

- jQuery는 웹사이트에서 자바스크립트를 보다 쉽게 쓸 수 있도록 한다.
- jQuery는 “적게 쓰고, 많은 일을 하는” 자바스크립트 라이브러리이다.
- jQuery는 HTML, CSS, HTML 이벤트, 효과와 애니메이션 등을 조작할 수 있다.
- jQuery는 현재 대부분의 최신 브라우저에서 완벽하게 실행된다.

유형	설명	“ http://jquery.com/download/ ” 참조
사용버전	실제 웹사이트에서 사용하기 위한 축소, 압축된 버전	『jquery-3.1.1.min.js』
개발버전	테스트 혹은 개발을 위해 마련된 축소, 압축되지 않은 버전	『jquery-3.1.1.js』

유형	용법	설명
다운로드 방식	<pre><head><script src="jquery-3.1.1.min.js"></script></head></pre> <pre><head><script src="jquery-3.1.1.js"></script></head></pre>	http://jquery.com/download 에서 해당 버전을 다운로드하여 사용할 HTML 페이지와 동일한 디렉토리에 위치시킨다. <head>태그 내에 <script>태그로 해당 파일을 연결한다.
CDN 연결 방식 (Content Delivery Network)	<pre><head><script src="http://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery-3.1.1.min.js"></script></head></pre> <pre><head><script src="http://ajax.aspnetcdn.com/ajax/jquery/3.1.1/jquery-3.1.1.min.js"></script></head></pre> <p>많은 사용자들이 인터넷의 웹사이트들을 이용하면서 기존에 다운로드 되어 있던 jQuery 파일이 사용자 컴퓨터 내부에 남아있을 수 있다. 따라서 다시 파일을 다운로드할 필요가 없으므로 빠르게 웹페이지를 로딩할 수 있게 된다.</p>	

jQuery 문법	
\$(선택자).메소드(); \$(선택자1).이벤트(function){ \$(선택자2).메소드(); });	/* (선택자)를 (메소드)하라 */ /* (선택자1)이 (이벤트)할 때 (선택자2)를 (메소드)하라 */
\$	jQuery를 정의/접근하는 기호
선택자	쿼리하려는(찾을) HTML 요소
이벤트	액션을 적용(수행)할 시기
메소드(액션)	선택된 요소에 적용(수행)시킬 행동
예시	
\$(this).hide()	현재 요소를 숨긴다.
\$("p").hide()	모든 p 태그들을 숨긴다.
\$("#apple").hide()	아이디가 apple인 요소를 숨긴다.
\$(".banana").hide()	클래스가 banana인 모든 요소를 숨긴다.
준비 이벤트	
아래와 같이 jQuery를 작성하면 페이지 로딩이 끝나기 전에 jQuery가 실행되는 것을 막을 수 있다.	
일반적인 표현	간소화 한 표현
<pre>\$(document).ready(function() { // jQuery 내용 });</pre>	<pre>\$(function() { // jQuery 내용 });</pre>

jQuery 선택자		
유형	용법	설명
*	\$("* ")	페이지 내 모든 요소를 선택한다.
this	\$(this)	현재 태그를 선택한다.
#id	\$("#apple")	아이디가 apple인 요소를 선택한다.
.class	\$(".banana")	클래스가 banana인 모든 요소들을 선택한다.
.class, .class	\$(".banana, .mango")	클래스가 banana, mango인 모든 요소들을 선택한다.
태그이름	\$("p")	모든 p태그들을 선택한다.
태그,태그,태그...	\$("p,h1,div")	모든 p, h1, div태그들을 선택한다.
:first	\$("p:first")	모든 p태그 중 가장 첫 번째에 등장하는 태그를 선택한다.
:last	\$("p:last")	모든 p태그 중 가장 마지막에 등장하는 태그를 선택한다.
:first-child	\$("p:first-child")	부모의 안에서 첫 번째로 등장하는 태그 중 p태그를 선택한다.
:last-child	\$("p:last-child")	부모의 안에서 마지막에 등장하는 태그 중 p태그를 선택한다.
:first-of-type	\$("p:first-of-type")	연속되는 p태그 중 첫 번째로 등장하는 태그를 선택한다.
:last-of-type	\$("p:last-of-type")	연속되는 p태그 중 마지막에 등장하는 태그를 선택한다.
:even	\$("tr:even")	tr태그 중 짝수 번째에 등장하는 태그를 선택한다. (0,2,4,6,...)
:odd	\$("tr:odd")	tr태그 중 홀수 번째에 등장하는 태그를 선택한다. (1,3,5,7,...)
:nth-child(n)	\$("p:nth-child(3)")	부모의 안에서 세 번째로 등장하는 태그 중 p태그를 선택한다.
:nth-last-child(n)	\$("p:nth-last-child(3)")	부모의 안에서 마지막에서 세 번째로 등장하는 태그 중 p태그를 선택한다.
:nth-of-type(n)	\$("p:nth-of-type(3)")	연속되는 p태그 중 세 번째로 등장하는 태그를 선택한다.
:nth-last-of-type(n)	\$("p:nth-last-of-type(3)")	연속되는 p태그 중 마지막에서 세 번째로 등장하는 태그를 선택한다.
:부모 > 아들	\$("div > p")	div의 직계 아들인 p태그를 선택한다.
:부모 후손	\$("div p")	div의 후손인 p태그를 선택한다.
:요소 다음요소	\$("div + p")	div의 바로 다음에 등장하는 p태그를 선택한다.
:요소 ~ 형제	\$("div ~ p")	div의 형제에 해당하는 p태그를 선택한다.
:eq(색인번호)	\$("ul li:eq(3)")	요소들 중 네 번째 아이템을 선택한다. (0부터 시작)
:gt(색인번호)	\$("ul li:gt(3)")	요소들 중 네 번째 아이템 이후의 아이템들을 선택한다.
:lt(색인번호)	\$("ul li:lt(3)")	요소들 중 네 번째 아이템 이전의 아이템들을 선택한다.
:not(선택자)	\$("input:not(empty)")	요소들 중 비어있지 않은 아이템을 선택한다.
:header	\$(".header")	모든 헤더 요소들을 선택한다. (<h1>, <h2>, <h3>...)
:animated	\$(".animated")	현재 애니메이션으로 움직이고 있는 모든 요소들을 선택한다.
:focus	\$(".focus")	현재 포커스가 위치하고 있는 요소를 선택한다.
:contains(텍스트)	\$(".contains('apple')")	태그 내용 중 "apple"이라는 단어를 포함하고 있는 모든 요소들을 선택한다.
:has(선택자)	\$("div:has(p)")	div태그 중 p태그를 포함하고 있는 모든 태그들을 선택한다.
:empty	\$(".empty")	비어있는 모든 요소들을 선택한다.
:hidden	\$("p:hidden")	p태그 중 hidden 스타일로 감춰진 태그들을 선택한다.
:visible	\$("p:visible")	p태그 중 현재 모습이 보이는 태그들을 선택한다.
[속성]	\$("[href]")	href 속성을 가진 모든 요소들을 선택한다.
[속성=값]	\$("[href='main.html']")	href 속성의 값이 "main.html"인 모든 요소들을 선택한다.
[속성!=값]	\$("[href!='main.html']")	href 속성의 값이 "main.html"이 아닌 모든 요소들을 선택한다.
[속성\$=값]	\$("[href\$='.jpg']")	href 속성의 값이 ".jpg"로 끝나는 모든 요소들을 선택한다.
[속성^=값]	\$("[href^='menu']")	href 속성의 값이 "menu"로 시작하는 모든 요소들을 선택한다.
[속성~=값]	\$("[href\$='menu']")	href 속성의 값에 "menu"라는 특정 단어가 포함되는 모든 요소들을 선택한다.
[속성*=값]	\$("[href*='menu']")	href 속성의 값에 "menu"가 포함되는 모든 요소들을 선택한다.
:input	\$(".input")	모든 input 태그들을 선택한다.
:text	\$(".text")	input 태그 중 type="text"가 포함된 태그들을 선택한다.
:password	\$(".password")	input 태그 중 type="password"가 포함된 태그들을 선택한다.
:radio	\$(".radio")	input 태그 중 type="radio"가 포함된 태그들을 선택한다.
:checkbox	\$(".checkbox")	input 태그 중 type="checkbox"가 포함된 태그들을 선택한다.
:submit	\$(".submit")	input 태그 중 type="submit"가 포함된 태그들을 선택한다.
:reset	\$(".reset")	input 태그 중 type="reset"이 포함된 태그들을 선택한다.
:button	\$(".button")	input 태그 중 type="button"이 포함된 태그들을 선택한다.
:enable	\$(".enable")	input 태그 중 사용 가능한 태그들을 선택한다.
:disable	\$(".disable")	input 태그 중 사용 불가능한 태그들을 선택한다.

jQuery 이벤트		
유형	용법	설명
\$(선택자).이벤트()	<pre>\$(“p”).click(function () { // p태그가 클릭되었을 때 실행할 jQuery 내용 });</pre>	
click()	<code>\$(“p”).click(function() { /* jQuery */ });</code>	p태그가 클릭 되었을 때
dblclick()	<code>\$(“p”).dblclick(function() { /* jQuery */ });</code>	p태그가 더블클릭 되었을 때
change()	<code>\$(“input”).change(function() { /* jQuery */ });</code>	input태그의 내용이 바뀌었을 때
blur()	<code>\$(“p”).blur(function() { /* jQuery */ });</code>	p태그가 포커스를 잃었을 때
focus()	<code>\$(“p”).focus(function() { /* jQuery */ });</code>	p태그가 포커스를 얻었을 때
focusin()	<code>\$(“div”).focusin(function() { /* jQuery */ });</code>	div태그의 안쪽에 있는 input이 포커스를 얻었을 때
focusout()	<code>\$(“div”).focusout(function() { /* jQuery */ });</code>	div태그의 안쪽에 있는 input이 포커스를 잃었을 때
keydown()	<code>\$(“input”).keydown(function() { /* jQuery */ });</code>	input태그에서 키가 눌리기 시작하는 순간
keypress()	<code>\$(“input”).keypress(function() { /* jQuery */ });</code>	input태그에서 키가 눌리고 있을 때
keyup()	<code>\$(“input”).keyup(function() { /* jQuery */ });</code>	input태그에서 키가 눌렀다가 다시 올라오는 순간
mousedown()	<code>\$(“div”).mousedown(function() { /* jQuery */ });</code>	div태그에서 왼쪽 마우스 버튼이 눌리기 시작하는 순간
mouseup()	<code>\$(“div”).mouseup(function() { /* jQuery */ });</code>	div태그에서 왼쪽 마우스 버튼이 눌렀다가 다시 올라오는 순간
mousemove()	<code>\$(document).mousemove(function() { /* jQuery */ });</code>	문서 내에서 마우스가 움직일 때
mouseout()	<code>\$(“p”).mouseout(function() { /* jQuery */ });</code>	p태그에서 마우스 포인터가 나갈 때 (자식 요소들도 상속)
mouseover()	<code>\$(“p”).mouseover(function() { /* jQuery */ });</code>	p태그로 마우스 포인터가 들어올 때 (자식 요소들도 상속)
mouseenter()	<code>\$(“p”).mouseenter(function() { /* jQuery */ });</code>	p태그에서 마우스 포인터가 나갈 때
mouseleave()	<code>\$(“p”).mouseleave(function() { /* jQuery */ });</code>	p태그로 마우스 포인터가 들어올 때
hover()	<pre>\$(“p”).hover(function(){ /* 롤오버 시 수행할 jQuery */ }, function(){ /* 롤아웃 시 수행할 jQuery */ });</pre>	p태그에서 마우스 포인터가 들어오고 나갈 때 (mouseenter, mouseleave 이벤트를 작동시킨 것과 같음)
on()	<code>\$(“p”).on(“click”, function() { /* jQuery */ });</code>	p태그에 특정 이벤트를 연결한다.
off()	<code>\$(“p”).off(“click”);</code>	p태그에 on()으로 연결한 특정 이벤트를 제거한다.
one()	<code>\$(“p”).one(“click”, function() { /* jQuery */ });</code>	p태그에 한번만 실행되는 특정 이벤트를 연결한다.
ready()	<code>\$(document).ready(function() { /* jQuery */ });</code>	문서가 완전히 로드되었을 때
resize()	<code>\$(window).resize(function() { /* jQuery */ });</code>	브라우저의 크기가 변경되었을 때
scroll()	<code>\$(document).scroll(function() { /* jQuery */ });</code>	문서에서 스크롤을 사용했을 때
select()	<code>\$(“input”).select(function() { /* jQuery */ });</code>	input태그에서 글을 마우스로 드래그하여 선택했을 때
submit()	<code>\$(“form”).submit(function() { /* jQuery */ });</code>	form태그의 데이터가 전송될 때
trigger()	<pre>\$(“p”).mouseover(function() { \$(“div”).trigger(“click”) });</pre>	p태그로 마우스 포인터가 들어왔을 때 div태그에 연결된 click 이벤트를 실행시킨다.
event.pageX	<pre>\$(“document”).mousemove(function(event) { \$(“span”).text(event.pageX+“, ”+event.pageY); });</pre>	현재 마우스 포인터의 위치 좌표를 반환한다. ※ 좌표의 기준점은 문서의 왼쪽 상단 끝 부분임.
event.pageY		

jQuery Effect 메소드				
유형	문법		용법	설명
\$ (선택자).메소드()		\$("p").click(function() { \$(this).hide(); });		
hide()	\$(선택자).hide(속도,가속도,콜백함수);	\$("p").hide();		요소를 사라지게 한다.
show()	\$(선택자).show(속도,가속도,콜백함수);	\$("p").show();		요소를 나타나게 한다.
toggle()	\$(선택자).toggle(속도,가속도,콜백함수);	\$("p").toggle();		요소를 사라짐/나타남 토글한다.
fadeIn()	\$(선택자).fadeIn(속도,가속도,콜백함수);	\$("p").fadeIn();		요소의 투명도를 조정하여 나타나게 한다.
fadeOut()	\$(선택자).fadeOut(속도,가속도,콜백함수);	\$("p").fadeOut();		요소의 투명도를 조정하여 사라지게 한다.
fadeTo()	\$(선택자).fadeTo(속도,투명도,가속도,콜백함수);	\$("p").fadeTo("fast",0.4);		요소의 투명도를 조정한다. (0.0 ~ 1.0)
fadeToggle()	\$(선택자).fadeToggle(속도,가속도,콜백함수);	\$("p").fadeToggle();		요소의 투명도를 조정하여 사라짐/나타남 토글한다.
slideDown()	\$(선택자).slideDown(속도,가속도,콜백함수);	\$("p").slideDown();		요소를 아래로 슬라이드하며 나타나게 한다.
slideUp()	\$(선택자).slideUp(속도,가속도,콜백함수);	\$("p").slideUp();		요소를 위로 슬라이드하며 사라지게 한다.
slideToggle()	\$(선택자).slideToggle(속도,가속도,콜백함수);	\$("p").slideToggle();		요소를 아래 위로 슬라이드하며 사라짐/나타남 토글한다.
메소드 내 매개변수	속도	밀리초(ms)	해당 숫자만큼의 시간 동안 애니메이션을 재생시킨다. (기본값 : 400)	
		"slow" / "fast"	애니메이션을 느리게 / 빠르게 재생시킨다.	
	가속도	"swing"	애니메이션 시작/끝 지점에서는 느리게, 중간 지점에서는 빠르게 재생시킨다.	
		"linear"	애니메이션의 처음부터 끝까지 동일한 속도로 재생시킨다.	
속도 필수 속도 옵션	콜백함수		\$("p").hide("slow", function(){ \$(this).after("<p>애니메이션 끝남</p>"); });	메소드가 완전히 끝난 후 실행시킬 함수 (시간차가 있는 애니메이션의 경우 이전 애니메이션이 끝나지 않았음에도 불구하고 다음 라인에 있는 명령문이 실행되는 경우를 막기 위해서도 사용한다.)
animate()	\$(선택자).animate(스타일,속도,가속도,콜백함수);		\$("p").animate({ left:"300px", top:"100px" }, "slow", "linear");	요소를 현재 모습에서 지정한 CSS 스타일 모습으로 움직이게 한다. 하나의 animate 메소드(큐)에 여러 스타일을 작성하면 동시에 작동된다. 여러 개의 큐를 작성하면 순차적으로 실행되며 실행이 예약된다.
	\$(선택자).animate(스타일,{옵션});		\$("p").animate({ left:"300px", top:"100px" }, { duration: 1000, easing: "linear", complete: function() {...} });	
	스타일	left/right/top/bottom/margin/padding/height/width/fontSize/opacity/borderWidth/.....	스타일의 이름은 CamelCase의 형태로 작성한다. (font-size → fontSize) 스타일의 값은 left:"300px"의 형태로 작성하며, 오직 숫자 값을 가지는 스타일만 허용된다. ("show", "hide", "toggle"은 예외적으로 가능) left:"+=10px"의 형태로 상대적인 값(증감값)을 넣을 수 있다.	
	옵션	duration	애니메이션의 속도 밀리초(ms) / "slow" / "fast"	
		easing	움직임의 가속도 "swing" / "linear"	
		complete	애니메이션이 끝난 후 실행시킬 함수 (=콜백함수)	
step		애니메이션의 매 스텝마다 실행시킬 함수 function(now, fx) { /* ... */ }		
delay()	\$(선택자).delay(속도,큐이름).hide();		\$("p").delay(3000).hide();	애니메이션 큐의 지연시간을 설정한다.
stop()	\$(선택자).stop(stopAll,goToEnd);		\$("p").stop(false, false);	현재 재생되고 있는 애니메이션 큐를 중지한다. stopAll이 false인 경우 현재 큐만 중지시킨 후 다음 큐를 재생하고, true인 경우 모든 큐를 중지한다. goToEnd가 true인 경우 중지 후 마지막 상태의 스타일로 중지된다.
finish()	\$(선택자).finish(큐이름);		\$("p").finish();	현재 재생되고 있는 애니메이션 큐를 중지시키고 마지막 상태의 스타일로 중지한다. \$("p").stop(false, false);와 동일한 결과임
queue()	\$(선택자).queue(큐이름);	var a = \$("p").queue("fx").length;		애니메이션에서 실행 될 큐들의 속성을 표시한다.
	\$(선택자).queue(함수);	\$("p").queue(function(){ \$(this).css(.....); \$(this).dequeue(); });		애니메이션 큐를 일시정지하고 animate 메소드 이외의 타 메소드를 적용한다.
dequeue()	\$(선택자).dequeue(큐이름);			queue 메소드로 정지시켰던 애니메이션 큐를 재개한다.
clearQueue()	\$(선택자).clearQueue(큐이름);		\$("p").clearQueue()	남은 모든 애니메이션 큐를 삭제하고 중지한다. (stop 메소드는 animate 메소드만 중지시키지만, clearQueue 메소드는 모든 큐 메소드들을 삭제한다.)

jQuery HTML/CSS 메소드			
유형	문법	용법	설명
text()	<code>\$(선택자).text();</code>	<code>var a = \$("p").text();</code>	선택된 요소들의 태그의 텍스트 내용을 가져온다.(태그는 삭제됨)
	<code>\$(선택자).text(텍스트내용);</code>	<code>\$("p").text("ㅋㅋㅋ");</code>	선택된 요소들의 태그에 텍스트 내용을 삽입한다.(태그는 삭제됨)
html()	<code>\$(선택자).html();</code>	<code>var a = \$("p").html();</code>	선택된 요소들의 첫 번째 태그의 태그를 포함한 모든 내용을 가져온다.
	<code>\$(선택자).html(태그내용);</code>	<code>\$("p").html("ㅋㅋㅋㅋ");</code>	선택된 요소들의 태그에 태그를 포함한 모든 내용을 삽입한다.
val()	<code>\$(선택자).val();</code>	<code>var a = \$("input:text").val();</code>	선택된 폼 요소들의 첫 번째 태그의 value 값을 가져온다.
	<code>\$(선택자).val(value값);</code>	<code>\$("input:text").val("ㅋㅋㅋ");</code>	선택된 폼 요소들의 태그에 value값을 삽입한다.
attr()	<code>\$(선택자).attr(속성명);</code>	<code>var a = \$("img").attr("width");</code>	선택된 요소들의 태그의 첫 번째 태그의 속성값을 가져온다.
	<code>\$(선택자).attr(속성명, 속성값);</code>	<code>\$("img").attr("src", "1.jpg");</code>	선택된 요소들의 태그에 해당 속성값을 삽입한다.
	<code>\$(선택자).attr({속성명:속성값, 속성명:속성값...});</code>	<code>\$("img").attr({src:"1.jpg", width:"100px"});</code>	
append()	<code>\$(선택자).append(콘텐츠...);</code>	<code>\$("ul").append("마지막");</code>	선택된 요소의 내용 중 끝 부분에 콘텐츠를 삽입한다.
prepend()	<code>\$(선택자).prepend(콘텐츠...);</code>	<code>\$("ul").prepend("첫번째");</code>	선택된 요소의 내용 중 처음 부분에 콘텐츠를 삽입한다.
after()	<code>\$(선택자).after(콘텐츠...);</code>	<code>\$("p").after("<p>이후</p>");</code>	선택된 요소의 뒤에 콘텐츠를 삽입한다.
before()	<code>\$(선택자).before(콘텐츠...);</code>	<code>\$("p").before("<p>이전</p>");</code>	선택된 요소의 이전에 콘텐츠를 삽입한다.
remove()	<code>\$(선택자).remove(선택자);</code>	<code>\$("div").remove();</code>	선택된 요소와 요소의 자식객체, 콘텐츠를 삭제한다.
empty()	<code>\$(선택자).empty();</code>	<code>\$("div").empty();</code>	선택된 요소의 자식객체, 콘텐츠를 삭제한다.
detach()	<code>\$(선택자).detach();</code>	<code>\$("div").detach();</code>	선택된 요소의 자식객체, 콘텐츠를 삭제한다.(data 및 event 보존 가능)
addClass()	<code>\$(선택자).addClass(클래스이름);</code>	<code>\$("div").addClass("leftbox");</code>	선택된 요소에 클래스를 추가한다.
removeClass()	<code>\$(선택자).removeClass(클래스이름);</code>	<code>\$("div").removeClass("leftbox");</code>	선택된 요소에서 클래스를 삭제한다. (빈 매개변수 → 모든 클래스 삭제)
toggleClass()	<code>\$(선택자).toggleClass(클래스이름);</code>	<code>\$("div").toggleClass("leftbox");</code>	선택된 요소에 클래스를 추가/삭제 토글한다.
css()	<code>\$(선택자).css(속성명);</code>	<code>\$("p").css("font-size");</code>	선택된 첫 번째 요소에서 스타일 속성명의 속성값을 가져온다.
	<code>\$(선택자).css(속성명, 속성값);</code>	<code>\$("p").css("font-size", "12px");</code>	선택된 요소의 스타일 속성명의 속성값을 설정한다.
	<code>\$(선택자).css({속성명:속성값, 속성명:속성값, 속성명:속성값...});</code>		
width()	<code>\$(선택자).width();</code>	<code>var a = \$("div").width();</code>	선택된 첫 번째 요소의 마진, 보더, 패딩을 제외한 너비 값을 가져온다.
	<code>\$(선택자).width(값);</code>	<code>\$("div").width(500);</code> <code>\$("div").width("50em");</code>	선택된 요소의 마진, 보더, 패딩을 제외한 너비 값을 설정한다.
height()	<code>\$(선택자).height();</code>	<code>var a = \$("div").height();</code>	선택된 첫 번째 요소의 마진, 보더, 패딩을 제외한 높이 값을 가져온다.
	<code>\$(선택자).height(값);</code>	<code>\$("div").height(300);</code> <code>\$("div").height("15pt");</code>	선택된 요소의 마진, 보더, 패딩을 제외한 높이 값을 설정한다.
innerWidth()	<code>\$(선택자).innerWidth();</code>	<code>\$("div").innerWidth();</code>	선택된 요소의 마진, 보더를 제외한 너비 값을 가져온다.
innerHeight()	<code>\$(선택자).innerHeight();</code>	<code>\$("div").innerHeight();</code>	선택된 요소의 마진, 보더를 제외한 높이 값을 가져온다.
outerWidth()	<code>\$(선택자).outerWidth(includeMargin);</code>	<code>\$("div").outerWidth();</code>	선택된 요소의 마진을 제외한 너비 값을 가져온다.
		<code>\$("div").outerWidth(true);</code>	선택된 요소의 마진을 포함한 너비 값을 가져온다.
outerHeight()	<code>\$(선택자).outerHeight(includemargin);</code>	<code>\$("div").outerHeight();</code>	선택된 요소의 마진을 제외한 높이 값을 가져온다.
		<code>\$("div").outerHeight(true);</code>	선택된 요소의 마진을 포함한 높이 값을 가져온다.

jQuery Traversing 메소드			
유형	문법	용법	설명
parent()	\$(선택자).parent(필터);	\$("p").parent();	선택된 요소의 직계 부모 요소를 선택한다.
parents()	\$(선택자).parents(필터);	\$("p").parents();	선택된 요소의 모든 조상 요소를 선택한다.
parentsUntil()	\$(선택자).parentsUntil(stop, 필터);	\$("p").parentsUntil("div");	선택된 요소의 모든 조상 요소 중 stop 이전까지의 요소들을 선택한다.
children()	\$(선택자).children(필터);	\$("div").children();	선택된 요소의 직계 자식 요소들을 선택한다.
find()	\$(선택자).find(필터);	\$("div").find("*");	선택된 요소의 모든 자손 요소 중 필터에 해당하는 요소들을 선택한다.
siblings()	\$(선택자).siblings(필터);	\$("span").siblings("p");	선택된 요소의 형제 요소들을 선택한다.
next()	\$(선택자).next(필터);	\$("span").next();	선택된 요소의 형제 요소들 중 바로 아래 동생 요소를 선택한다.
nextAll()	\$(선택자).nextAll(필터);	\$("span").nextAll();	선택된 요소의 형제 요소들 중 아래 동생 요소들을 선택한다.
nextUntil()	\$(선택자).nextUntil(stop, 필터);	\$("span").nextUntil("p");	선택된 요소의 형제 요소들 중 stop 이전까지의 동생 요소들을 선택한다.
prev()	\$(선택자).prev(필터);	\$("span").prev();	선택된 요소의 형제 요소들 중 바로 위 형 요소를 선택한다.
prevAll()	\$(선택자).prevAll(필터);	\$("span").prevAll();	선택된 요소의 형제 요소들 중 위 형 요소들을 선택한다.
prevUntil()	\$(선택자).prevUntil(필터);	\$("span").prevUntil("p");	선택된 요소의 형제 요소들 중 stop 이전까지의 형 요소들을 선택한다.
first()	\$(선택자).first();	\$("div p").first();	선택된 요소의 자식요소 중 처음으로 등장하는 요소를 선택한다.
last()	\$(선택자).last();	\$("div p").last();	선택된 요소의 자식요소 중 마지막으로 등장하는 요소를 선택한다.
eq()	\$(선택자).eq(인덱스번호);	\$("p").eq(1);	선택된 요소 중 n번째로 등장하는 요소를 선택한다. (0부터 시작)
filter()	\$(선택자).filter(선택자필터);	\$("p").filter(".demo");	선택된 요소 중 선택자 필터에 해당하는 요소들만 선택한다.
not()	\$(선택자).not(선택자필터);	\$("p").not(".demo");	선택된 요소 중 선택자 필터에 해당하지 않는 요소들만 선택한다.