**JavaScript**

**<<01. 개요(ECMAScript) 및 프로젝트 초기화>>**

**\* ECMA(=ES, 에크마 스크립트)**: 자바스크립트를 표준화 하기 위해 만들어짐

- 2015년, ES6버전부터 크게 바뀜

**\* ES5와 ES6 차이점**

- 인터넷 익스플로어: 5버전 이하 지원

- 신형 브라우저: 6버전부터 지원

- Barbel과 같은 플러그인의 도움으로 ES6버전 이후의 문법을 구형 브라우저에 맞게 호환될 수 있도록 변환시키게 됨

**\* Java !== JavaScript(JS)**

**\* JavaScript 시작**

- VS코드 터미널에서 npm init -y 해주기

- index.html과 main.js 작성

- VS코드 터미널에서 npm install(혹은 i) parcel-bundler --save-dev(혹은 -D)

- VS코드 터미널에서 npm run dev //개발 서버 열음

**<<02. 데이터 타입 확인>>**

**\* 코드 컨밴션:** 관습에 따라 작성해야 가독성이 좋아짐

- 한 줄에 코드 하나만

**\* typeof:** 특정한 데이터들의 타입을 확인할 수 있음. 하지만 명확하게 내가 원하는 답이 안 나올 수도 있음.

**\* 내가 원하는 답을 얻기 위해 특정 함수 작성(getType):**

**\* 해당 함수를 다른 js소스에서 사용하려면:**

- getType함수를 따로 소스로 작성함

- export default function getType(data)… 로 선언

- export: 밖으로 내보낼 수 있음

- default: 해당 함수를 기본값으로 설정

- main.js에서 import getType from ’./getType.js’; //.js 생략 가능

- import 함수명 from ‘소스 이름’;

- test.js

**<<03. 산술, 할당 연산자>>**

**\* 산술 연산자(arithmetic operator):**

- 생략

**\* 할당 연산자(assignment operator):**

- const: 재할당 X

- let: 재할당 O

**<<04. 비교, 논리 연산자>>**

**\* 비교 연산자(comparison operator):**

- a >= b //O

- a => b //X

**\* 논리 연산자(logical operator):**

- 생략

**<<05. 삼항 연산자>>**

**\* 삼항 연산자(ternary operator):**

- a ? ‘참’ : ‘거짓’

**<<06. 조건문 If Else>>**

**\* Math.floor:** 내림 처리(= 소수점 버림)

**\* Math.floor(Math.random() \* 10):** 0~9 사이의 랜덤한 정수 나옴

**<<07. 조건문 Switch>>**

- 생략

**<<08. 반복문 For>>**

**\* 반복문(For statement):**

- for(시작조건; 종료조건; 변화조건)

**\* defer:** 현재 html 구조를 다 이해하고 실행되게 설정

**\* li 요소 추가하기**

- const ulEl = document.querySelector(‘ul’); //ulEl 가져옴

- const li = document.createElement(‘li’); //li 요소 생성

- li.textContent = ‘list- ${i + 1}‘; //li 요소의 내용

- ulEl.appendChild(li); //ulEl 요소에 Child로 li를 붙임

**\* Event Listener 설정**

**<<09. 변수 유효범위>>**

**\* 변수 유효범위(Variable Scope):** var, let, const

**\* 블록 레벨:** 변수가 선언되어져있는 블록에서 유효함(let, const)

**\* 함수 레벨:** 함수 범위 내에서 사용 가능(var)

- 의도하지 않은 범위에서 변수가 사용될수 있음, 메모리 누수로 발전 가능

- 일반적인 경우 let이나 const를 사용하는 것이 올바름

**<<10. 형 변환>>**

**\* 형 변환(Type conversion):**

**- Truth(참 같은 값):** true, {}, [], 1, 2, ‘false’, -12, ‘3.14’, …

**- Falsy(거짓 같은 값):** false, ‘’, null, undefined, 0, -0, NaN

**\* ==,** 동등 연산자: 형 변환 이후 값만 비교. 타입 비교 X

**\* ===, 일치 연산자:** 값 비교 O, 타입 비교 O.