**Javascript (자바스크립트) ES6 ( ECMA Script6 )**

**자바스크립트 개요**

1995년 Mocha (모카) 라는 이름으로 네스케이프 커뮤니케이션즈사에서 처음 개발,

넷스케이프 네비게이터 브라우저에서 정적인 Html 문서에 동적인 기능과 함께 사용자 인터렉션 ( Interaction : 상호작용 ) 기능을 수행하는 객체지향 언어로 상용화,

2005년 Ajax 발표 이후 다양한 자바스크립트 라이브러리 언어개발과 함께 발전,

2009년 Node.js 컴포넌트 개발과 함께 2015년 Ecma International ( 자바스크립트 표준화기구 ) 에서 Ecma Script 6 표준안 출시,

이후 현제까지 클라이언트 (사용자) 스크립트 언어와 함께 서버측 스크립트언어로도 각광을 받고 있슴

크롬, 사파리, 파이어폭스, 오페라 브라우저에서는 ES6 버전을 97% 이상구현

**인터넷 익스플러러 11 버전까지 ES6 버전 11% 밖에 구현 못함**

ES6 버전 각 브라우저별 구현 현황 <https://kangax.github.io/compat-table/es6/>

**자바스크립트 특징**

- 동적인 프로토타입 기반의 객체 지향언어로 다른 언어에 비해 개발하기 쉽다.

- 프로그램언어를 동기적으로 브라우저에서 바로 번역해서 실행시키는 인터프리터 언어이다.

- 함수형 프로그래밍 언어이다.

- 클라이언트 (사용자) 스크립트 언어와 함께 서버측 스크립트언어로도 발전

**자바스크립트 문법 ( Core )**

**1) 변수(Variable) 와 데이터 타입**

**변수정의**

변수(Variable) : 특정한 변수에 임의의 데이터값을 대입하여 수시로 그 데이터값을 변경하면서 사용할수 있는 수

- 데이터 효율적 관리

**형식**

var 변수명 = 대입할 데이터값

**호출**

변수명만 호출 ( 같은 변수를 사용할 때는 var 선언을 한번만 사용 )

**변수명 규칙**

* 사용자가 지정, 반드시 영문, 대소문자구분, 숫자로 끝날수있으나 숫자로 시작할수 없다, 띄어쓰기나 불필요한 기호들 ( ( ) , . 등) 을 사용할수없다, \_ 기호는 사용가능, 예약어를 변수명을 사용할수없다,

**여러 개의 변수 사용과 데이터값의 종류**

문자 ( “ “ 붙임 ) , 숫자 ( 소수점을 포함한 실수 ) , 논리값 ( true, false ) , **undefined, null**, 함수, 클래스, 인스턴스

- 변수는 여러 개 사용이 가능

예1)

var a=”임의의 문자열” ; ( 문자열은 큰따옴표사용 “ ” 또는 ‘ ‘ 작은따옴표 사용, 끝에 세미콜론 ; 생략가능 )

var b=3.14 ; ( 소수점을 포함한 실수 )

var c=true ; ( true, false 논리값 )

- 여러 개의 변수선언을 가로로 코딩

예2) var a=”임의의 문자열” , b=3.14, c=true ( var 은 한번만 사용하고 콤마로 구분 )

var d;

console.log(d) 결과: **undefined** : 변수에 값이 정의 되지 않음

var d=1;

console.log(d) 결과 : 1

d=**null**; 변수값을 초기화

**undefined : 변수값이 정의 자체가 되어 있지 않은 것 , null : 아무것도 없는 값으로 정의 되어있는 것**

그 외 데이터타입은 함수 사용후 설명

**주의) 같은 변수명을 다시 선언하면 최근에 선언한 변수값으로 지정**

예) var gender=”여자”;

var gender=”남자”

console.log(gender) 결과는 남자 ( 변수 충돌 )

**ES6 버전에서 추가된 변수 선언자**

**let** : 같은 변수명을 한번만 사용 ( 중복되면 에러 )

전역변수 : 함수 밖에서 선언하여 어떤 함수에서도 사용가능 한 변수

지역변수 : 함수 안에서 선언하여 해당 함수 안에서만 사용가능 한 변수

**주의) 함수 안에서도 var 을 생략하면 전역변수로 지정되어 불편함**

**- let 을 사용**

함수 밖에서 선언하면 전역변수로만 사용가능

함수 안에서 선언하면 지역변수로만 사용가능

**전역변수, 지역변수** – 함수 안 에서의 변수사용 - 함수 설명후 연습

**const** : 같은 변수명과 값도 한 번만 사용 ( 중복되면 에러 )

**임의의 문자열에 개행 이나 기호 사용**

이스케이프 스퀀스 사용 ( \역슬래시 ) : 문자열 개행이나 다양한 기호를 문자열에 삽입

a=”임의의\n문자열” ( 문자열개행 )

a=”임의의\”문자열” ( “ 기호사용 )

a=”임의의\’문자열” ( ‘ 기호사용 )

a=”임의의\\문자열” ( \ 기호사용 )

**ES6 버전 ` ` ( 역따옴표 ) 사용 ( \ 없이 문자열 개행이나 여러가지 기호를 사용할 수 있다. )**

a=`임의의

문자열`

a= `임의의 “문자열” `

**ES6 버전에서 추가된 데이터 타입 Symbol()**

**Symbol() :** 자기 자신을 제외한 그 어떤 값과도 다른 유일무이한 상징적인 값 ( 변수명이 다르더라도 각 변수에 저장되어 있는 값이 같을 때 변수가 충돌되는 현상을 막음 )

예) 동영상 작동 상태를 나타낼 때

var isPlay;

isPlay=1 ( 플레이 상태 ) ;

isPlay=0 ( 일시정지 상태 )

isPlay=2 ( 꺼진상태 )

다른변수의 데이터값이 1, 2, 3 중에 하나일 때 변수가 서로 충돌

**이때 데이터 값 Symbol()을 사용하면 데이터값이 같다고 하더라도 변수값이 서로 다른 변수가 된다.**

예) var isPlay;

isPlay=**Symbol(1)**

isPlay=Symbol(2)

isPlay=Symbol(3)

**문자열을 Symbol() 값으로 사용할 때**

isPlay=**Symbol.for**(“play”);

isPlay=Symbol.for(“pause”)

isPlay=Symbol.for(“stop”)

**변수값 호출**

console.log**(Symbol.keyFor**(isPlay))

**객체(클래스) 에서 Symbol() 사용 – 독립된 객체(클래스)로 지정가능**  - 객체(클래스) 수업후 설명

**2) 여러가지 연산자 ( Operator )**

|  |  |  |
| --- | --- | --- |
| **구분** | **연산자** |  |
| 사칙연산자 | + - \* / % |  |
| 문자연산자 | + |  |
| 복합연산자 | += -= \*= /= %= |  |
| 증감연산자 | ++ -- |  |
| 비교연산자 | > >= < <= == != ! === |  |
| 논리연산자 | && || |  |
| 조건연산자 | ? : |  |
|  |  |  |
|  |  |  |

사칙연산자 ( 일반적인 사칙연산 )

a%b : a를 b로 나눈 나머지 값 ( 어떤수%2 결과값이 0 이면 짝수, 1 이면 홀수 )

**+ 문자연산자**

**문자 + 문자 또는 문자 + 숫자 연산하게 되면 + 연산자가 서로 이어주는 역할 ( 결과는 문자 )**

숫자 + 숫자 = 숫자 , 문자 + 숫자 = 문자, 문자 + 문자 = 문자

1+2 결과 3 (숫자)

“잘생기고 멋진” + “스크립트 선생님” 결과 : “잘생기고 멋진 스크립트 선생님” (문자)

“Visual Studio Code” + 1.39 결과: “Visual Studio1.39” (문자) **1.39 숫자가 강제 항변환**

**항변환 : 변수에 대입되는 데이터값의 데이터 타입을 묵시적 또는 강제적으로 변경**

묵시적 항변환

숫자 + 문자 = 문자

true + 숫자 = 숫자 예) true + 1 결과) 2 (숫자) true 숫자1 , false 숫자 0

강제적 항변환

String() 숫자를 문자로 변환

예) String(true) + 1 결과) “true1”

Number(), 문자를 숫자로 변환,

예) “2019” + 10 결과) “201910” (문자)

Number(“2019”) + 10 결과) 2029 (숫자) 숫자로 변환되어 연산됨

parseInt() 문자나 또는 숫자(소수점) 중에 정수만 숫자로 변환

2019년 + 10년 결과) 2019년10년 (문자)

parseInt(2019년) + parseInt(10년) 결과) 2029 (숫자)

**변수와 문자연산자+ 활용**

“홍길동 님 환영합니다.”

let name=”홍길동”

name + “님 반갑습니다.”

“ 당신의 점수는“ + 98 + “점 입니다.”

let score=98;

“당신의 점수는” + score + “점 입니다.”

**ES6 버전 ` ` 역따옴표와 ${} 보간식 사용**

**( +연산자를 사용하지 않고 변수나 또는 연산식 등을 문자열에 넣을수있다. )**

**`당신의 점수는 ${score} 점 입니다.`**

**복합연산자** (연산자를 함축해서 사용 )

a=a+10 >> a+=10, a=a-10 >> a-=10, a=a\*10 >> a\*=10

**증감연산자** ( 1 씩 증가하거나 1씩 감소하는 연산자를 함축해서 사용 )

후위연산자 ( 대입을 먼저한후 증감 ) a=a+1 >> a++ a=a-1 >> a--

전위연산자 ( 증감을 먼저하고 대입 ) a+1=a >> ++a a-1=a >> --a

**후위연산자와 전위연산자의 차이**

a=10 a++ a 결과값: 11 a=10 ++a a 결과값: 11 a값이 서로 같다.

다른 변수에 대입 할 때는 차이

a=10 b=a++ b결과값 : 10 ( b값에 a을 먼저대입하고 나중에 더한다.)

a=10 b=++a b결과값 : 11 ( a값을 먼저 더하고 b값에 대입한다.)

**비교연산자**

= 연산자 == 연산자 === 연산자의 차이

a=b ( a값에 b값을 대입 의미 ) a==b ( a값과 b값이 서로 같다 의미 )

a===b ( a값과 b값이 서로 같고 변수타입 까지 같다 )

a!=b ( a 값과 b값이 서로 같지않다 )

a=!true 결과: a 값은 false ( ! 반대값 )

**논리연산자**

&& ( 조건식에서 and 조건 ) || ( 조건식에서 or 조건 )

a && b ( a값과 b값 둘다 참일 때만 참인 값 ) a || b ( a값 또는 b값 둘중에 하나값이 참일 때 참인 값 )

**조건연산자 ( 3항연산자 )**

조건 **?** 참일때 값 **:** 거짓일때 값

var a=10, b=20

var result = (a>b) **?** “맞아요” **:** “틀려요”

console.log(result) 결과 : “틀려요”