



7_6_2022_js__dataType10-example03

Javascript Step01_dataType09.html
~ Step01_example03.html

함수 타입

function type

```
number, string
let num1=10;
let num2=10.1;
let myName='dubu';
let yourName="yuki";
let orName=`soju`;
```

boolean - 참 거짓

```
let isRunning=true;
let isWaiting=false;
```

number, string, boolean은 값 value 가 명확함

object - 순서 상관없이 묶음으로 관리 key값으로 정리
여러개의 type이 섞여있으면 object를 사용하는게 좋음.
console은 오브젝트 타입

```
let obj={};
let mem={num:1, name:"dubu", addr="Fairfax"}
      {key:value, key:value, key:value}
```

array - 순서대로 묶음으로 관리 index값으로 정리
한가지 타입으로 순서를 줄 때 사용하면 좋음.

```
let nums[10,20.30.40...];
let names["두부", "유키", "소주"...]
let members=[{}, {}, {}...]
```

function - 특정 시점에 실행할 js를 저장해놓음
참조와 호출 구분하기

```
f          f()
let f=function(){ javascript 모아놓기 };
f() 로 {}안의 js를 실행
function f2(){ }
f2()
let f3=()=>{};      화살표 함수
```

heap 영역 (object, array, function type)
사물함 영역 참조값 reference value 가 들어있다
겹치는 값들이 다른 영역의 키가 들어있을 수도 있음.

```
array[f1(),f2()...]
```

stack 영역 (앞에 지어준 이름들이 저장됨)
stack 영역은 value, reference value들과 연결되어 있다.

매개변수

대입연산자를 기억하면 편함
두개 이상 들어갈 수 도 있음.

document object 문서 객체 XXX. 식으로 참조할 수 있음
필요하다면 접근할 수 있음.
첫번째 p element를 조작하기 위해서는 그것의 key를 알아야함

```
document.querySelectorAll("p")
p요소의 참조값이 배열 순서대로 있다.
document.querySelectorAll("p")[0] 0번방 참조할 수 있음
document.querySelectorAll("p")[0].innerText 이너 텍스트 보기
document.querySelectorAll("p")[0].innerText="Dubu" 이너 텍스트를 Dubu로 바꾸기
```

```
document.querySelector
배열을 불러오지 않는다
가장 첫번째만 불러옴
document.querySelector(" ") 괄호 안에 선택자를 입력하면
그 선택자를 불러와서 실행함.
```

```
function plus(){
    //입력한 값을 읽어온다
    let a=document.querySelector("#num1").value;
    let b=document.querySelector("#num2").value;
    //연산의 값을 얻어낸다
    let result=Number(a)+Number(b)
    // 종종 문자열 값을 숫자로 바꿔줄 필요가 있다
    document.querySelector("#result").innerText=result;
    //span 요소의 innerText로 출력하기
```