

자바스크립트 문법

Hellow JavaScript

코딩에 필요한 것

- 최신 브라우저
 - 강좌에서는 크롬 브라우저 사용
- 편집기(에디터)
 - 손에 익숙한 것 사용
- 사용하고 있는 것이 없다면
 - Visual Studio Code(VS Code)
 - 강좌에서는 VS Code 사용

HTML 파일 작성

- 확장자: html
- 인코딩: utf-8

```
<!DOCTYPE html>
<html lang=ko>
<head>
  <meta charset="utf-8">
  <title>자바스크립트</title>
</head>
<body>
</body>
</html>
```

html 파일에 JS 파일 작성 방법

- <head> 또는 <body>의
 - <script>에 js 파일 작성
 - 강좌에서는 <head>에 작성
- defer 속성 작성
- HTML과 JavaScript 분리 [코드 1](#)

```
<head>
  <script src="hellow.js" defer> </script>
</head>
<body>
  <script src="hellow.js" defer> </script>
</body>
```

html 파일에 **JS** 코드 직접 작성

- `<script>`에 아래 처럼 JS 코드를 직접 작성 `</script>`
- 웹 페이지에 한정할 때 사용

```
<script>  
    var hello = "안녕하세요.";  
</script>
```

문장

문장

- Statement
 - JS 코드 실행 단위
 - 세미콜론(;)까지 하나의 문장
 - `var book = "책";` [코드 1](#)
- 문장 시작 위치
 - 위치 제약 없음
 - 들여쓰기: 일반적으로 2칸, 4칸

변수

변수

- Variable
 - 값을 저장하는 영역
- 변수 사용 목적
 - 변수에 저장된 값의 재사용
- 변수 선언 방법 [코드 1](#)
- 값을 변수에 할당하는 방법 [코드 2](#)

변수 선언 방법

- 콤마로 구분하여 다수 작성 [코드 1](#)
- 줄을 바꾸어 작성 [코드 2](#)
- 변수 이름 앞에 콤마 작성 [코드 3](#)
- 의미를 부여하여 변수 이름 작성
 - 시맨틱 Semantic이라고 합니다.

변수에 값 할당 방법

- 나중에 할당한 값으로 바뀜 [코드 1](#)
 - `var point = 123, point = 456;`
- 같은 값 할당 [코드 2](#)
 - `var point = amount = 123;`

주석

주석

- 주석은 JS 코드로 인식하지 않음
- 한 줄 주석 [코드 1](#)
- 블록Block 주석 [코드 2](#)
- `/** 코드 */` 형태
 - JS 스펙에 정의된 주석은 아니며
 - 블록 주석과 같음 [코드 3](#)
- JS 스펙 Specification이란?
 - 자바스크립트 문법을 작성한 문서
 - ES3, ES5는 JS 스펙의 에디션 Edition

console.log()

console.log()

- console.log(...data)
 - 소괄호() 안에 작성된 값을 브라우저 콘솔 창에 출력
 - 문자, 숫자 등을 출력
 - 콤마로 구분하여 다수 작성 가능
 - 소괄호 안에 작성한 값을
강좌에서는 파라미터 값이라고 부릅니다.
- JS에서 제공하는 것은 아니며

[Console API 참조](#)

브라우저 콘솔 창 열기

- 크롬 브라우저, Window 기준
 1. 브라우저 상단 오른쪽 끝의 "Chrome 맞춤설정 및 제어"를 클릭
 2. 도구 더보기
 3. 개발자 도구(D) 클릭
 - 개발자 도구 단축 키: Ctrl+Shift+I
 4. 도구 창 메뉴에서 Console 클릭 [코드 1](#)
- 강좌에서는 오른쪽 형태 사용
 - 가독성을 위해 개발한 것 [코드 2](#)

정수, 실수

정수, 실수

- 정수
 - 소수가 없는 숫자
 - 123, -123
- 실수
 - 소수를 가진 숫자
 - 1.23, 1.0
- JS는 정수, 실수를 구분하지 않음
 - 1, 1., 1.0 모두 1.0으로 간주
단, 표시는 1로 표시 [코드 1](#)

숫자 처리

- 정수와 실수를 구분하지 않음
 - 64비트^{Bit} 부동 소수점 처리
 - IEEE 754 표준
- 부동 소수점 처리란?
 - 123을 123.0으로 처리 [코드 1](#)
- ES6에 정수, 실수 구분 추가

상수, 진수

상수

- 변경할 수 없는 값
- 상수 변수
 - 상수가 설정된 변수
 - JS는 변수의 값을 변경할 수 있으므로
 - 상수 변수는 선언적 의미
- 상수 변수 표기 방법
 - 코딩 관례로 영문 대문자 사용
 - `var ONE = 1;`
 - 상수로 사용한다는 시맨틱 선언 [코드 1](#)
- JS가 제공하는 상숫값은 변경 불가
 - MAX_VALUE, MIN_VALUE 등

진수

- 10 진수: 123
- 16 진수
 - 0xFF: 255
 - 1번째에 숫자 0 작성
 - 2번째에 영문자 x 작성
 - 3번째 이후: 0 ~ f 작성대소문자 구분하지 않음 [코드 1](#)
- 8 진수
 - ES3에서 폐지, ES6 재정의
- 2 진수: ES6에서 특별한 방법 제공

데이터 타입(**Data Type**)

데이터 타입

- 데이터 Data의 사전적 의미는 자료
 - 강좌에서는 데이터로 표기
- 데이터 타입 Type 형태
 - 숫자 타입: `var value = 123;`
 - 문자 타입: `var value = "sports";`
- `typeof` 연산자
 - 데이터(값) 타입 반환
 - `typeof value`에 데이터를 작성 [코드 1](#)
- 키워드 Keyword
 - 특별한 기능을 가진 단어

데이터 타입

- 데이터 타입을 자료형이라고도 부름
 - 강좌에서는 데이터 타입으로 표기
- 데이터는 타입을 가짐
 - JS는 데이터를 기준으로 타입을 결정
 - 타입을 먼저 선언하고 타입에 맞는 데이터를 할당하지 않음 [코드 1](#)

Number 타입, String 타입

데이터 타입 분류

- 언어 타입과 스펙 타입
- 언어 타입
 - JS 프로그램에서 사용할 수 있는 타입
 - Undefined, Null, Boolean, String, Number, Object
- 스펙(문서) 타입
 - 언어 알고리즘을 위한 타입으로 JS 프로그램에서 사용 불가
 - Reference, List, Completion, Property Descriptor, Data Block, Lexical Environment, Lexical Record 등

Number 타입

- Number 타입
 - 부호(+, -)를 가진 값 [코드 1](#)
- 숫자 값 범위
 - 18,437,736,874,454,810,627
(2의 64승 - 2의 53승 + 3)
- Number 타입의 특수한 3개 값
 - NaN: Not-a-Number [코드 2](#)
 - Infinity: 양수 무한대
 - -Infinity: 음수 무한대

String 타입

- 문자 타입
 - 값을 "" 또는 " 사이에 작성
 - 최대 문자수: 2의 53승 - 1
- 큰따옴표와 작은따옴표를 같이 사용할 때
 - 따옴표 작성 방법 [코드 1](#)
- 따옴표에 숫자를 작성하면 문자 타입이 됩니다. [코드 2](#)

Undefined, Null 타입

Undefined 타입

- Undefined(대문자) 타입
 - 값: undefined(소문자)
- 변수의 디폴트 Default 값
 - `var point;`
 - 변수를 선언만 한 것으로 `undefined`가 초기값으로 설정 [코드 1](#)
 - 변수에 값을 할당하지 않은 것을 나타내는 시맨틱
- 변수에 `undefined` 할당 가능 [코드 2](#)

Null 타입

- Null(대문자) 타입
 - 값: null(소문자)
- null과 undefined 차이
 - undefined는 단지 변수만 선언
 - null을 할당해야 값이 null이 됨
 - 의도적으로 값을 할당한 것으로 코드를 수행한 것이 됩니다. [코드 1](#)

Boolean 타입, Object 타입

Boolean 타입

- 불리언 타입
 - 값: true, false [코드 1](#)

Object 타입

- Object 형태
 - {name: value} 형태 [코드 1](#)
- 프로퍼티Property
 - name과 value 하나를 지칭
- Object는 프로퍼티 집합

타입 정리

- JS의 기본 데이터 타입 정리
- 기본 데이터 타입을 **Primitive** 타입이라고 합니다. [코드 1](#)
- 데이터 타입이 같다? [코드 2](#)