

# 4강. 자바스크립트&제이쿼리

## 데이터타입

### 수(Number)

1+2

### 문자열(String)

"1+2" / "true"

### 불린 (Boolean)

true / false

// Number + Number -> 합

1 + 2 // 3

// Boolean + Number -> 합

true + 1 // 2

// Boolean + Boolean -> 합

false + false // 0

// Number + String -> 연결

5 + "foo" // "5foo"

// String + Boolean -> 연결

"foo" + false // "foofalse"

// String + String -> 연결

"foo" + "bar" // "foobar"

**연산자**(+, -, /, \*, %, ++, --)**% (나머지)**

```
17 % 5 // 2
```

**++ (증가)**

```
var a = 1  
a++ // 2
```

**-- (감소)**

```
var a = 1  
a-- // 0
```

**변수**(Variable) – 변하는 수, 값을 담는 그릇**문자열**

```
var a = "가나다";
```

**숫자**

```
var b = 100;
```

**불린**

```
var c = true;
```

**배열**(Array) – 다수의 데이터를 저장**배열선언**

```
var arr1 = [];  
var arr2 = [1, "가", 10];
```

**값 추가**

```
arr1.push(20)  
arr2.push("나")
```

**호출**

```
arr1[0] // 20  
arr2[1] // "가"
```

**함수(function)** – 하나의 로직(프로그래밍)을 재실행 할 수 있도록 함.

```
function 함수명(){  
    // 코드  
}  
함수명() // 코드 실행
```

**조건문(if)** – 주어진 조건에 충족하면 실행시킴

```
if(10>5){  
    // 참일경우 실행하는 code ...  
}
```

```
if(10>5){  
    // 참일경우 실행하는 code ...  
}else{  
    // 거짓일경우 실행하는 code ...  
}
```

**반복문 (for)** – 반복적인 작업을 지시함.

```
console.log(0);  
console.log(1);  
console.log(2);  
console.log(3);  
console.log(4);
```

```
for(var i=0; i<5; i++){  
    console.log(i)  
}
```

## jquery란

자바스크립트의 **생산성을 향상**시켜주는 자바스크립트 라이브러리

## 사용법

```
<script src="https://code.jquery.com/jquery-3.2.1.min.js"></script>
<script>
  $ (function(){ [
    // code ...
  } );
</script>
```

## 선택자 – css 선택자와 동일

```
$(".btn > span") / $(".btn").eq(2) / $(".btn").next() / $(".btn").parent()
$(".btn").find("span")
```

## css – 동적으로 inline-style 적용가능(메소드)

```
$(".btn > span").css("border", "1px")
$(".btn").next().css({"border":"1px", "background":"#ff0000"})
```

**class관련** – class를 동적으로 추가, 삭제 함.

```
$(".btn").addClass("on") // on class 추가  
$(".btn").removeClass("on") // on class 삭제  
$(".btn").toggleClass("on") // on class 추가&삭제
```

**slideUp, slideDown, slideToggle** – 슬라이드 애니메이션 효과.

```
$(".box").slideUp();  
// 선택한 요소가 서서히 올라가면서 사라지는 효과를 보여준다.  
$(".box").slideDown(500);  
// 선택한 요소가 서서히 내려오면서 나타나는 효과를 보여준다.  
$(".box").slideToggle();  
// 선택한 요소의 표시 상태에 따라 다른 동작을 하는 .slideToggle() 메소드를 사용할 수 있다.
```

**setInterval** – 설정시간 간격을 두고 반복 실행

```
setInterval(function(){  
    // code ...  
,1000)
```

**animate** – 선택한 요소를 설정한 시간동안 애니메이션 처리

```
$(.box).animate({  
    width:"200px"  
,1000)
```

**click** – 클릭 이벤트가 발생 했을 경우

```
$(".btn").click(function(){  
    // code ...  
})
```

**keypress** – 선택한 요소에서 키보드를 눌렀을 경우 발생

```
$(".input").keypress(function(event){  
    console.log(event.keyCode)  
})
```

**scroll** – 선택한 요소에서 스크롤을 위아래로 이동 시켰을때 발생

```
$("html").scroll(function(){  
    console.log( $(document).scrollTop() );  
    // 수직위치 출력  
})
```

**scrollTop** – 선택한 요소의 수직위치를 알아내거나 특정값으로 이동시킴

```
$("html").scrollTop(0);
```