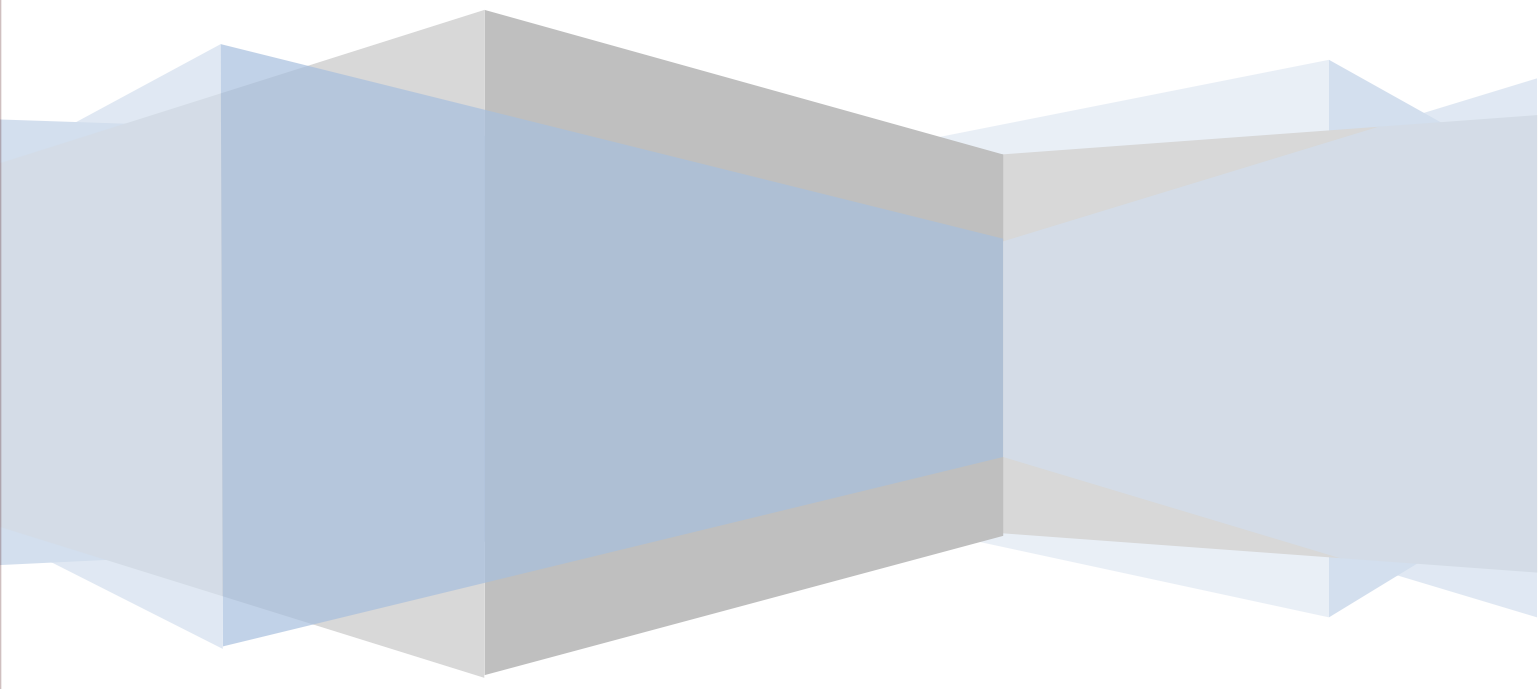
MEGAZONE CLOUD

JAVA 개발자 양성과정

다이나믹 웹 문서 제작

JavaScript & jQuery



제이쿼리(jQuery) 소개

· 자바스크립트를 쉽게 활용할 수 있도록 도와주는 오픈소스 기반의 자바스크립트 라이브러리이다.

· https://jquery.com/ 사이트 참고할 것.

● jQuery 메서드

· $( ) 형식으로 표기한다.

1. jQuery(selector [, context])

· DOM 에서 선택자에 해당하는 엘리먼트들을 검색하여 jQuery 객체로 반환한다.

2. jQuery(element)

· DOM 엘리먼트를 jQuery 객체로 반환한다.

3. jQuery(object)

· PlainObject 를 jQuery 객체로 반환한다.

※ PlainObject : { }

jQuery(html [, ownerDocument])

- HTML 문자열인 경우 새로운 DOM 엘리먼트를 생성한 후 jQuery 객체로 반환한다.

```
// <p id='test'>text</p> JQuerey 객체로 변환 후 body 태그의 자식 요소로 추가한다.
$("<p id='test'>text</p>").appendTo("body");
```

선택자 (Selectors)

1. 아이디(#) 선택자

```
$('#zero')
```

2. 클래스(.)

```
$('.zero')
```

3. 태그

```
$('p')
```

4. 속성([ ])

```
$('[value]')
```

```
$('[value=”zero”]')
```

5. 자식 선택자

```
$('#zero > p')
```

6. 후손 선택자

```
$('#zero div')
```

7. 형제자매 선택자

+ 는 다음 하나만 선택하고, ~ 는 형제자매중 일치하는 것을 모두 선택한다.

```
$ ( '#zero + div' )
```

```
$ ( '#zero ~  p' )
```

8. 가상 선택자

:가상선택자

1) 각각 모든 인풋, 버튼, 이미지, 체크박스, 라디오, 텍스트인풋 태그(엘리먼트)를 선택한다.

$ (':input') , $ (':button') , $ (':image') , $ (':checkbox') , $ (':radio') , $ (':text') ;

2) 각각 홀수 번째, 짝수 번째 태그, 주어진 순서보다 순서가 큰 태그, 순서가 작은 태그, 선택된 것들 중에 마지막 태그를 선택한다.

$ (':odd') , $ (':even') , $ (':gt(순서)') , $ (':lt(순서)') , $ (':last')

3) 각각 현재 포커스된 태그, 자식이 하나라도 있는 태그, 자식이 없는 태그, 비활성화된 태그, 활성화된 태그, 투명이 아닌, 숨겨진 태그를 선택한다.

$ (':focus') , $ (':parent') , $ (':empty') ; , $ (':disabled') , $ (':enabled') , $ (':visible') , $ (':hidden') ;

4) 각각 체크된(체크박스), 선택된(select), 부모의 유일한 자식인 태그, 첫 번째 자식인 태그, 마지막 자식인 태그를 선택한다.

$ (':checked') , $ (':selected') , $ (':only-child') ; , $ (':first-child') , $ (':last-child')

5) 각각 순서 번째 자식 태그, 해당 타입 중 순서 번째 자식 태그, 순서 번째 태그를 찾는다.

인덱스가 eq 는 0 부터 시작하고, nth-child 는 1 부터 시작한다.

$ (':nth-child(순서)') , $ (':nth-of-type(순서)') , $ (':eq(순서)')

6) 각각 선택자를 자식으로 갖고 있는 태그나 해당 텍스트를 갖고 있는 태그를 선택한다.

$ (':has(선택자)') , $ (':contains(텍스트)')

DOM 탐색 메소드와 필터

● DOM 탐색 Filter

:paraent	자식요소를 가진 DOM 객체 탐색
:empty	자식요소가 없는 DOM 객체 탐색
:first-child	첫번째 자식요소인 DOM 객체 탐색
:last-child	마지막 자식요소인 DOM 객체 탐색
:nth-child	n 번째 자식요소인 DOM 객체 탐색
:only-child	자신이 유일한 자식요소인 DOM 객체 탐색

● 트리구조 탐색

parent()	선택한 노드의 부모 노드를 선택한다
parents()	선택한 노드의 모든 조상 노드를 선택한다
children()	선택된 노드의 모든 자식노드만 선택한다.
prev()	선택한 노드의 이전 노드를 선택한다.
prevAll()	선택한 노드의 모든 이전 노드를 선택한다.
nextAll()	해당요소의 형제노드중 이후 요소 모두 탐색한다.
find()	선택한 노드의 자손 노드 중 조건에 맞는 노드를 선택한다.
next()	선택한 노드의 다음 노드를 선택한다.
nextAll()	선택한 노드의 모든 다음 노드를 선택한다.
closest()	선택한 노드를 포함하면서 가장 가까운 상위노드를 선택한다.
nextUntil()	다음에 위치한 노드를 조건에 맞을 때까지 찾는다.
parentsUntil()	조건이 참이 될 때까지 부모 노드를 찾는다.
prevUntil()	이전에 위치한 노드를 조건에 맞을때까지 찾는다.
siblings()	형제 노드를 모두 찾는다.

```
// li 태그의 부모 노드를 선택한 후 스타일을 적용한다.
$('li').parent().css('background-color', 'red');
```

● DOM 필터링

· 선택된 집합에서 다시 조건에 맞는 jQuery 집합을 선택한다.

· 필터링 메소드를 사용하면 메서드 체인 방식을 사용할 수 있다.

eq()	인덱스 번호에 해당하는 노드를 찾는다. 인덱스는 0 부터 시작
filter()	선택된 노드 집합에서 선택자를 추가하거나 함수를 사용하여 원하는 결과를 추출한다.
first()	선택된 노드 집합에서 첫 번째 자식 노드를 찾는다.
has()	선택된 노드들이 자신의 ~~자식(자손)~~요소에서 주어진 선택자가 있는지를 확인하여 범위를 축소한다.
is()	매개변수로 selector, element, jQuery 를 입력하여 입력한 객체가 있으면 true 를 반환한다.
last()	선택된 노드 집합에서 마지막 자식 노드를 찾는다.
siblings()	해당요소를 제외한 형제노드 모두 탐색
map()	대상이 되는 노드 집합을 변경한다.
not()	조건에 맞지 않는 것들만 찾아서 선택한다.
slice()	선택된 집합을 조건의 범위로 재선택한다.

※ find() 메서드를 이용한 chain

· chain 의 대상을 바꿔서 체인을 계속 연장시킬 수 있는 방법을 제공한다.

```
<html>

    <body>
        <ul class="first">
            <li class="foo"> list item 1 </li>
            <li> list item 2 </li>
            <li class="bar"> list item 3 </li>
        </ul>
        <ul class="second">
            <li class="foo"> list item 1 </li>
            <li> list item 2 </li>
            <li class="bar"> list item 3 </li>
        </ul>

        <script type="text/javascript">

        $('ul.first').find ('.foo').css('background-color', 'red').
        end().find('.bar').css('background-color', 'green');
        </script>

    </body>

</html>
```

● 다양한 탐색(Miscellaneous Traversing)

contents()

· 선택한 요소의 자식 요소를 가져온다.(text node 포함)

```
$('.container').contents().filter(function(){ });
// class container 요소의 자식요소(text 포함)를 가져와 filter 한다.
```

add()

· 요소를 추가 선택한다.

```
$('p').add('div')
// p 요소와 div 요소를 선택한다.
```

end()

· jQuery 함수를 연쇄적으로(chain)으로 사용할 경우 앞쪽에 이미 선택되었던 요소로 돌아간다.

```
// ul 의 첫번째 요소에서 class 속성값이 foo 인 엘리먼트를 찾아 배경색을 red 로 변경한다.
// find 를 호출하기 전의 요소($('ul:first))에서 class 속성값이 bar 인 엘리먼트를 찾아 배경색을 green 로 변경한다.
$('ul:first').find('.foo).css('background-color' , 'red')
 .end().find('bar').css('background-color', 'green');
```

andSelf()

· 선택된 구조 요소를 이어 붙인다.

```
// div 와 내부의 p 요소를 선택해서 결합 후 css 적용한다.
$('div').find('p').andSelf().css(..);
```

● DOM Manipulation

· DOM 구조를 변경할 수 있는 jQuery 메소드를 제공한다.

1. DOM insertion inside

append()	선택된 내부의 맨 뒤에 자식 노드를 새로 추가한다.
appendTo(target)	새로운 노드를 target 에 해당하는 노드 내부의 마지막에 추가한다.
prepend()	선택된 노드 내부의 맨 앞에 자식 노드를 새로 추가한다.
prependTo(target)	새로운 노드를 target 에 해당하는 노드 내부의 첫 번째로 추가한다
html()	노드 내부의 HTML 을 읽고 쓸 수 있다.
text()	노드의 텍스트를 읽고 쓸 수 있다.

2. DOM insertion outside

after()	선택된 노드 뒤에 새로운 노드를 추가한다.(형제 관계)
before()	선택된 노드 앞에 새로운 노드를 추가한다.(형제 관계)
insertAfter()	작성된 jQuery 객체를 target 뒤에 삽입한다. (형제 관계)
insertBefore()	작성된 jQuery 객체를 target 앞에 삽입한다. (형제 관계)

```
//  <b>Hello</b> jQuery 객체로 변환한 후 p 태그 뒤에 삽입한다. (형제관계)
$("p"). after ("<b>Hello</b>");
```

3. DOM Insertion, Around

wrap()	선택된 집합을 각각의 매개 변수로 넘긴 HTML 구조로 감싼다.
wrapAll()	선택된 집합의 전체 외곽을 매개 변수로 넘긴 HTML 구조로 감싼다.
wrapInner()	선택된 집합의 내부에 매개변수로 넘긴 HTML 구조로 감싼다.

4. DOM Removal

detach()	DOM 에서 조건에 일치되는 노드들을 제거한다. (단, 메모리에 남아 있기 때문에 다시 사용할 수 있다.)
empty()	DOM 에서 조건과 일치하는 노드들의 자식 노드들을 제거한다
remove()	DOM 에서 조건과 일치하는 노드들을 제거한다.
unwrap()	wrap()의 반대로, 선택된 집합을 감싸고 있는 HTML 을 제거한다

```
// id가 content인 div의 모든 내용을 DOM 구조에서 제거하고, 제거된 내용은 temp에 저장한다.

var temp = $("div#content").detach();
```

5. DOM Replacement

replaceAll()	조건에 맞는 노드들을 타겟 노드로 대체한다.
replaceWith()	조건에 맞는 노드들을 매개 변수로 넘긴 새로운 HTML 로 대체한다.

clone : 선택한 노드와 똑같은 노드를 복사한다.

● Attribute/CSS

· 선택한 노드의 속성 값을 가져오거나 텍스트를 변경할 수 있고, CSS() 함수를 사용하면 CSS에 의해 적용되는 상태를 동적인 상황에서도 변경할 수 있다.

1. jQuery CSS 관련 메서드

addClass()	특정한 클래스를 노드에 추가할 수 있다.
css()	css() 함수는 element 의 속성 값을 알아낼수 도 있고, 설정할 수도 있다.
hasClass()	특정한 클래스가 있는지를 찾을 수 있다.
removeClass()	특정한 클래스를 요소에서 제거할 수 있다.
toggleClass()	특정한 클래스의 추가, 제거를 한 번에 처리할 수 있다.

css(propertyName)

```
// div 요소 배경색의 컬러값을 반환한다.

$("div").css('background-color');
```

css(propertyName, value)

```
// div 태그의 내용을 빨강색으로 설정한다.

$("div").css('color', '#f00');

$("div").css({'background-color': '#ddd',
              'color' : '#f00',
              'font-size' : '10px'});
```

addClass(className)

```
// p 요소에 myclass 스타일 추가
$('p').addClass('myclass');
<p class=”myclass”></p>
```

removeClass(className)

· removeClass는 기존 태그에 적용되어 있는 클래스를 제거하는 함수이다.

```
// p 요소에서 하나 이상의 클래스를 젝거하려면 공백을 사용한다.
$('p').removeClass('classA classB classC');
```

jQuery Event

JQuery 이벤트 처리 방식

● 객체에 직접 이벤트를 등록

```
$('div').click(function() {
	// 이벤트 처리
});
```

● on 메서드를 이용하여 등록

```
$('#btn').on('click', function() {
	// 이벤트 처리
});


$('#btn').on('mouseenter mouseleave', function() {
	// 이벤트 처리
	$(this).toggleClass('entered');
});
```

```
$('#foo').bind ({

    click: function() {

         // Do someting on click

    },
    mouseenter: function() {

         // Do someting on mouseenter
    }

});
```

● 이벤트 제거

```
// id 가 obj 객체에서 click 이벤트를 제거한다.
$("#obj").unbind("click");

//id 가 obj 객체에 바인딩된 모든 이벤트를 제거한다.
$("#obj").unbind();
```

● hover 이벤트

· hover 이벤트는 mouseenter 와 mouseleave 이벤트를 하나로 묶어 처리해주는 이벤트이다.

· hover 는 2 개의 함수를 가지고 있는데, 첫 번째 함수는 mouseenter 이벤트 때에 호출되는 함수이고, 두 번째 함수 mouseleave 이벤트 때에 호출되는 함수이다

```
$("selector").hover(function() {
                        // Do something on mouseenter
                    },
                    function() {
                        // Do something on mouseleave
                    });
```

```
<script>
	$("div").click(function(e) {
		$(this).css("background", "yellow");
	});

	$("a").click(function(e) {
		$(this).css("background", "yellow");
		// 이벤트 흐름을 정지하여 더이상 부모에게 이벤트 흐름이 전달되지 않는다. bubbling 차단
		e.stopPropagation();
	});
 </script>
```

● Event Handler

· 이벤트 핸들러는 일반적으로 콜백함수를 사용하여 이벤트를 처리한다.

· 이벤트 핸들러내에서 this 키워드는 이벤트 핸들러에 바인딩된 DOM 엘리먼트를 참조한다.

jQuery 에서 엘리먼트를 사용하려면 $() 함수를 사용하여 jQuery 객체로 변환해야 한다.

```
$("#foo").bind( 'click', function() {
    alert( $(this).text() );
});
```

● 이벤트 객체(event object)

· 이벤트 핸들러 함수가 jQuery에서 콜백될 때 이벤트 객체(event object)가 함수의 인자로 전달된다.

· 전달받은 이벤트 객체를 이용하여 이벤트의 특성을 결정하거나, 이벤트의 기본 동작을 막을 수도 있다.

type	이벤트 종류
pageX	브라우저 화면을 기준으로 한 마우스 X 좌표 위치
pageX	브라우저 화면을 기준으로 한 마우스 X 좌표 위치
preventDefault()	기본 이벤트를 제거한다.
stopPropagation()	이벤트 전파를 제거한다.