

JAVASCRIPT 기초

자바 스크립트란?

: 프로토타입 기반 객체 생성 지원하는 동적 스크립트 언어
웹브라우저에서 주로 사용, Node.js 이용하여 콘솔 환경에서 사용
웹브라우저 UI를 제어하기 위해 만들어진 프로그래밍 언어
자바와 기본 구문 비슷 (c언어 기본 구문 바탕)

HTML 자바스크립트 사용

- `<script></script>` 태그 사용
- 문서 내 위치 제약 없음 → body 태그 맨 밑에 보통 적음

외부스크립트 참조하기

1. .js 확장자 가진 파일 생성
2. 문서에서 `<script src = "외부 파일 위치"></script>`

변수

1. 자바 스크립트 변수 타입: 가리키는 값에 대한 타입 나타냄
2. var, let, const 이용해 변수 선언
 - var: ES6이전 / let, const: ES6 이후
3. var는 중복 선언 가능
4. undefined: 변수에 아무 값 X, 타입 모름
5. 동적 타입: 대입되는 값에 따라 용도 변경
6. 문자, \$, _로 시작, 대소문자 구분, 예약어 사용 X

var : 중복 선언 가능

- 재선언, 재할당 O
- ES6 이전 변수 선언 시 사용 → 자주 사용하지는 x

- 호이스팅(Hoisting) 특성
- 함수 스코프

let : 중복 선언 불가능

- 재선언 X, 재할당 O
- 블록 스코프

const: 중복 선언, 값 변경 둘 다 불가능

- 재선언, 재할당 X
- 블록 스코프
- 대문자 스네이크 케이스 사용
- 선언 시 값 할당해야 함
- 상수로 사용 → 대문자 스네이크 케이스

undefined: 변수에 값이 대입되어 있지 않음

데이터 타입

- 기본 데이터 타입

String, Number, Boolean, null, undefined

- 객체 타입

Object - function, array

- 새로 추가

Symbol(변경 불가능한 기본 타입)

typeof - 변수의 자료형 검사

typeof 데이터

typeof (데이터)

결과: 문자열 반환

null의 데이터 타입은 object 반환 (설계 오류)

동적 데이터 타입: 다양한 값 대입 가능

숫자형(number)

- 정수와 실수로 나누어 구분하지 않음 (부동소수점 형식)
- 일반 숫자와 특수 숫자 포함(Infinity, Nan ...)
- e를 활용하여 거듭제곱 표현 가능

문자열(String)

- “”, “로 감싸기 cf) java - “”, “는 문자 하나
- `(backtick)로 감싸기 (ES6)

여러 줄 입력 가능 - 공백, 줄 넘김 유지

문자열 내 \$(변수명) 이용하여 변수와 문자열 결합

- UTF-16 형식

문자열 연산

문자열 숫자타입 + → 문자열

문자열 숫자타입 + 외 연산 → 숫자

1+1+”2” = 22

“1”+1+2= 112

자바 스크립트 false

- null, undefined, 0, “(빈 문자열) , NaN → false로 인식
- 나머지 값은 true 예) ‘0’ → true
- !!, Boolean()으로 확인

cf) 느낌표 하나(!) → not

연산자

- +(덧셈), 단항 사용시 Number()와 동일한 역할
- -, *, /, %(나머지), **(거듭 제곱)
- = 할당 연산자
- +-, ., +=, *=
- ++, —

일치 연산자

- 값과 타입이 일치하는지 체크

===, !==

↳ 데이터 타입 비교(엄격)

조건문, 반복문

조건문

if

switch

반복문

for

while

do while

배열(Array)

- 배열의 생성: [], Array() 활용

- 배열의 크기 동적 변경
- 크기가 지정되어 있지 않아도 데이터 입력 가능
- 배열의 길이: 가장 큰 인덱스 +1
- 여러 가지 데이터 타입을 하나의 배열에 입력 가능
- push 함수 이용하여 데이터의 추가 가능

<객체 Object>

: 문자열로 이름 붙인 값들의 집합체 (Key: Value)

- Key는 문자열이어야 한다 → 단어일 때는 쌍따옴표 사용 안 할 수 있음
- 객체에 저장하는 값: property
- 객체는 prototype이라는 특별한 프로퍼티 가지고 있음

객체 만들기

- 객체 리터럴 이용 { }
- Object 생성자 이용: new Object()
- 생성자 함수 이용 function Member () { }

객체 프로퍼티

. 또는 [] 이용하여 프로퍼티 조회 및 변경

객체 만들기

객체 리터럴 이용: { }

new Object()

생성자 이용

객체 변수는 주소 저장되어 공유 가능

제이슨(JSON)

- 자바스크립트 오브젝트 노테이션

key, value 쌍으로 표현

JSON.parse(): JSON → 자바스크립트 객체

JSON.stringify(): 자바스크립트 객체 → JSON

<함수>

- 함수는 객체타입으로 값처럼 사용 가능
- 함수를 변수에 대입하거나 매개변수로 넘길 수 있다
- 배열의 요소에 넣거나 객체의 프로퍼티로 설정 가능
- 매개변수 개수 일치하지 않아도 호출 가능
- JavaScript의 함수는 일급객체에 해당
 - 변수에 할당 가능
 - 함수의 매개변수로 전달 가능
 - 함수의 반환 값으로 사용 가능

함수 만들기

- 함수 선언식 → 호이스팅 가능

```
function 함수명 () {  
}
```

- 함수 표현식 → 호이스팅 X

```
let 함수명 = function () {  
}
```

함수 선언식

함수 이름과 함께 정의

- 함수 이름

- 매개 변수
- 내용
- 호이스팅 가능

```
function func () {
    console.log('선언식');
}
```

함수 표현식

- 익명함수로 정의가능
- 매개변수
- 내용
- 호이스팅 X

선언식 vs 표현식

- 선언식 함수는 호이스팅 영향받아 함수 선언 이전에 호출 가능
- 표현식 함수는 선언 이전에 호출 불가능

함수의 리턴

- 함수 실행 결과로 함수 반환 가능
- 특별한 값 리턴 하지 않으면 undefined 반환

함수의 호출

함수의 매개변수

- 함수는 호출 시 매개변수 영향 x
- arguments 라는 함수 내부 프로퍼티 이용하여 매개변수 처리 가능
- 자바스크립트 함수는 오버로딩 지원 X
- 기본 인자 사용 가능

화살표 함수

- 함수 심플하게 정의
- 형태

(매개변수) ⇒ {명령어}

- {} 생략 시 return도 생략해야 함

DOM

1. 선택 ⇒ get, query
2. 변경 ⇒ create EI, append, append ChildWo, innerHTML, innerTEXT
 - XML, HTML 문서의 각 항목 생성, 변형, 삭제(조작)할 수 있도록 돕는 인터페이스
 - Document Object Model은 문서 요소 집합을 트리 형태 계층 구조로 HTML 표현

cf) 브라우저와 JS 소통하기 위한 모델

- HTML 문서 요소 제안하기 위해 지원
- 상단의 document 노드 통해 접근

window.print

- alert: 알림
- confirm: 확인 / 취소
- prompt: 입력
- open: 새로운 창 or 기존창
- parseInt, parseFloat
- setTimeout, clearTimeout
- setInterval, clearInterval

Event

- 중요한 사건, 일
- 웹페이지에서 여러 종류 상호 작용 있을 때마다 이벤트 발생

예) 마우스 이용, 키보드 클릭

- 이벤트 감지. 대응 작업 수행

고전 이벤트 처리 방식

인라인 이벤트 설정 -엘리먼트에 직접 지정 - 설정하려는 이벤트 정하고 on이벤트 종류 형식으로 지정'

고전 이벤트 처리 방식 2

엘리먼트에서 이벤트 직접 설정하지 않고 스크립트에서 이벤트 설정

표준 이벤트 처리 방식

이벤트요소.addEventListener(이벤트타입, 함수, [option]);

Web Storage

- Web Storage API
- 키/값 쌍으로 값을 저장
- sessionStorage
- localStorage

localStorage

- setItem(key, value)
- getItem(key)
- removeItem(key)
- clear()
- key(index)
- length
- 값은 반드시 문자열로 저장

과제: `setAttribute`, `getAttribute`, 문자열 자체 속성에서 가져오기