

# jQuery 1.3 Cheatsheet

<http://www.jquery.com> <http://docs.jquery.com>

Download at: <http://www.gmtaz.com>

## SELECTORS

<b>Basics</b>	<b>Content Filters</b>	<b>Forms</b>
#id	:contains(text)	:input
element	:empty	:text
.class	:has(selector)	:password
*	:parent	:radio
selector1, selectorN, ...		:checkbox
	<b>Visibility Filters</b>	:submit
<b>Hierarchy</b>	:hidden	:image
ancestor descendant	:visible	:reset
parent > child		:button
prev + next	<b>Attribute Filters</b>	:file
prev ~ siblings	[attribute]	:hidden
	[attribute=value]	
<b>Basic Filters</b>	[attribute != value]	<b>Form Filters</b>
:first	[attribute ^=value]	:enabled
:last	[attribute \$=value]	:disabled
:not(selector)	[attribute *=value]	:checked
:even	[attributeFilter1][attributeFilterN]	:selected
:odd		
:eq(index)	<b>Child Filters</b>	
:gt(index)	:nth-child(index/even/odd/equation)	
:lt(index)	:first-child	
:header	:last-child	
:animated	:only-child	

## ATTRIBUTES / CSS

<b>Attributes</b>	<b>Value</b>
attr(name)	val()
attr(properties)	val(val)
attr(key, value)	val(val)
attr(key, fn)	
removeAttr(name)	
	<b>Positioning</b>
<b>Class</b>	offset()
addClass(class)	position()
hasClass(class)	scrollTop(val)
removeClass(class)	scrollLeft(val)
toggleClass(class)	scrollLeft(val)
toggleClass(class, switch)	
	<b>CSS</b>
<b>Height and Width</b>	css(name)
height()	css(properties)
height(val)	css(name, value)
width()	
width(val)	

## EVENTS

<b>Event Helpers</b>			
blur()	error()	keyup()	mouseover(fn)
blur(fn)	error(fn)	keyup(fn)	mouseup(fn)
change()	focus()	load(fn)	resize(fn)
change(fn)	focus(fn)	mousedown(fn)	scroll(fn)
click()	keydown()	mouseenter(fn)	select()
click(fn)	keydown(fn)	mouseleave(fn)	select(fn)
dblclick()	keypress()	mousemove(fn)	submit()
dblclick(fn)	keypress(fn)	mouseout(fn)	submit(fn)
			upload(fn)
<b>Event Handling</b>	<b>Live Events</b>	<b>Page Load</b>	
bind(type, [data], fn)	live(type, fn)	ready(fn)	
one(type, [data], fn)	die([type], [fn])		
trigger(event, [data])	<b>Interaction Helpers</b>		
triggerHandler(event, [data])	hover(over, out)		
unbind([type], [fn])	toggle(fn, fn2, [fn3], [fn4],...)		

## EFFECTS

<b>Basics</b>	<b>Settings</b>
show()	\$.fx.off
show(speed, [callback])	
hide()	
hide(speed, [callback])	
toggle()	
toggle(speed, [callback])	
toggle(switch)	
toggle(speed, [callback])	
<b>Sliding</b>	
slideDown(speed, [callback])	
slideUp(speed, [callback])	
slideToggle(speed, [callback])	
<b>Fading</b>	
fadeIn(speed, [callback])	
fadeOut(speed, [callback])	
fadeTo(speed, opacity, [callback])	
<b>Custom</b>	
animate(params, [duration], [easing], [callback])	
animate(params, options)	
stop([clearQueue], [gotoEnd])	

## MANIPULATION

<b>Changing Contents</b>	<b>Replacing</b>
html()	replaceWith(content)
html(val)	replaceAll(selector)
text()	
text(val)	<b>Removing</b>
	empty()
<b>Inserting Inside</b>	remove([expr])
append(content)	
appendTo(selector)	<b>Copying</b>
prepend(content)	clone()
prependTo(selector)	clone(bool)
<b>Inserting Outside</b>	<b>Inserting Around</b>
after(content)	wrap(html)
before(content)	wrap(elem)
insertAfter(selector)	wrapAll(html)
insertBefore(selector)	wrapAll(elem)
	wrapInner(html)
	wrapInner(elem)

## TRAVERSING

<b>Filtering</b>	<b>Finding</b>
eq(index)	add(expr)
filter(expr)	children([expr])
filter(fn)	closest([expr])
is(expr)	contents()
map(callback)	find(expr)
not(expr)	next([expr])
slice(start, [end])	nextAll([expr])
	offsetParent()
<b>Chaining</b>	parent([expr])
andSelf()	parents([expr])
end()	prev([expr])
	prevAll([expr])
	siblings([expr])

## AJAX

<b>Ajax Request</b>
\$.ajax(options)
load(url, [data], [callback])
\$.get(url, [data], [callback], [type])
\$.getJSON(url, [data], [callback])
\$.getScript(url, [callback])
\$.post(url, [data], [callback], [type])
<b>Ajax Events</b>
ajaxComplete(callback)
ajaxError(callback)
ajaxSend(callback)
ajaxStart(callback)
ajaxStop(callback)
ajaxSuccess(callback)
<b>Misc</b>
\$.ajaxSetup(options)
serialize()
serializeArray()

## CORE

<b>The jQuery Function</b>	<b>Data</b>
\$(expression, [context])	data(name)
\$(html, [ownerDocument])	data(name, value)
\$(elements)	removeData(name)
\$(callback)	queue([name])
<b>jQuery Object Accessors</b>	queue([name], queue)
each(callback)	dequeue([name])
size()	
length	<b>Plugins</b>
selector	\$.fn.extend(object)
context	\$.extend(object)
eq(position)	<b>Interoperability</b>
get()	\$.noConflict()
get(index)	\$.noConflict(extreme)
index(subject)	

## UTILITIES

<b>Browser and Feature D</b>	<b>Test operations</b>
\$.support	\$.isArray(obj)
\$.browser	\$.isFunction(obj)
\$.browser.version	<b>String operations</b>
\$.boxModel	\$.trim(str)
<b>Array and Object operations</b>	<b>URLs</b>
\$.each(object, callback)	\$.param(obj)
\$.grep(array, callback, [invert])	
\$.makeArray(obj)	
\$.map(array, callback)	
\$.inArray(value, array)	
\$.merge(first, second)	
\$.unique(array)	
\$.extend([deep], target, object1, [objectN])	

# 13. 문서 객체 탐색

Wallpaper created by Gustavo Tandeciarz and adapted from <http://oscarotero.com/jquery/>

# 13.1 기본 필터링 메서드

## ❖ 기본 필터링 메서드

- jQuery의 선택자를 사용하면 원하는 문서 객체 대부분 선택 가능
- 기본 지원하지 않는 필터를 사용해 문서 객체를 선택해야 한다면?
- **filter(expr)**
  - 대상이 되는 element들 중 입력 값에 일치하는 element를 리턴한다.
- **filter(fn)**
  - 대상이 되는 element중 function에 구현된 내용이 true를 리턴할 경우 대상 element 집합에 포함시키고, false일 경우 대상 element 집합에서 제외시킨다

표 13-1 jQuery 메서드

메서드 이름	설명
filter()	문서 객체를 필터링합니다.

filter() 메서드는 다음과 같은 두 가지 형태로 사용합니다.

- 1 \$(selector).filter(selector);
- 2 \$(selector).filter(function () { });



## 13.1 기본 필터링 메서드

### ❖ filter() 메서드 사용법 예제

- h3 태그에서 홀수 번째에 위치하는 문서 객체 선택
  - 코드 14-2는 홀수 번째에 위치하는 h3 태그의 스타일 속성 변경 (선택자)

코드      body 태그 구성

```
<body>
  <h3>Header-0</h3>
  <h3>Header-1</h3>
  <h3>Header-2</h3>
  <h3>Header-3</h3>
  <h3>Header-4</h3>
  <h3>Header-5</h3>
</body>
```

코드      필터 선택자

```
<script>
  $(document).ready(function () {
    $('h3:even').css({
      backgroundColor: 'Black',
      color: 'White'
    });
  });
</script>
```



## 13.1 기본 필터링 메서드

### ❖ filter() 메서드 사용법 예제

- 1번 형태의 filter() 메서드 사용
- 예제 코드
  - filter() 메서드의 매개 변수에 선택자 입력
  - ':even'을 입력할 수도 있음

코드 filter() 메서드

```
<script>
$(document).ready(function () {
    $('h3').filter(':even').css({
        backgroundColor: 'Black',
        color: 'White'
    });
});
</script>
```



## 13.1 기본 필터링 메서드

### ❖ filter() 메서드 사용법 예제

- 2번 형태의 filter() 메서드를 사용
  - filter() 메서드의 매개 변수에 함수를 넣음
  - 입력하는 함수는 매개 변수로 index를 가짐
  - 함수에서 리턴하는 값에 따라 문서 객체 선택

코드 filter() 메서드의 매개 변수로 함수를 넣을 경우

```
<script>
    $(document).ready(function () {
        $('h3').filter(function (index) {
            return index % 3 == 0;
        }).css({
            backgroundColor: 'Black',
            color: 'White'
        });
    });
</script>
```

그림 filter() 메서드의 사용

**Header-0**

**Header-1**

**Header-2**

**Header-3**

**Header-4**

**Header-5**



## 13.2 문서 객체 탐색 종료

### ❖ 예제

- 체이닝을 사용해 한 줄로 서로 다른 문서 객체에 스타일 적용 가능

코드      HTML 페이지 구성

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-1.7.js"X/script>
  <script>
    $(document).ready(function () {
      $('h1').css('background', 'Orange').filter(':even').css('color', 'Red');
    });
  </script>
</head>
<body>
  <h1>Header-0</h1>
  <h1>Header-1</h1>
  <h1>Header-2</h1>
</body>
</html>
```



## 13.2 문서 객체 탐색 종료

### ❖ 체이닝하려면?

- 체이닝할 때 추가한 filter() 메서드 제거하려면 end() 메서드 사용

코드 14-6 연습

```
<script>
$(document).ready(function () {
    $('h1').css('background', 'Orange');
    $('h1:even').css('color', 'White');
    $('h1:odd').css('color', 'Red');
});
</script>
```

표 14-6 jQuery 메서드

메서드 이름	설명
end()	문서 객체 선택을 한 단계 뒤로 돌립니다.

end() 메서드를 사용하면 코드 14-6을 코드 14-7의 형태로 사용할 수 있습니다.

코드 14-7 end() 메서드를 사용한 체이닝

```
$('h1').css('background', 'Orange').filter(':even').css('color',
'White').end().filter(':odd').css('color', 'Red');
```





## 13.3 특정 위치의 문서 객체 선택

### ❖ 필터 선택자 이용

- 특정 위치에 존재하는 문서 객체 선택
- 필터 선택자는 자주 사용하므로 메서드로 제공
- **eq(index)**
  - 여러 개의 일치하는 element들 중 입력 값으로 받은 값의 위치에 해당하는 element를 리턴한다. 입력 값의 시작은 '0'부터 시작하며, 입력 값에 해당하는 위치에 element가 존재하지 않을 경우 null을 리턴한다.

표 | jQuery 메서드

메서드 이름	설명
eq()	특정 위치에 존재하는 문서 객체를 선택합니다.
first()	첫 번째에 위치하는 문서 객체를 선택합니다.
last()	마지막에 위치하는 문서 객체를 선택합니다.

코드 | body 태그 구성

```
<body>
  <div>
    <h1>Header-0</h1>
    <h1>Header-1</h1>
    <h1>Header-2</h1>
  </div>
</body>
```





## 13.3 특정 위치의 문서 객체 선택

### ❖ eq() 메서드

- 매개 변수에 숫자 입력
  - 양수를 입력할 수도 있지만, 음수를 입력할 수도 있음
  - 음수를 입력하면 뒤쪽 기준으로 선택
  - 맨 뒤의 문서 객체가 -1번째

코드 eq() 메서드

```
<script>
    $(document).ready(function () {
        $('h1').eq(0).css('background', 'Orange');
        $('h1').eq(-1).css('background', 'Red');
    });
</script>
```

그림 14-2 eq() 메서드의 사용

**Header-0**

**Header-1**

**Header-2**



## 13.4 문서 객체 추가 선택

### ❖ add() 메서드

- jQuery를 통해 선택된 element set에 add(expr)에 일치하는 element들을 추가
- (단, add()에 의해 추가되는 element가 document에 추가되는 것은 아니며, 검색된 대상 집합에만 추가된다.)

표 jQuery 메서드

메서드 이름	설명
add()	문서 객체를 추가로 선택합니다.

코드 처럼 body 태그를 구성해주세요.

코드	body 태그 구성
	<pre>&lt;body&gt;   &lt;h1&gt;Header-0&lt;/h1&gt;   &lt;h2&gt;Header-1&lt;/h2&gt;   &lt;h1&gt;Header-2&lt;/h1&gt;   &lt;h2&gt;Header-3&lt;/h2&gt;   &lt;h1&gt;Header-4&lt;/h1&gt; &lt;/body&gt;</pre>



## 13.4 문서 객체 추가 선택

### ❖ 예제

- add() 메서드 활용 예제
- h1 태그의 background 스타일 속성에 Gray 입력
- h2 태그를 추가로 선택, float 스타일 속성에 left 입력

코드 add() 메서드

```
<script>
$(document).ready(function () {
    $('h1').css('background', 'Gray').add('h2').css('float','left');
});
</script>
```

그림 add() 메서드의 사용

**Header-0**Header-1**Header-2**Header-3Header-4



## 13.5 문서 객체의 특징 판별

### ❖ 문서 객체가 특징을 가지고 있는지 판단

#### ■ is(expr)

- 대상이 되는 element 중에서 입력 값인 expr에 만족하는 element가 하나라도 존재할 경우 true를 리턴하고, 그렇지 않을 경우 false를 리턴한다.
- is() 메서드는 매개 변수로 선택자 입력

표 jQuery 메서드

메서드 이름

설명

is()

문서 객체의 특징을 판별합니다.

코드 body 태그 구성

```
<body>
  <h1 class="select">Header-0</h1>
  <h1>Header-1</h1>
  <h1 class="select">Header-2</h1>
</body>
```



## 13.5 문서 객체의 특징 판별

### ❖ 예제

- h1 태그 중 select 클래스 속성이 있는 문서 객체에 스타일 적용

코드 1 is() 메서드

```
<script>
$(document).ready(function () {
    $('h1').each(function () {
        if ($(this).is('.select')) {
            $(this).css('background', 'Orange');
        }
    });
});
</script>
```

그림 is() 메서드의 사용

**Header-0**

**Header-1**

**Header-2**



## 13.6 특정 태그 선택

### ❖ 문서 객체에서 특정 태그를 선택하는 방법

#### ■ find(expr)

- 입력 값으로 받은 표현식에 해당하는 모든 DOM element를 검색한다.
- 입력 값으로 받는 표현식은 CSS selector 문법과 기본적인 XPath의 검색 표현식을 따른다.

표 jQuery 메서드

메서드 이름	설명
find()	특정 태그를 선택합니다.

코드 body 태그 구성

<body>

</body>



## 13.6 특정 태그 선택

### ❖ XML 문서에서 데이터를 추출하는 예제

- find() 메서드 사용
- parseXML() 메서드를 사용해 문자열을 XML 문서 객체로 변경
- 인터넷 익스플로러에서는 문제가 발생
  - parseXML() 메서드 사용





## 13.6 특정 태그 선택

### ❖ 문자열에 XML 문서 입력

코드 ; XML 문서

```
<script>
  // 변수를 선언합니다.
  var xml = '';
  xml += '<friends>';
  xml += '  <friend>';
  xml += '    <name>연하진</name>';
  xml += '    <language>Ruby</language>';
  xml += '  </friend>';
  xml += '  <friend>';
  xml += '    <name>윤명월</name>';
  xml += '    <language>Basic</language>';
  xml += '  </friend>';
  xml += '  <friend>';
  xml += '    <name>윤하린</name>';
  xml += '    <language>C#</language>';
  xml += '  </friend>';
  xml += '</friends>';

  $(document).ready(function () {
    // 변수를 선언합니다.
    var xmlDoc = $.parseXML(xml);
  });
</script>
```



## 13.6 특정 태그 선택

### ❖ 예제

- `$()` 메서드의 매개 변수에 생성한 XML 문서 객체 입력
- `$()` 메서드에는 문서 객체를 곧바로 넣을 수 있음
- `find()` 메서드를 사용해 friend 태그 선택
  - friend 태그는 총 세 개이므로 `each()` 메서드 사용

코드      `$.parseXML()` 메서드와 `each()` 메서드

```
$(document).ready(function () {  
    // 변수를 선언합니다.  
    var xmlDoc = $.parseXML(xml);  
    $(xmlDoc).find('friend').each(function (index) {  
  
    });  
});
```



## 13.6 특정 태그 선택

### ❖ 예제

- each() 메서드 안에서 find() 메서드를 한 번 더 사용해 데이터 추출
- text() 메서드를 사용하면 태그 내부 글자를 가져올 수 있음
- 각각의 데이터로 HTML 태그를 만든 후 innerHTML 속성에 추가

코드 XML 파싱

```
$(document).ready(function () {  
    // 변수를 선언합니다.  
    var xmlDoc = $.parseXML(xml);  
    $(xmlDoc).find('friend').each(function (index) {  
        // 변수를 선언합니다.  
        var output = '';  
        output += '<div>';  
        output += '    <h1>' + $(this).find('name').text() + '</h1>';  
        output += '    <p>' + $(this).find('language').text() + '</p>';  
        output += '</div>';  
  
        // 출력합니다.  
        document.body.innerHTML += output;  
    });  
});
```

그림 XML 파싱을 사용한 문서 객체 생성

연하진	윤명월	윤하린
Ruby	Basic	C#



## 13.7 기타

### ❖ **hasClass(class)**

- 입력 값에 해당하는 class가 존재하는 경우 true를 리턴하고 일치하지 않을 경우 false를 리턴한다

### ❖ **map(fn)**

- 대상이 되는 element집합을 변경한다.

### ❖ **slice(start[, end])**

- 대상이 되는 element중 시작 번째부터 끝 번째까지의 element를 리턴한다.

### ❖ **children([expr])**

- 대상이 되는 모든 element에서 입력 값으로 받은 expr에 적합한 바로 하위의(인접한) child-element를 리턴한다.

### ❖ **contents()**

- 대상이 되는 element 내부의 탐색 및 변경을 가능하게 한다.
- 현재 document에 있는 element의 작업뿐 아니라, iframe을 통해 포함된 document의 탐색 및 변경도 가능하다.

### ❖ **next([expr])**

- 대상이 되는 element를 기준으로 동일한 노드의 현재 위치 다음의 인접한 element를 찾는다.
- 입력 값(expr)이 생략된 경우 대상 element 바로 다음의 element를 리턴하며, 입력 값(expr)이 있을 경우 next(...)의 대상이 되는 element가 expr 표현식과 일치하는 경우만 리턴한다.



## 13.7 기타

### ❖ **nextAll([expr])**

- 대상이 되는 element를 기준으로 현재 노드와 같거나 하위의(현재 위치 이후의)element를 찾는다.
- 입력 값(expr)이 생략된 경우 대상 element 바로 뒤의 element들을 리턴한다.

### ❖ **parent([expr])**

- 입력 값에 해당하는 인접한 부모 노드를 찾는다. 부모 노드가 둘 이상일 경우 parent()는 jQuery 배열 형식으로 리턴한다.

### ❖ **parents([expr])**

- 입력 값(expr)에 해당하는 부모 노드 모두를 찾는다.

### ❖ **prev([expr])**

- 대상이 되는 element를 기준으로 동일한 노드의 현재 위치 이전의 인접한 element를 찾는다. 입력값이 존재하지 않을 경우 대상 element 바로 앞의 element를 리턴한다.

### ❖ **prevAll([expr])**

- 대상이 되는 element를 기준으로 현재 노드와 같고 현재 위치 이전의 모든 element를 찾는다. 입력값이 존재하지 않을 경우 대상 element와 같은 노드에 존재하고 이전 위치에 존재하는 모든 element를 리턴한다.

### ❖ **not(expr)**

- 대상이 되는 element집합에서 not(expr)에 일치하는 element를 제외시킨다.



## 13.7 기타

### ❖ **siblings([expr])**

- 대상이 되는 노드를 기준으로 같은 노드의 위치에 있는 모든 노드를 찾는다.
- 입력 값이 존재할 경우 입력 값에 해당하는 element를 찾는다.

### ❖ **andSelf()**

- 결과 집합 element에 최초 대상이 되었던 element를 포함시켜서 리턴한다.

### ❖ **end()**

- 결과 집합 element에 최초 대상이 되었던 element를 포함시켜서 리턴한다.
- jQuery의 장점 중 하나가 chaining을 제공하는 것이다.
- chaining이란 원하는 function을 특정 element에 적용한 후 적용된 결과를 jQuery 객체 형태로 다시 리턴한다.
- 따라서 리턴받은 jQuery 객체에 또 다른 function을 적용하는 작업을 연이어 할 수 있다.
- 하지만, 때에 따라서는 이전 상태의 jQuery 객체가 필요할 경우도 발생한다.
- 이런 경우 end()를 사용하여, jQuery 객체의 이전 상태의 참조가 가능하다.

