

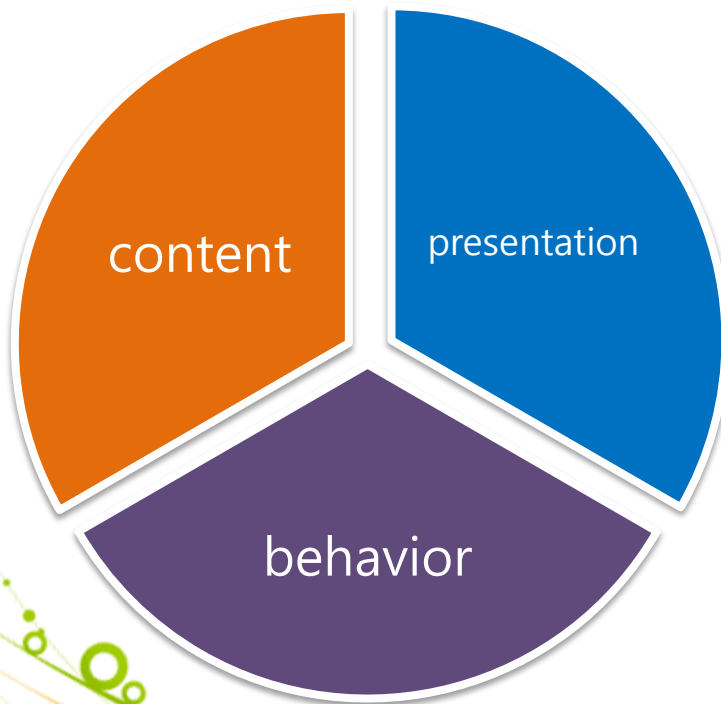
jQuery Study

장영석



WEB = HTML + CSS + javascript

웹은 구조(content) / 표현(presentation) / 동작(behavior)로 구성



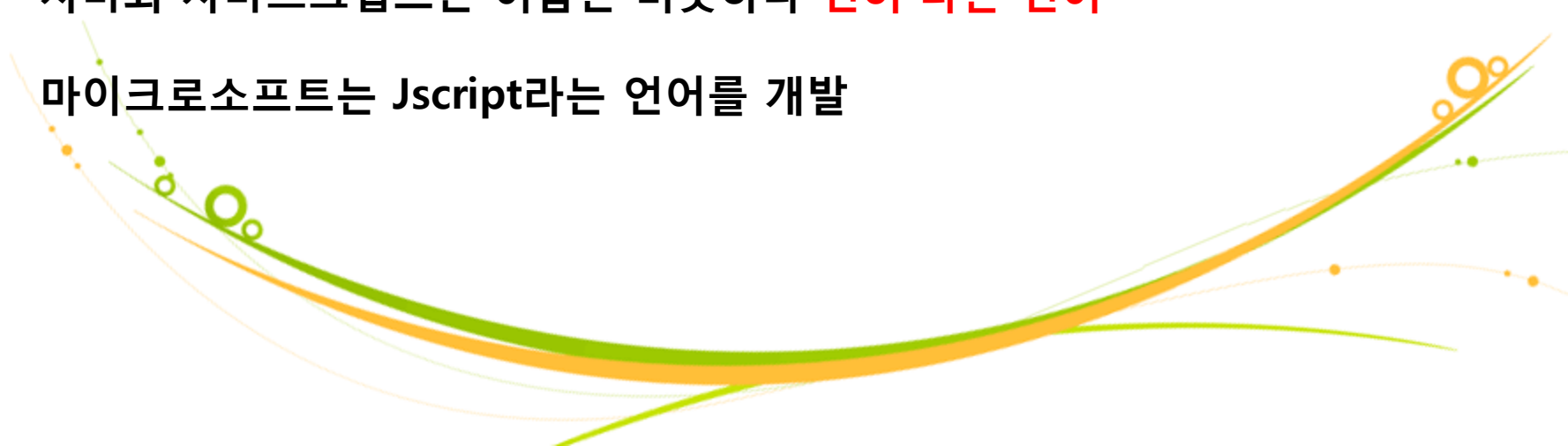
구조(content)는 **HTML**,

표현(presentation)은 **CSS**,

동작(behavior)은 **javascript**

Javascript

- 정적인 문서인 HTML에서 **동작** 할 수 있게 해주는 프로그래밍 언어
- 객체기반의 스크립트 프로그래밍 언어
- 넷스케이프사에서 자사의 웹브라우저인 Netscape 브라우저에서 동작하는 언어로 LiveScript 개발
- 썬 마이크로시스템즈와 넷스케이프가 공동으로 작업시 Javascript로 명칭 변경
- 자바와 자바스크립트는 이름은 비슷하나 **전혀 다른 언어**
- 마이크로소프트는 Jscript라는 언어를 개발



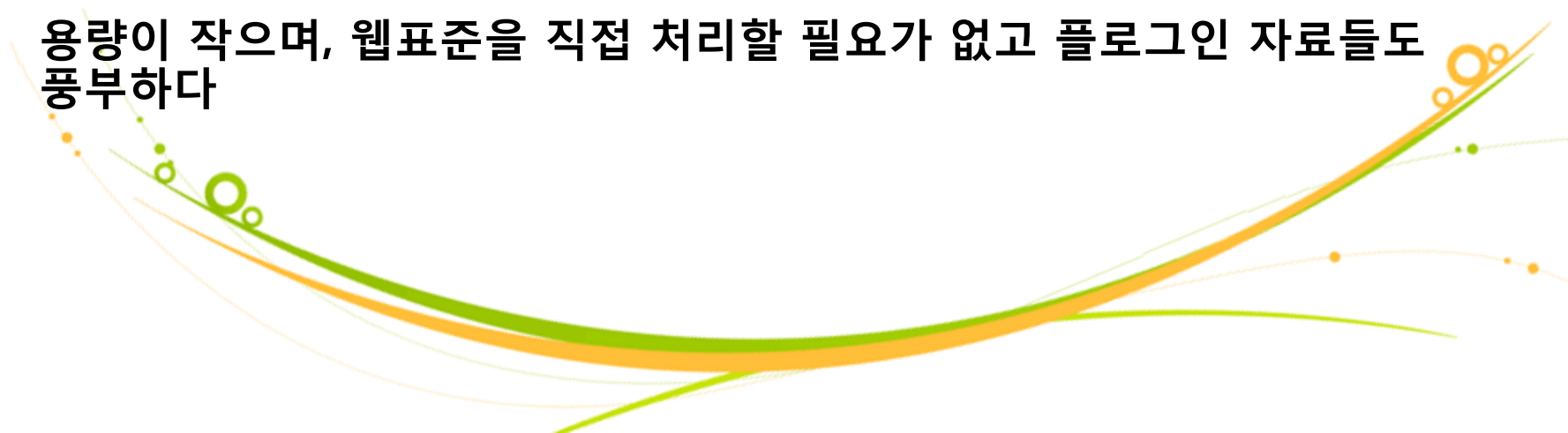
Javascript의 특징

- JavaScript는 컴파일할 필요가 없는 인터프리터 언어
- JavaScript는 웹브라우저의 자원을 활용하고 제어
- 장점 : 생산성, 운영체제 탈피, 쉬운 구문
- 단점 : 소스 코드 노출, 한정된 객체와 메소드



jQuery

- 문서 개체 모델(DOM)과 자바스크립트 사이의 상호작용을 간단하게 해주는 오픈 소스 자바스크립트 라이브러리
- HTML문서를 간단히 탐색하고 조작하며, 브라우저 이벤트 처리, DOM 애니메이션, Ajax 상호작용 및 자바스크립트 크로스 브라우저에 개발에 용이
- jQuery는 개발자가 플러그인을 개발할 수 있는 기능을 제공도록 해주는 오픈소스형태(무료)
- 용량이 작으며, 웹표준을 직접 처리할 필요가 없고 플러그인 자료들도 풍부하다



문서객체모델(DOM)

- Document Object Model

- Document(문서) :

- 웹문서를 만들어 웹브라우저에 띄우는 순간부터 DOM이 시작된다.

- Object(객체)

- 사용자 정의 객체,
 - 배열 · 수학 · 날짜와 같이 자바스크립트에서 이미 만들어진 객체,
 - 웹브라우저가 제공하는 주요 객체

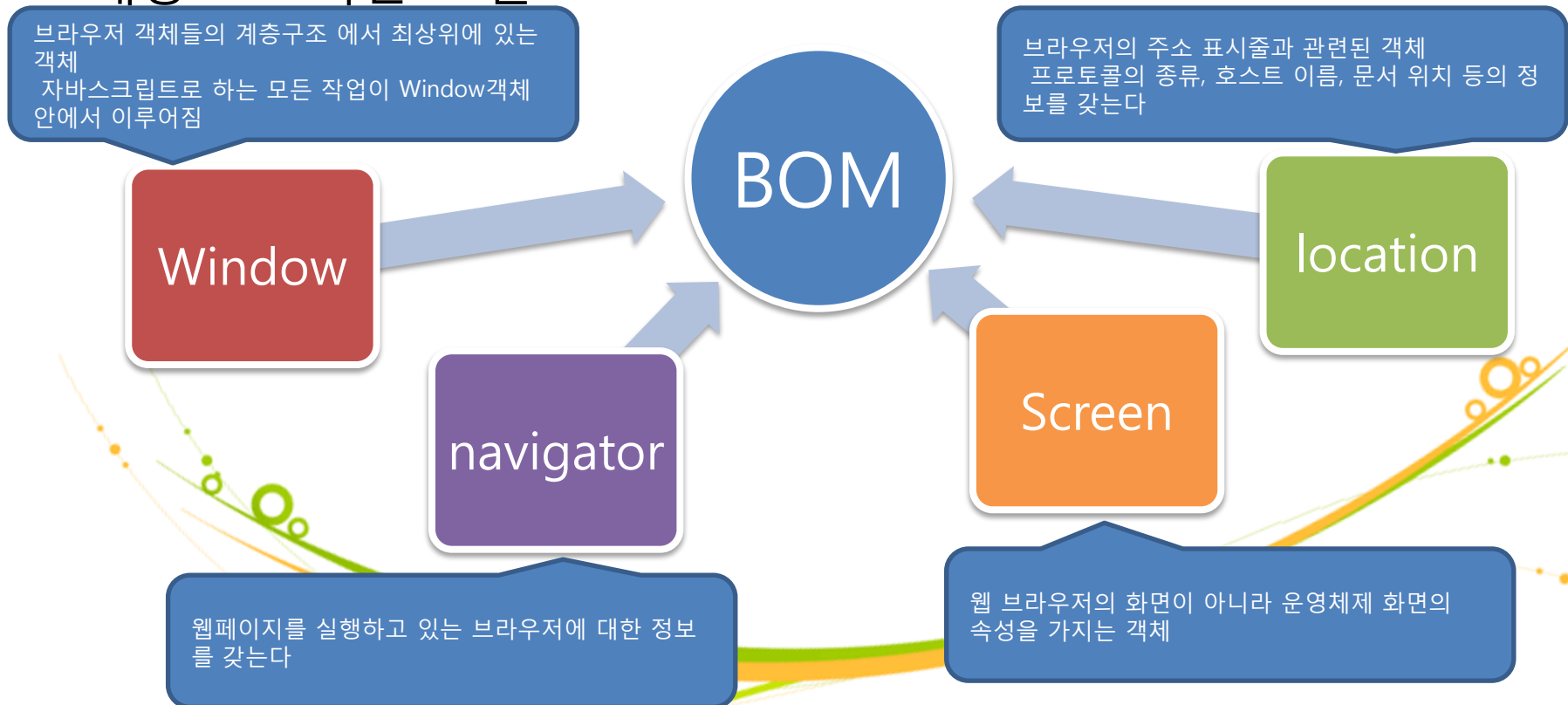
- Model(모델)

- tree구조로 표현하는 것



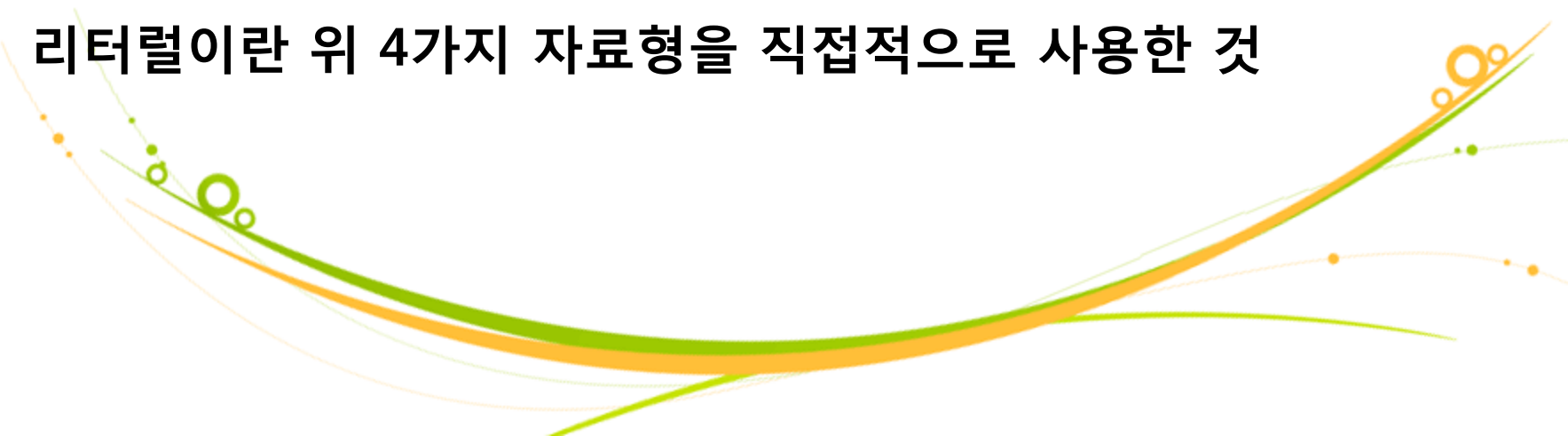
문서객체모델(BOM)

- Browser Object Model
- 웹 브라우저 창을 관리할 목적으로 제공되는 객체 모음을 대상으로 하는 모델



jQuery - 자료형

- 프로그램에서 다양한 정보를 표현해야함
- 그 정보들을 프로그램에서는 자료형으로 분류함
- 숫자형, 논리형, 문자열형, null형
- 변수나 리터럴로 표현
- 리터럴이란 위 4가지 자료형을 직접적으로 사용한 것



자료형 - 숫자형

- 내부적으로 64비트 부동소수점 형식
- 자바의 double형
- 정수 / 실수 구분이 없음
- 숫자형으로 인한 오류를 피할 수 있음



자료형 - 문자형

- 이름, 문장, 혹은 어떤 문자를 나타내기 위해
- 문자열은 따옴표로 감싸여진 글자들의 집합
- 한 글자 or 여러 글자가 합쳐진 문장
- 큰 따옴표와 작은 따옴표 사용 / 혼용 불가



자료형 - 논리형

- 숫자와 문자열은 거의 무한한 표현이 가능한데 비해, 논리형은 단순히 true, false로만 값을 가짐
- "true" "false" 는 문자열(논리형이 아님)
- false 값 : false / null / undefined / "" / 0 / NaN



변수

- 프로그램에서는 많은 정보를 저장해야 하는 경우가 많음
- 계속 변하는 정보를 담아놓을 수 있는 저장공간이 바로 변수
- 변수의 선언 : `var index;`
- `var`는 변수 선언 키워드
- `index`라는 변수가 만들어짐
- 선언된 지역에 따라 전역변수 / 지역변수



변수 명명 규칙

- 변수명은 영문자, \$, _로만 시작할 수 있음
- 변수명은 문자, 숫자, \$, _만 포함할 수 있음
- 한글로 변수명을 만들 수 없음
- 변수명은 대소문자를 정확히 구별함
- 자바스크립트 내에서 먼저 정의된 키워드는 사용불가



jQuery - 함수

- 무언가의 동작
- 함수의 선언 : `function` 함수명(인자){
 //실행
}
- 함수의 호출 : 함수명(인자)



jQuery - 연산

- 기본연산

연산자	역할	사용법
+	두 숫자를 더함, 문자열을 합침	6+12
-	한 숫자에서 다른 하나를 뺌	7-4
*	두 숫자를 곱함	50*13
/	한 숫자를 다른 숫자로 나눔	14/5



jQuery - 연산

- 특징적인 연산

연산자	역할	사용법
++ or --	증감연산자	++10
%	나머지 연산자	10%3
==	항등연산자	
&& or	And 또는 or 연산자	A && b
+= or -=		



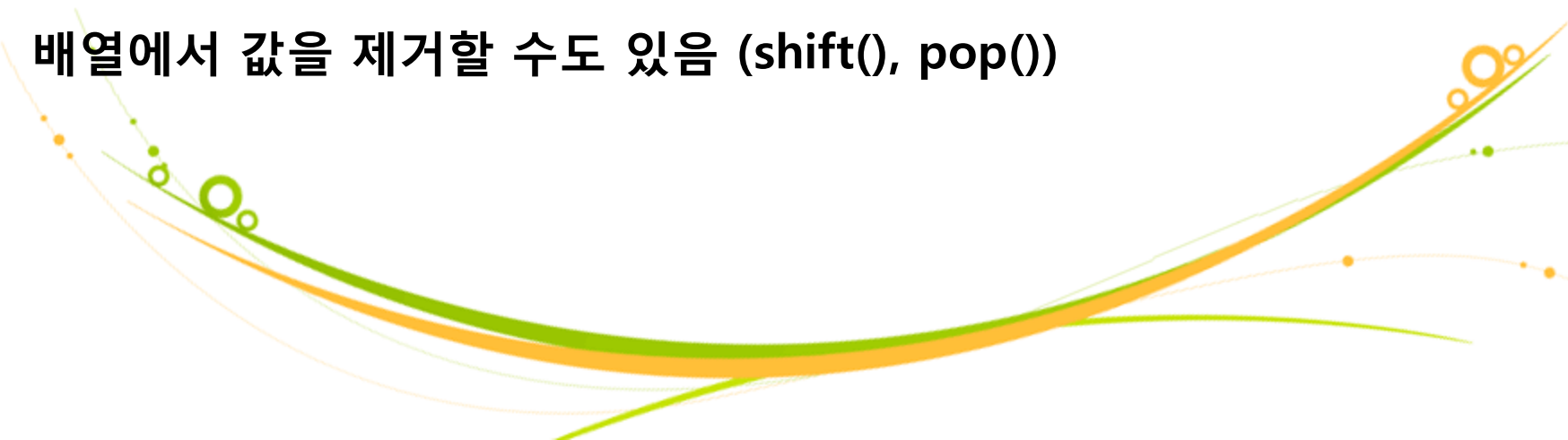
jQuery - 연산

- 연산자 우선순위

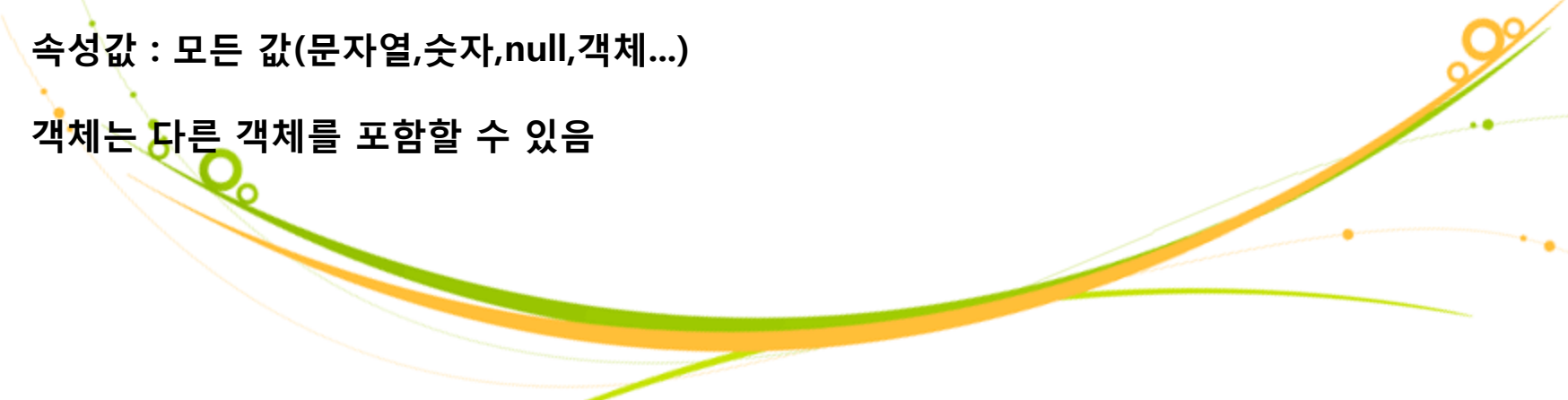
순위	연산자
1	(), [], .
2	! ++ --
3	* / %
4	+ -
5	> >= <= <
6	=== == !=
7	&&
8	
9	= += -= *= /=

jQuery - 배열

- `var arr = [3,4,5]`
- 배열은 값들이 들어가 있고 `index(숫자)`로 접근
- 사용법 `arr[0] => "3" 호출`
- 배열의 길이를 알고 싶으면 `배열명.length`로 알 수 있음
- 배열에는 값을 삽입할 수 있음 (직접, `push()`)
- 배열에서 값을 제거할 수도 있음 (`shift()`, `pop()`)



jQuery - 객체

- `var obj = {name:'장영석', age:'27', 이름 : 값}`
 - 객체는 '이름'과 '값'으로 구성 / 이름으로 값에 접근
 - `obj.name => 장영석` 호출
 - 숫자, 문자열, true/false, null, undefined 아니면 객체
 - 객체는 변경 가능한 속성들의 집합
 - 속성명 / 속성값을 가짐
 - 속성명 : 문자열
 - 속성값 : 모든 값(문자열,숫자,null,객체...)
 - 객체는 다른 객체를 포함할 수 있음
- 

jQuery 실행

