**DB 연결 웹 앱**

**\* 자바스크립트 변수-연산자-타입**

변수는 var, let, const 키워드로 선언할 수 있다. 어떤 키워드를 사용하는가에 의해서 변수의 유효범위(scope)가 달라진다.

|  |  |
| --- | --- |
| **var** | var a= 2;  var a= “I love you”  var a= true;  var a= []; // 배열  var a= {}; // 오브젝트  var a= undefined; |
| **let** | ES6 버전 |
| **const** | ES6 버전. 자바의 상수라고 생각하면 된다 |

자바 스크립트에서의 연산자는 일반적인 사칙연산, MOD 등등 자바와 비슷하다. 자바 스크립트에서 OR 연산자는 조금 독특하다.

const name= “crong”;

/\* name에 값이 할당되어 있으면 name을 쓰고 아니면 오른쪽 값을 사용한다. \*/

const myname= name || “defaultName”

**\* 비교 연산자 주의할 점**

자바 스크립트에서는 일반적으로 == 보다는 ===을 많이 사용한다. 여러가지 문제점을 야기할 수 있기 때문이다. 아래 예를 보면,

0 == false, 0 == “”, 0 == “0”, null == undefined 의 결과가 모두 true가 나온다. 자바 스크립트에선 타입까지 정확하게 비교하기 위해서는 === 를 사용한다. == 를 사용하면 자바 스크립트가 암묵적으로 타입을 자동 변경하여 비교하게 된다.

**\* 자바 스크립트의 타입**

자바 스크립트의 타입은 선언 시점이 아닌 실행 시점에 결정된다. 타입을 체크하는 또렷한 방법은 없고 typeof, isArray 함수를 호출하여 타입을 체크할 수 있다.

|  |
| --- |
| **undefined** |
| **null** |
| **boolean** |
| **number** |
| **string** |
| **object** |
| **function** |