▶ 선언 : 자료 타입에 상관없이 **var** 키워드를 사용

```
var 변수명 = 값;
```

- ▶ 문자열은 큰따옴표, 작은따옴표 구분을 하지 않음!



- ▶ https://developer.mozilla.org/ko/docs/Web/JavaScript/Data_structures

연산자

우선순위	연산자	내용
1	() , []	괄호 / 대괄호
2	!, ~, ++, --	부정 / 증감 연산자
3	*, /, %	곱셈 / 나눗셈 연산자
4	+, -	덧셈 / 뺄셈 연산자
5	<<, >>, >>>	비트단위의 쉬프트 연산자
6	<, <=, >, >=	관계 연산자
7	==, !=	
8	&	비트단위의 논리연산자
9	^	
10		
11	&&	논리곱 연산자
12		논리합 연산자
13	?:	조건 연산자
14	=, +=, -=, *=, /=, %= &=, ^=, ~=	대입 / 할당 연산자

주의) 문자열 붙이기(concat)도 + 기호를 사용

typeof 키워드

- ▶ typeof는 자료형을 알려주는 '연산자'입니다.

실습 코드 : <https://github.com/owo-study/nodejs2016/tree/master/1weeks>

Table 20 — **typeof** Operator Results

<i>Type of val</i>	<i>Result</i>
Undefined	"undefined"
Null	"object"
Boolean	"boolean"
Number	"number"
String	"string"
Object (native and does not implement <code>[[Call]]</code>)	"object"
Object (native or host and does implement <code>[[Call]]</code>)	"function"
Object (host and does not implement <code>[[Call]]</code>)	Implementation-defined except may not be "undefined", "boolean", "number", or "string".

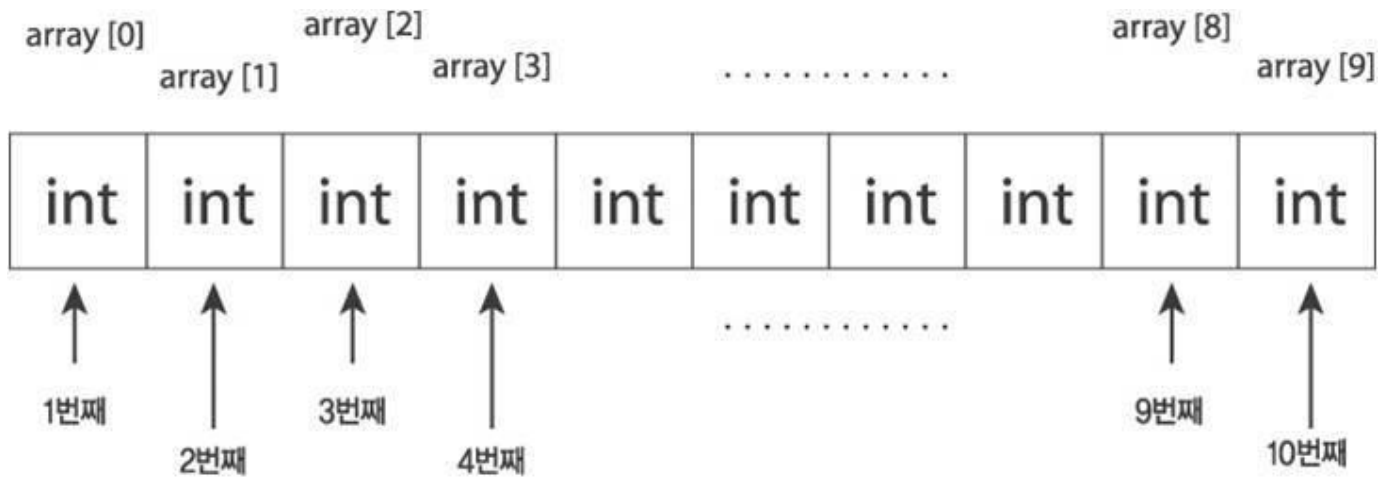
배열

▶ 배열이란?

- ▶ 같은 타입의 자료형을 연속된 **key-value**쌍으로 저장하는 자료구조
- ▶ 이론적으로 메모리에 같은 크기의 공간을 연속으로 할당 받아서 사용

A	0xF000
B	0xF001
C	0xF002
D	0xF003
E	0xF004
F	0xF005
G	0xF006

메모리에 연속해서 존재



int : int형 정수 저장을 위한 4바이트 메모리 블록

<http://programfrall.tistory.com>

컬렉션(Collections)

- ▶ Map

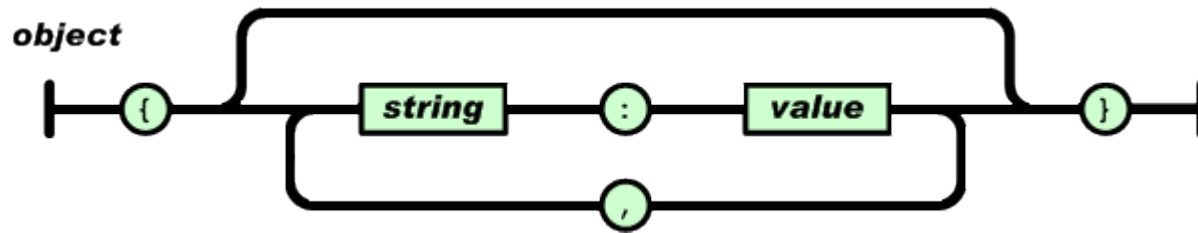
- ▶ Key-value가 1:1 매칭 되는 자료 구조

- ▶ Set

- ▶ 집합! 중복되는 값이 없는 자료구조

JSON

- ▶ key-value 형태의 쌍으로 구현되는 자료 구조 중 하나 타입.
 - ▶ 하위 value에 또 key-value쌍(재귀적)으로 값이 들어갈 수 있어 다양한 자료 수용가능
- ▶ <http://www.json.org/json-ko.html>



조건문 - if

```
if ( 조건식1* )  
{  
    // 조건식1이 true일 때 수행할 코드  
}  
else if ( 조건식2* )  
{  
    // 조건식2이 true일 때 수행할 코드  
}  
else  
{  
    // 모든 조건식에 해당하지 않을 때 수행할 코드  
}
```

조건식 : true/false로 나올 수 있는 값/식/함수 등 모든 것들

조건문 - switch

```
switch ( expression ) {  
  case 값1:  
    // expression의 값이 '값1'일 때 수행할 코드  
    break;  
  case 값2:  
    // expression의 값이 '값2'일 때 수행할 코드  
    break;  
  case 값3:  
    // expression의 값이 '값3'일 때 수행할 코드  
    break;  
  default:  
    // expression의 값이 '값1', '값2', '값3' 어디에도 해당하지  
    않을 때 수행할 코드  
}
```

실습코드 : <https://github.com/owo-study/nodejs2016/blob/master/1weeks/switch.js>

반복문 - while / do..while

```
while ( 조건식 ) {  
    // 조건식이 true일 때 반복할 코드  
}
```

```
do {  
    // 조건식이 true일 때 반복할 코드 (최초 1번은 기본 수행)  
} while ( 조건식 )
```

반복문 - for

```
for ( 초기값; 조건식; 증감식 ) {  
    // 조건식이 true일 때 수행할 코드  
}
```

비고) 초기값, 증감식 : 형식을 이렇게 사용할 뿐이지, 아무 명령어나 써도 됨

반복문 - for..in

```
for ( var(변수) in object(배열) ) {  
    // object의 하위 항목 개수만큼 만큼 var에 넣어서 반복할 코드  
}
```

비고) 초기값, 증감식 : 형식을 이렇게 사용할 뿐이지, 아무 명령어나 써도 됨

실습 코드 : <https://github.com/owo-study/nodejs2016/blob/master/1weeks/forin.js>

함수

- ▶ function 키워드로 생성

```
function 함수명(인자1, 인자2, ...) {  
    // 수행할 코드  
    [return 반환값]  
}
```

- ▶ 자주보는 특이한(!) 형태, 함수를 변수 다루듯이

```
var 함수명 = function (인자1, 인자2, ...) {  
    // 수행할 코드  
    [return 반환값]  
}
```

함수 (2)

- ▶ 심지어 **var**로 선언하지 않고 바로 변수의 인자 같이 쓰는 경우 (익명함수)

```
$(this).on("click", function (인자1, 인자2, ...) {  
    // 수행할 코드  
    [return 반환값]  
});
```

함수와 메소드

- ▶ 함수의 개념과 메소드의 개념 두가지를 같이 사용하는 독특한 형태
 - ▶ 메소드는 객체 지향에서의 개념!
- ▶ OOP (객체지향)
 - ▶ <https://namu.wiki/w/%EA%B0%9D%EC%B2%B4%20%EC%A7%80%ED%96%A5%20%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D>
 - ▶ 모든 것(프로그램의 단위)를 **object**(객체)를 기본으로 설계하는 프로그래밍의 방법 이론

모듈화 (exports, require)

- ▶ 프로그램을 모듈(파일 단위)로 나눌 수 있게 해주는 키워드
 - ▶ (다른 언어의 `include`에 해당)
- ▶ 실습 코드를 사용해서 알아보기
 - ▶ <https://github.com/owo-study/nodejs2016/blob/master/1weeks/square.js>
 - ▶ <https://github.com/owo-study/nodejs2016/blob/master/1weeks/module.js>

ES6

- ▶ ECMA라는 단체에서 JavaScript 표준을 제정
 - ▶ JavaScript가 브라우저마다 해석하는 엔진이 지원하는 규격이 달랐던 문제 해결
- ▶ ES6 = ECMAScript2015 (2015년에 정리된 6번째 버전)
- ▶ http://wiki.ecmascript.org/doku.php?id=harmony:specification_drafts
 - ▶ 영문 표준 가이드라인 페이지
- ▶ <http://es6-features.org/#Constants>
 - ▶ ES5(현재 우리가 가장 널리 사용하는 문법)와 ES6 문법의 비교해서 보여주는 사이트

간단한 콘솔 어플리케이션 만들기

- ▶ 로또 번호 생성기
 - ▶ Hint) 랜덤 정수 만들기 : `Math.floor(Math.random() * 10) + 1;`
- ▶ 가위바위보 or 숫자 야구 게임
 - ▶ Hint) `readline` 모듈 사용하여 입력 받기 가능