**자바스크립트의 원시데이터형태(가장 초창기에 만들어진 타입)**

**(1) String – 문자데이터** “가” + ’가‘ + `가` = 가가가

**(2) Number – 숫자데이터** 1+2+3=6 (\* / - %가능) %(나머지연산-나머지값)

**(3) Boolean – 참 거짓 값 데이터 형태**

**- 비교를 통한 참거짓 (주의 : ! = not, == 같다. )**

**- 사실에 근거 참거짓 (숫자데이터 0, 빈문자인 문자데이터, 초기화안한변수)**

**- 논리연산자 &&** - and 좌우둘다 true라면 전체가 true

**- 논리연산자 ||** - or 좌우하나라도 true라면 전체가 true

**- 삼항연산자 – (조건?”참값”:”거짓값”)**

**- 수학관련도구 - Math객체함수 console.log(Math.round(값)) 반올림**

**복합데이터형태**

**참조데이터형태**

**객체데이터형태**

(기본형태) let 변수 = {key:value(=property), key:value, key:value}

(예시) let person = {name(key):”손나라”(value), age:33, gender:”여자”, married:false}

(부르는방법) alert(person1[”name”]), alert(person1변수.name키)

Key는 객체 안에서 변수로 사용된다.

--------------------------------------------------------------------------------------------------------------

**typeof()** – 괄호 안의 데이터 형식 확인 도구 (문자 or 숫자)

**prompt()** – 질문 후 답변을 그 자리에 문자데이터로 남기는 도구

**Number()** – 진짜 숫자데이터로 변경 도구

**document.write()** – 그대로 필기해주는 도구

--------------------------------------------------------------------------------------------------------------

**상수** : 변하지 않는 수

**변수** : 변할 가능성이 있는 수

--------------------------------------------------------------------------------------------------------------

**•if조건문**

- if(조건){소스코드}

- if(조건){소스코드1}else{소스코드2}

조건이 참이라면 소스코드1을 해석하고, 거짓인 경우 소스코드2를 해석.

- if(조건){소스코드1}else if(조건2){소스코드2}else{소스코드3}

조건1이 참이라면 소스코드1을 해석, 거짓이면 조건2가 참이라면 소스코드2를 해석, 거짓이면 소스코드3을 해석.

--------------------------------------------------------------------------------------------------------------

**• while반복문 (조건이 참인동안 소스코드를 반복한다.)(**보안약함. 바깥쪽에서 접근가능함.)

- while(조건){반복소스코드}

- <ul><script> let n = 0; while(n<9) {n++; document.write(`<li>2x${n}=${n\*2}</li>`)} </script></ul>

**•while반복문의 중첩**

- while(조건){반복소스코드 while(조건){반복소스코드}}

- let a=0; while(a<3) {a++; let b = 0; while(b<2) {b++; document.write(`a의 값은 ${a}이고 b의 값은 ${b}이다.<br>`)}}

--------------------------------------------------------------------------------------------------------------

**• for반복문** (while보다 안전성이 더 좋음. 바깥쪽에서 접근 못하고, 안쪽에서만 접근 가능함.)

– for**(**반복변수;조건식;증감식**){**반복소스코드**}**

- for (let t = 0; t <= 2; t++){document.write(t+”<br>”)}

**• for반복문의 중첩**

- for**(**반복변수;조건식;증감식**){**반복소스코드 for**(**반복변수;조건식;증감식**){**반복소스코드**}}**

- for ( let a = 1; a <= 9; a++ ) { document.write(`<ul>`)

for (let b = 1; b <= 9; b++) {document.write(`<li>${a}x${b}=${a\*b}</li>`) } document.write(`</ul>`)}

<증식> - j에 1을 더한 값을 적용한다.

J=j+1 / J+=1 / J++

<감식> - k의 1을 뺀 값을 적용한다.

K=k-1 / k-=1 / k—

--------------------------------------------------------------------------------------------------------------

**• 배열 – array**

- let 변수이름 = [원소0, 원소1, 원소2, [원소3-1, 원소3-2,,,], 원소4, 원소5]

- 조건식의 변수 뒤에 .length 붙여주면 변수개수만큼 배열개수 늘려줄 수 있음.

--------------------------------------------------------------------------------------------------------------

**• 함수 -function**

- function 함수명(재료){동작할 소스코드}

--------------

Function a ( ) {

alert(“hi”)

}

a( )

Function a( ) 함수의 정의부

{alert(“hi”)} a라는 함수가 실행될 때 동작하는 소스코드

a( ) 함수의 실행부

- ( ) 실행한다는 의미

-------------

함수의 매개변수(재료) 설정

function b (name) { alert (name) }

b(“손나라”)

b(“오나라”)

-------------

매개변수의 개수는 제한이 없다.

function c (num1, num2) {

alert (num1 + num2)

}

c ( 1, 2 )

let number = Math.round(32.1)

let name = prompt(“당신의 이름을 입력해주세요”)

let result = document.write(“test”)

alert(result)

function으로 만드는 함수는 끌어올려진다(호이스팅 가능)

a( );

function a( ){alert(“A함수 실행함”)}

변수로 만드는 함수는 끌어올려지지 않는다.(호이스팅 불가)

a( );

let a = function ( ){alert(“A함수 실행함”)}