



# JS Study

제목 없는 데이터베이스

## #1 Basic of JavaScript

### *Variable*

종류 및 선언 방식

**const, let, var**

const 와 let의 차이점, var 사용하면 안되는 이유 > **항상 const, 업데이트가 필요한 변수는 let, var은 사용x**

boolean(true, false), null, undefined

let a = 5; 와 a = 7 의 차이 > 왼쪽은 변수를 생성과 동시에 값을 초기화 & 오른쪽은 변수의 값을 업데이트

### *Arrays*

선언 방식, 값을 가져오는 방법, 값을 추가하는 방법(push, unshift),

## Objects

array 와 objects의 차이점

array에 item을 넣을 때는 item마다 의미를 넣기 힘들지만 objects는 item마다 의미부여 가능!!

console은 objects

const 는 update 불가

const로 objects 를 선언 후 const를 다른 type으로 변환 불가

const objects = { } 선언 후 objects = boolean (x) but!! constant(objects) 안의 속성을 update하는 것은 가능!!

objects에 속성을 추가 objects.properties = "value";

## Function

function 은 어떤 코드를 캡슐화해서 실행을 여러번 가능하게 한다.

( ) 는 function을 실행하는 방법

argument는 function을 실행하는 동안 데이터를 보내는 방법 > ( ) 안에 argument 값을 넣어서 function에 데이터를 전송

argument의 순서 중요

예) 필요한 argument가 하나만 존재할 때 전달되는 argument가 여러개가 전달될 경우, 제일 첫번째 argument를 값으로 받고

그 뒤에오는 argument는 받지 않는다.

argument 명은 지역변수로 존재

argument는 function의 body { } 안에서만 사용 가능 body 외부에서 접근 불가

objects 안에 function 선언

```
const objects = {  
  function명: function (argument) {  
    내용  
  }  
}
```

```
}  
}
```

## Returns

console.log를 사용하지 않고, function이 계산의 결과를 나에게 제공  
console.log나 alert를 사용하면 결과를 console창이나 페이지 경고창으로 보여주고 끝,  
function을 사용하는 궁극적인 목적은 function의 결과를 가지고 운영하는 데 있다.  
return을 사용하지 않고, console.log를 사용하면 function의 결과 값은 undefined  
하지만 return을 사용하면 그 계산 결과를 계속해서 활용 가능

```
const calculator = {  
  plus: function (a, b) {  
    alert(a + b);  
  }  
};  
  
console.log(calculator.plus(2,3))  
실행 결과 > 실제 calculator.plus의 값은 undefined
```

```
const calculator = {  
  plus: function () {  
    alert("hi");  
  }  
};  
  
console.log(calculator.plus)  
실행 결과 > function 명시
```

!!return 사용 시 function은 종료

```
const calculator = {  
  plus: function (a, b) {  
    console.log("hello")  
    return a + b  
    console.log("bye bye")  
  }  
};
```

```
}  
}  
실행 시 console.log("hello") 는 실행되지만, console.log("bye bye")는 실행안됨
```

## Conditionals

```
const age = parseInt(prompt("How are you?"));  
  
console.log(isNaN(age));
```

### ▼ 사용 function

prompt ( ): 사용자에게 입력을 받는 function (아주 오래된 사용 방법)

css 적용 불가, 브라우저에서 기본으로 지원

typeof : value의 type을 확인 하기 위한 function

parseInt ( ) : String type의 value를 Number type으로 변환

isNaN ( ) : 무엇인가 NaN인지 판별하는 function

```
if(condition) {  
  // condition === true  
} else {  
  // condition === false  
}
```

**condition(조건)은 boolean type 이어야 한다.( true or false )**

```
if (isNaN(age) || age < 0) {           //조건문 ||(OR) 은 둘 중 하나만 TRUE면 TRUE
    console.log("Write A Number");
} else if (age < 18) {
    console.log("you are too young.");
} else if (age >= 18 && age <= 50) {    //조건문 &&(AND) 는 둘다 TRUE 일 때 TRUE
    console.log("you can drink");
} else {
    console.log("you can't drink");
}
```

else 는 선택사항

**= & == & === !==**

= : 하나의 value를 할당 하는 것

=== : equals 같은지 판단

---

## #2 JavaScript On The Browser

JavaScript는 HTML에 접근하고 읽을 수 있게 설정되어있다.

브라우저가 HTML 정보가 아주 많이 들어 있는 document 라는 object를 전달해 준다.

### Document

document 인터페이스는 브라우저가 불러온 웹 페이지를 나타내며, 페이지 콘텐츠(DOM트리)의 진입점 역할을 수행.