2024\_1학기 프론트엔드프레임워크

3/6

**console.log("hello world")**

console 객체의 log 메소드는, 화면(console)에 값을 출력한다.

Java의 System.out.println 메소드와 기능이 같다.

main 함수가 필요없다

파일의 소스 코드들이 한 줄씩 순서대로 실행된다

1. **let 키워드를 사용하여, 지역 변수를 선언**

let message = "hello world"; //messsage 지역 변수에 "hello world" 문자열이 대입

javascript 언어에서는 변수를 선언할 때 변수의 타입을 지정하지 않는다

1. **var 키워드로 변수 선언**

var message = "hello world"; //var 키워드를 사용하여 변수를 선언할 수도 있다

let 키워드로 선언한 지역변수는 Java나 C의 지역 변수 문법과 같다.

var 키워드 문법이 특별하다. > 같은 이름의 변수가 중복선언되어도 에러 발생x

**호이스팅(hoisting)**

함수가 호출되면 함수 본문의 실행을 시작하기 전에,

먼저 함수에 var 키워드로 선언된 지역 변수들을 전부 미리 생성한 후 함수 본문을 실행

**3) Object.freeze(..)**

Object.freeze(객체); Object.freeze(배열); 이렇게 호출하면, 객체나 배열의 내부 값을 수정 못하게 된다. freeze된 객체나 배열을 수정하려고 할 때,

그냥 무시되고 수정되지 않을 뿐, 에러는 발생하지 않는다.

**함수**

1. **default parameter**

함수의 파라미터 변수 개수가 3개 일 때, 파라미터 값 3개 다 전달 안해도 됨.

값이 전달되지 않은 파라미터 변수의 값은 undefined이다.

값이 전달되지 않은 파라미터 변수의 디폴트 값은 function에서 지정 가능하다.

**2) 가변 파라미터 함수**

> function sum(...numbers) {

sum 함수에 전달된 파라미터 값들은 전부 배열 한 개로 묶여서,

그 배열이 numbers 파라미터 변수에 전달된다.

즉 numbers 파라미터 변수의 타입은 배열이다.

* for (let i of numbers)

numbers 배열의 값들을 하나씩 i 변수에 대입하면서

for 문의 본문을 반복 실행한다.

> function sum(numbers) // number는 배열취급 …number는 결과가 정반대

console.log(sum([4, 3, 2, 1])); // OK

console.log(sum(4, 3, 2, 1));   // 에러

sum([4, 3, 2, 1])

sum 함수를 호출하면서, 파라미터 값으로 전달되는 것은 **배열**이다.

[4, 3, 2, 1] 이 배열이 numbers 파라미터 변수에 전달되어 함수 본문이 실행.

**구조 분해 할당**

1. **배열 구조 분해 할당**

let a = [1, 2, 3]; //지역변수 a에 배열 1,2,3 할당

let [x, y, z] = a; //x,y,z 변수 생성되고 a[0], a[1], a[2] 값이 대입

console.log(x, y, z); // 1 2 3

1. **객체 구조 분해 할당**

let person = { name: "홍길동", age: 16, email: "hong@skhu.net" };

// 지역변수 person에 name, age, email 변수 생성

let { name, age, email } = person; //객체 쪼개지고 내부 속성값들 변수 대입

console.log(name, age, email); // 홍길동 16 hong@skhu.net

let person = { name: "홍길동", age: 16, email: "hong@skhu.net" };

let { name: nm, age: ag, email: em } = person;

//변수 이름과 속성 명이 일치하지 않으면 속성명 명시

console.log(nm, ag, em);

**전개 연산자**

1. **배열 분해**

let a = [1, 2];

let b = [3, 4];

let c = [5, 6];

let m = [a, b, c];

console.log(m.length) // 3

console.log(m); // [ [ 1, 2 ], [ 3, 4 ], [ 5, 6 ] ]

let n = [...a, ...b, ...c];

console.log(n.length); // 6

console.log(n); // [ 1, 2, 3, 4, 5, 6 ]