**1. 자바스크립트의 원시타입에는 어떤 것들이 있는가. 소스코드를 통해 설명하시오.**

원시 타입이란 말이 어려울 수도 있습니다만 기본 타입이라고 간단하게 생각하면 됩니다. 원시 타입의 특징 중의 하나는 불변성(immutable)입니다. 원시 타입은 복사나 전달 시에 값 자체를 복사 또는 전달하게 됩니다. 값을 전달하니 사본의 값이 따라서 바뀌는게 없는 것입니다.

**Number**

C나 Java의 경우, 정수와 실수를 구분하여 int, long, float, double 등과 같은 다양한 숫자 타입이 존재한다. 하지만 자바스크립트는 독특하게 하나의 숫자 타입만 존재한다.

ECMAScript 표준에 따르면, 숫자 타입의 값은 배정밀도 64비트 부동소수점 형(double-precision 64-bit floating-point format : -(253 -1) 와 253 -1 사이의 숫자값)을 따른다. 즉, 모든 수를 실수를 처리하며 정수만을 표현하기 위한 특별한 데이터 타입(integer type)은 없다.

추가적으로 3가지 특별한 값들도 표현할 수 있다.

Infinity : 양의 무한대

-Infinity : 음의 무한대

NaN : 산술 연산 불가(not-a-number)

let age = "20";

console.log(3 + age); // 320

console.log(3 \* age); // 60

**String**

문자열(String) 타입은 텍스트 데이터를 나타내는데 사용한다. 문자열은 0개 이상의 16bit 유니코드 문자(UTF-16) 들의 집합으로 대부분의 전세계의 문자를 표현할 수 있다. 문자열은 작은 따옴표(‘’) 또는 큰 따옴표(“”) 안에 텍스트를 넣어 생성한다. 가장 일반적인 표기법은 작은 따옴표를 사용하는 것이다.

**Boolean (논리형)**

Boolean타입의 값은 논리적 참, 거짓을 나타내는 true 와 false 뿐이다.



**Null**

null은 의도적으로 변수에 값이 없다는 것을 명시할 때 사용한다. JavaScript는 대소문자를 구별하므로 null은 Null,NULL 등과 다르다.

**Undefined**

선언 이후 값을 할당하지 않은 변수는 undefined 값을 가진다. 어떤 변수를 만들고 그 값을 정의해주지 않았을때나 존재하지 않는 객체 프로퍼티에 접근할 경우 undefined가 반환된다.

**Symbol**

심볼은 ES6에서 새롭게 추가된 타입으로 변경 불가능한 원시 타입의 값이다. 심볼은 주로 이름의 충돌 위험이 없는 유일한 객체의 Property Key를 만들기 위해 사용한다.

const s = Symbol("this is example");

const y = Symbol("this is example");

심볼은 이런 형태로 만들어 집니다. s와 y의 변수엔 각각 다른 심볼들이 들어 있습니다. 심볼은 위의 소스코드처럼 선언할 때 마다 다른 심볼들을 만들어 냅니다. 괄호 안에 적힌 것은 그저 심볼에 대한 설명일 뿐 중복되거나 해도 상관없습니다. 하지만 각기 다른 심볼을 만들어내는 특성 때문에 고유한 식별자를 만들어낼 수 있기 때문에 그 식별자에 대한 설명이 들어가는게 제일 좋습니다. 또 심볼은 아직 배우지 않았지만 객체의 키에 들어갈 수 있습니다. 키는 쉽게 설명한다면 이름입니다. 키에 심볼이 들어가면 "비공개"기능이 있어서 보통 접근하던 방식보다 좀 더 어렵게 만들어 줍니다.

**1. 자바스크립트의 Number Type은 다른 언어들과의 차이점이 무엇인가. 왜 하나만 존재하는가**

IEEE 754는 IEEE에서 개발한 컴퓨터에서 부동소수점을 표현하는 가장 널리 쓰이는 표준이다. ±0 등의 수와 무한, NaN 등의 기호를 표시하는 법과 이러한 수에 대한 연산을 정의하고 있다.

**- 자바스크립트의 숫자는 double-정확도의 부동 소수점 숫자다.**

double은 53비트까지의 정확도로 완벽하게 integer로 표현할 수 있다. -9,007,199,254,740,992(-2^53)부터 9,007199,254,740,992(2^53)까지의 모든 integer는 유효한 double 값들이다. 따라서 자바스크립트에서 integer 연산은 별도 integer형 없이도 완벽하게 가능하다

**- 자바스크립트의 정수는 별개의 데이터형이 아니라 double의 부분집합이다.**

**- 비트단위 연산자는 숫자를 32비트의 부호과 있는 integer처럼 처리한다.**

**- 부동 소수점 산술 연산의 정확도가 한계가 있음을 주의해야 한다.**

정수를 처리할 때 모든 연산이 -2^53과 2^53 범위 내에 맞춰진다는 점도 염두에 두어야 한다.

**2. null과 undefined 그리고 undeclared의 차이점은 무엇인가**

**Null**

변수를 선언하고 'null'이라는 빈 값을 할당한 경우이다.

null이라는 빈 값을 할당하면 해당 변수의 타입은 객체가 된다. undefined와 null의 차이점은 undefined는 타입이 결정되지않은 변수이고,null은 타입은 객체며, 비어있는 변수이다.

**Undefined**

접근 가능한 스코프에 변수가 선언되었으나 현재 아무런 값도 할당되지 않은 상태.

var test;

console.log(test); *//undefined*

console.log(typeof test); *//undefined*

**Undeclared (미선언 변수)**

**접근 가능한 스코프에 변수 선언조차 되어있지 않은 상태.**

console.log(test2); *// 오류를 뱉어낸다.*

*/\* Uncaught ReferenceError: test2 is not defined at <anonymous>:1:13 \*/*

*// 이것의 타입을 살펴보면 undefined*

console.log(typeof test2); *//undefined*

미선언변수 test2는 typeof 결과로는 "undefined"를 뱉는다. 실제로 typeof는 undeclared인 경우에도 undefined를 뱉어 브라우저가 오류 처리를 하지 않도록 한다.

null 또는 undefined를 검사할 때, 주의할 점

null 또는 undefined를 검사할 때, 동등 연산자(==)와 일치 연산자(===)의 차이를 주의해야한다. 동등 연산자는 자료형 변환을 수행한다.

console.log(undefined == null);  //true

console.log(undefined === null);  //false