**<2.1> 표현식과 문장**

표현식과 문장 : 자바스크립트에서는 문장의 끝에 세미콜론을 붙여도 되고 안 붙여도 됨. 안 붙이면 자바스크립트의 인터프리터가 세미콜론을 자동으로 붙여줌. 그러나 붙이는 것을 추천.

키워드 : 암기할 필요는 없음

식별 자 : 키워드 사용 X, 숫자로 시작 X, 특수문자는 \_와 $만 허용, 공백 문자 포함 불가

주석 : // , /\* \*/ 사용

인터프리터

**<2.2> 출력**

Alert(String[message])

alert메서드는 매개변수로 문자형을 받기 때문에, alert(value)에서 value는 문자형이어야 함. 만약 다른 형의 값을 전달받으면 이 값은 문자형으로 자동 형변환됨.

**< 자바스크립트의 자료형 >**

* 문자열(String)
* 숫자(Number)
* 불
* 함수
* 객체
* Undefined

자바스크립트는 초기화를 하면 타입이 생김

**<2.3> 문자열 자료형**

큰 따옴표, 작은 따옴표 모두 사용 가능

덧셈 연산(+)으로 문자열을 이어 붙일 수 있음

문자열은 변경 불가능

객체처럼 사용가능(속성과 메서드를 가지고 있음)하다는 특징이 있음

**<2.4> 숫자 자료형**

정수, 실수 구분없이 모두 같은 자료형으로 인식됨

Integer.isInteger()로 구분은 가능함

그럼 정수와 실수의 연산 결과는 어떻게 나옴? 실수

**<2.5> 불 자료형**

true, false가 존재

불 자료형끼리 비교 연산 진행 시 true는 1로, false는 0으로 변환된 뒤 비교 진행

alert(30>20>10)의 결과는?

30>20 => true =>1 => alert(1 > 10) => false

**<2.6> 변수**

선언 : var [식별자]

자바스크립트는 동적(dynamic)언어이므로 변수의 타입을 미리 선언할 필요 없음. 프로그램이 처리되는 과정에서 자동으로 파악됨.

**<2.6.4> 변수의 특성**

var 변수는 재 선언 가능

키워드 변수(alert같은 것)를 재 선언할 경우 문제가 발생할 수 있음

**<2.7> 자료형 검사**

자료형을 확인할 때는 typeof 연산자 사용

**<2.8> undefined 자료형**

선언하지 않은 변수 or

선언했지만 초기화하지 않은 변수의 자료형

var a; -> 선언과 동시에 a의 값을 undefined로 초기화

undefined는 자료형인가 값인가?

**<2.8.1> null 과 undefined 의 구분**

중요한 건 c언어처럼 null이나 undefined의 값이 0이 아님

var str = “”;

Console.log(Number(str)) // 0

말그대로 ‘값이 없음’을 의미하는 어떤 값임

undefined : 변수를 선언만 하고 값이 할당되지 않은 변수의 자료형

Null 은 Object자료형의 빈값을 의미

0 은 Number자료형의 빈값을 의미

“”은 String 자료형의 빈값을 의미

console.log(typeof null) // Object

console.log(typeof undefined) // undefined

undefined는 자료형이고 null 은 값이다. null이라는 값의 자료형은 Object이다

var a; // undefined

var b = null; // Object

alert(a == b) // true

alert(a === b) // false

그럼 내부적으로는 같은 값을 가지는 건가?

**<2.10> 숫자와 문자열 자료형 변환**

숫자 + 문자 -> 숫자를 문자로 형변환

그 외의 모든 연산 -> 문자를 숫자로 형변환

강제형변환 함수 : Number(), String()

Number() : 숫자가 아닌 값을 입력한 경우 NaN값을 출력

**<2.11> 불 자료형 변환**

강제형변환 함수 : Boolean()

**<2.12> 일치 연산자**

== 연산자: 값만 비교

=== 연산자: 값과 자료형을 비교

**<2.13.1> 템플릿 문자열**

다루지 않겠음

**<2.13.2> let과 const키워드**

스코프 : 특정 변수를 사용할 수 있는 유효 범위

var : 프로그램 전 스코프에서 사용 가능, 재선언 가능

let : 특정 스코프 내부에서만 사용 가능, 재선언 불가

const : 변경할 수 없는 값, 지역 변수

**< 비동기 함수 문제>**

비동기란? 동시에 일어나지 않는다, 즉 요청과 결과가 동시에 일어나지 않음

var 키워드를 사용한 for 문에서 setTImeout() 을 실행 할 경우 예상대로 결과 값이 나오지 않음

**<4.2> 배열**

배열안에 입력된 값을 요소라고 함. 어떠한 종류의 자료형도 요소가 될 수 있음.

length 속성 : 배열의 요소의 개수

push() : 배열의 요소를 맨추가

< for in 반복문 >

var arr = [1,2,3,4];

for (var i in arr) {

console.log(arr[i]);

}