



## ❖ jQuery

- 2006년 미국의 존 레식(John Resig)이 만든 자바스크립트 라이브러리
- 자바스크립트로 만든 함수들의 집합
- 오픈소스로 제공되어 누구나 무료로 사용가능

### ◆ 코드의 단순화

- \* Javascript의 코드 형식을 좀 더 직관적으로 이해할 수 있도록 짧고 단순한 코드 형태로 변형하여 제공

### ◆ 호환성 문제 해결

- \* Javascript의 호환성(크로스 브라우징) 문제 해결

### ◆ 쉽고 편리한 애니메이션 효과 기능 구현

- \* 쉽고 다양한 효과의 애니메이션 메서드 제공

## ❖ jQuery 라이브러리 연동

### ◆ 직접 다운 받는 방법

1. jQuery 사이트(<http://jquery.com/download/>) 에 접속해서 직접 다운 받는다.
2. js 폴더를 만들어 폴더 안에 저장한다.
3. <head> 영역에 다음과 같이 파일을 불러온다.

```
<script type="text/javascript" src="js/파일이름"></script>
```

### ◆ 네트워크 전송 방식 ( CDN )

- \* 네트워크 전송 방식(CDN : Content Delivery Network)은 라이브러리 파일을 서버의 저장소에서 불러오는 방식
- \* 여러 경로로 제공 받을 수 있다.

서버	src 경로
jQuery CDN	<a href="http://code.jquery.com/jquery-3.4.1.min.js">http://code.jquery.com/jquery-3.4.1.min.js</a>
	<a href="http://code.jquery.com/jquery-latest.js">http://code.jquery.com/jquery-latest.js</a>
Google CDN	<a href="http://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">http://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js</a>
Microsoft CDN	<a href="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.4.1.min.js">http://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.4.1.min.js</a>

## ❖ jQuery 함수

### ◆ jQuery 함수 사용

jQuery(함수이름);

⊠ 매개변수로 함수이름을 넣어주면 함수의 내용이 onload 처리된다.

```
<script type="text/javascript">
    function func01() {
        var h1 = document.getElementById("h1");
        h1.style.color = "#0000ff";
        h1.style.backgroundColor = "#d5d5d5";
        h1.style.fontSize = "50px";
        h1.style.padding = "15px";
        h1.style.margin = "0";
        h1.innerHTML = "Hello jQuery";
    }
    function func02() {
        alert("안녕하세요. 제이쿼리");
    }
    jQuery(func01);
    jQuery(func02);
</script>
</head>
<body>
    <h1 id="h1"></h1>
</body>
```

jq\_func01.html

## ❖ \$ 객체의 사용

- ◆ \$는 jQuery의 모든 기능을 담고 있는 객체(최상위)이자 함수
- ◆ jQuery함수가 특정 함수 호출하면 그 함수는 \$ 객체 반환
- ◆ \$ 객체는 자신이 jQuery()를 대신할 수 있다.

\$(함수이름);

```
<script type="text/javascript">
  function func01() {
    var h1 = document.getElementById("h1");
    h1.style.color = "#0000ff";
    h1.style.backgroundColor = "#d5d5d5";
    h1.style.fontSize = "50px";
    h1.style.padding = "15px";
    h1.style.margin = "0";
    h1.innerHTML = "Hello jQuery";
  }
  function func02() {
    alert("안녕하세요. 제이쿼리");
  }
  $(func01);
  $(func02);
</script>
</head>
<body>
  <h1 id="h1"></h1>
</body>
```

jq\_func02.html

## ❖ \$ 객체의 사용

- ◆ \$함수에 선택항목을 매개변수로 넣어주면 그 선택항목을 객체로 반환받는다.

`$("선택항목");`

```
<script type="text/javascript">
    function func01() {
        var h1 = $("h1");
        h1.style.color = "#0000ff";
        h1.style.backgroundColor = "#d5d5d5";
        h1.style.fontSize = "50px";
        h1.style.padding = "15px";
        h1.style.margin = "0";
        h1.innerHTML = "Hello jQuery";
    }
    function func02() {
        alert("안녕하세요. 제이쿼리");
    }
    $(func01);
    $(func02);
</script>
</head>
<body>
    <h1 id="h1"></h1>
</body>
```

jq\_func03.html

## ❖ jQuery 선택자

### ◆ 기본 선택자 종류

구분	종류	사용법	설 명
직접 선택자	전체 선택자	<code>\$("*")</code>	모든 요소 선택
	아이디 선택자	<code>\$("#아이디명")</code>	id 속성에 지정한 값을 가진 요소 선택
	클래스 선택자	<code>\$(".클래스명")</code>	class 속성에 지정한 값을 가진 요소를 선택
	요소명 선택자	<code>\$("요소명")</code>	지정한 요소명과 일치하는 요소들만 선택
	그룹선택자	<code>\$("선택1, 선택2, ...")</code>	선택1, 선택2, ... 에 지정된 요소들을 한번에 선택
인접 관계 선택자	부모 요소 선택자	<code>\$("요소선택").parent()</code>	선택한 요소의 부모 요소를 선택
	상위 요소 선택자	<code>\$("요소선택").parents()</code>	선택한 요소의 상위 요소를 모두 선택
	하위 요소 선택자	<code>\$("요소선택 하위요소")</code>	선택한 요소의 지정한 하위 요소를 선택
	자식 요소 선택자	<code>\$("요소선택 &gt; 자식요소")</code>	선택한 요소를 기준 자식 관계에 지정한 요소만 선택
	자식 요소들 선택자	<code>\$("요소선택").children()</code>	선택한 요소의 모든 자식요소를 선택
	이전 요소 선택자	<code>\$("요소선택").prev()</code>	선택한 요소의 바로 이전 형제 요소 만 선택
	이전 요소들 선택자	<code>\$("요소선택").prevAll()</code>	선택한 요소의 바로 이전 형제 요소 모두 선택
	지정 이전 요소들 선택자	<code>\$("요소선택").prevUntil("요소명")</code>	선택한 요소부터 지정한 요소의 이전 요소까지 모두 선택
	다음 요소 선택자	<code>\$("요소선택").next()</code> <code>\$("요소선택+다음요소")</code>	선택한 요소의 다음 형제 요소를 선택
	다음 요소들 선택자	<code>\$("요소선택").nextAll()</code>	선택한 요소의 다음 형제 요소 모두를 선택
	형제 요소들 선택자	<code>\$("요소선택").siblings()</code>	선택 요소의 모든 형제 요소 선택

## ❖ jQuery 선택자

### ◆ 위치 탐색 선택자 종류

종류	사용법	설 명
\$(“요소선택:first”)	\$(“li:first”)	전체 <li> 요소 중 첫 번째 요소만 선택
\$(“요소선택”).first()	\$(“li”).first()	
\$(“요소선택:last”)	\$(“li:last”)	전체 <li> 요소 중 마지막 요소만 선택
\$(“요소선택”).last()	\$(“li”).last()	
\$(“요소선택:odd”)	\$(“li:odd”)	<li> 요소 그룹 중 짝수 번째 요소만 선택
\$(“요소선택:even”)	\$(“li:even”)	<li> 요소 그룹 중 홀수 번째 요소만 선택
\$(“요소선택:first-of-type”)	\$(“li:first-of-type”)	<li> 요소 그룹 중 첫 번째 요소만 선택
\$(“요소선택:last-of-type”)	\$(“li:last-of-type”)	<li> 요소 그룹 중 마지막 요소만 선택
\$(“요소선택:nth-child(숫자)”)	\$(“li:nth-child(3)”)	<li> 요소 그룹 중 세 번째 요소만 선택
\$(“요소선택:nth-child(숫자n)”)	\$(“li:nth-child(3n)”)	<li> 요소 그룹 중 3의 배수 번째 요소만 선택
\$(“요소선택:nth-last-of-type(숫자)”)	\$(“li:nth-last-of-type(2)”)	<li> 요소 그룹 중 마지막 위치에서 두 번째 요소만 선택
\$(“요소선택:only-child”)	\$(“li:only-child”)	부모 요소 내에 <li> 요소가 1개 뿐인 <li> 요소만 선택
\$(“요소선택:eq(index)”)	\$(“li:eq(2)”)	<li> 요소중 index가 2인 요소를 선택
\$(“요소선택”).eq(index)	\$(“li”).eq(2)	
\$(“요소선택:gt(index)”)	\$(“li:gt(1)”)	<li> 요소중 index가 1보다 큰 요소를 선택
\$(“요소선택:lt(index)”)	\$(“li:lt(1)”)	<li> 요소중 index가 1보다 작은 요소를 선택
\$(“요소선택”).slice(index)	\$(“li”).slice(2)	<li> 요소중 index가 2부터 이후 요소를 선택



## ❖ jQuery 선택자

### ◆ 속성 탐색 선택자 종류

종류	사용법	설 명
<code>\$("요소선택[속성]")</code>	<code>\$("li[title]")</code>	<li> 요소 중 title 속성이 포함된 요소만 선택
<code>\$("요소선택[속성=값]")</code>	<code>\$("li[title='리스트']")</code>	<li> 요소 중 title 속성값이 '리스트' 인 요소만 선택
<code>\$("요소선택[속성^=텍스트]")</code>	<code>\$("a[href^='http://']")</code>	<a> 요소 중 href 속성 값이 "http://" 로 시작하는 요소만 선택
<code>\$("요소선택[속성\$=텍스트]")</code>	<code>\$("a[href\$='.com']")</code>	<a> 요소 중 href 속성 값이 ".com" 으로 끝나는 요소만 선택
<code>\$("요소선택[href*=텍스트]")</code>	<code>\$("a[href*='easypub']")</code>	<a> 요소 중 href 속성 값중 "easypub" 이 포함된 요소만 선택
<code>\$("요소선택:hidden")</code>	<code>\$("li:hidden")</code>	<li> 요소 중 숨겨진 요소만 선택
<code>\$("요소선택:visible")</code>	<code>\$("li:visible")</code>	<li> 요소 중 보이는 요소만 선택
<code>\$(":text")</code>	<code>\$(":text")</code>	<input> 요소 중 type 속성값이 "text"인 요소만 선택
<code>\$(":selected")</code>	<code>\$(":selected")</code>	selected 속성이 적용된 요소만 선택
<code>\$(":checked")</code>	<code>\$(":checked")</code>	checked 속성이 적용된 요소만 선택

## ❖ jQuery 선택자

### ◆ 콘텐츠 탐색 선택자 종류

종류	사용법	설 명
<code>\$("요소선택:contains(텍스트)")</code>	<code>\$("li:contains('내용2')")</code>	<code>&lt;li&gt;</code> 요소 중 '내용2'라는 텍스트를 포함하는 요소만 선택
<code>\$("요소선택").contents()</code>	<code>\$("p").contents()</code>	<code>&lt;p&gt;</code> 요소의 하위 요소 중 가장 가까운 하위 요소 선택
<code>\$("요소선택:has(요소명)")</code>	<code>\$("li:has('span')")</code>	<code>&lt;li&gt;</code> 요소 중 <code>&lt;span&gt;</code> 을 포함하는 요소만 선택
<code>\$("요소선택").has(요소명)</code>	<code>\$("li").has("span")</code>	
<code>\$("요소선택:not(:제외요소)")</code>	<code>\$("li:not(:first)")</code>	<code>&lt;li&gt;</code> 요소 중 첫 번째 요소를 제외한 요소만 선택
<code>\$("요소선택").not(:제외요소)</code>	<code>\$("li").not(:first)</code>	
<code>\$("요소선택").filter(필터요소)</code>	<code>\$("li").filter("list2")</code>	<code>&lt;li&gt;</code> 요소 중 class 속성값이 'list2'인 요소만 선택
<code>\$("요소선택1").find(요소선택2)</code>	<code>\$("li").find("strong")</code>	<code>&lt;li&gt;</code> 요소의 하위 요소인 <code>&lt;strong&gt;</code> 요소만 선택
<code>\$("요소선택1").closest(요소선택2)</code>	<code>\$("strong").closest("div")</code>	<code>&lt;strong&gt;</code> 을 감싸는 상위 <code>&lt;div&gt;</code> 요소 중 가장 가까운 상위 요소 선택
<code>end()</code>	<code>\$("li").children("a").end()</code>	필터링이 일어나기 이전의 요소인 <code>&lt;li&gt;</code> 선택

## ❖ jQuery 선택자

### ◆ 선택자와 함께 사용하면 편리한 method

종류	사용법	설 명
is(":요소상태")	<code>\$(<b>"li"</b>).is(<b>":visible"</b>)</code>	선택한 <li> 요소가 보이면 true 반환
<code>\$.noConflict()</code>	<code>var m = \$.noConflict(); 변수("요소선택")</code>	선택자 메서드를 사용할 때 '\$'를 앞에 붙여 사용한다. \$.onConflict()를 사용하면 '\$' 메서드를 변수에 참조시켜 '\$' 문자대신 임의의 문자를 지정해서 사용할 수 있다.
get()	<code>\$(<b>"li"</b>).get(0).style.color=<b>"red"</b>;</code>	jQuery로 선택한 요소에 get(0) 메서드를 적용하면 자바스크립트에 DOM 선택자로 선택한 태그로 인식되어 이전 자바스크립트의 DOM 방식을 따른다.
each() / \$.each()	<code>\$(<b>"li"</b>).each( function(매개변수1,매개변수2){}); \$.each(\$(<b>"li"</b>), function(매개변수1,매개변수2){});</code>	<p>선택한 요소들을 각각 순차적으로 접근해 선택해 온다.</p> <p><b>매개변수1 : 순차적으로 접근해 선택된 요소의 인덱스 저장</b>  <b>매개변수2 : 순차적으로 선택된 요소가 저장</b></p>
<code>\$.map()</code>	<code>\$.map(배열, 함수)</code>	배열 데이터를 변환시켜 다시 새로운 배열 객체를 만든다.

## ❖ jQuery 객체 조작 Method

### ◆ 속성 조작 method

종류	사용법	설 명
html()	<code>\$("요소선택").html();</code> <code>\$("요소선택").html("새요소");</code>	선택한 요소의 하위 요소들을 반환하거나 새 요소로 반환
text()	<code>\$("요소선택").text();</code> <code>\$("요소선택").text("새텍스트");</code>	선택한 요소에 포함된 텍스트를 반환하거나 새 텍스트로 변환
css("속성") / css("속성", "값")	<code>\$("요소선택").css("color");</code> <code>\$("요소선택").css("color", "blue");</code>	선택한 요소의 스타일(CSS) 속성값을 반환 또는 변환
attr("속성") / attr("속성", "값")	<code>\$("요소선택").attr("href");</code> <code>\$("요소선택").attr("href", "main.html");</code>	선택한 요소의 속성값을 반환 또는 변환
removeAttr("속성")	<code>\$("요소선택").removeAttr("title");</code>	선택한 요소의 지정 속성을 삭제
addClass("클래스 값")	<code>\$("요소선택").addClass("abc");</code>	선택한 요소에 지정한 클래스를 생성 또는 추가
removeClass("클래스 값")	<code>\$("요소선택").removeClass("abc");</code>	선택한 요소의 지정한 클래스 값을 삭제
toggleClass("클래스 값")	<code>\$("요소선택").toggleClass("abc");</code>	선택한 요소의 지정 클래스 값을 생성, 삭제를 반복
hasClass("클래스 값")	<code>\$("요소선택").hasClass("abc");</code>	선택한 요소의 지정 클래스 값의 포함 여부 검사 있으면 true / 없으면 false
prop("속성")	<code>\$("요소선택").prop("tagName");</code>	선택한 태그의 속성들을 반환
val() / val(값)	<code>\$("요소선택").val();</code> <code>\$("요소선택").val("홍길동");</code>	선택한 입력 요소에 지정한 value 값을 반환 또는 변환

## ❖ jQuery 객체 조작 Method

### ◆ 수치 조작 method

종류	사용법	설 명
height()	<code>\$("요소선택").height();</code> <code>\$("요소선택").height(100);</code>	안쪽 여백과 선을 제외한 높이를 반환하거나 변환
width()	<code>\$("요소선택").width();</code> <code>\$("요소선택").width(100);</code>	안쪽 여백과 선을 제외한 너비를 반환하거나 변환
innerHeight()	<code>\$("요소선택").innerHeight();</code> <code>\$("요소선택").innerHeight(300);</code>	안쪽 여백을 포함한 높이를 반환하거나 변환
innerWidth()	<code>\$("요소선택").innerWidth();</code> <code>\$("요소선택").innerWidth(100);</code>	안쪽 여백을 포함한 너비를 반환하거나 변환
outerHeight()	<code>\$("요소선택").outerHeight();</code> <code>\$("요소선택").outerHeight(300);</code>	선과 안쪽 여백을 포함한 높이를 반환하거나 변환
outerWidth()	<code>\$("요소선택").outerWidth();</code> <code>\$("요소선택").outerWidth(100);</code>	선과 안쪽 여백을 포함한 너비를 반환하거나 변환
position()	<code>\$("요소선택").position().left;</code> <code>\$("요소선택").position().top;</code>	선택한 요소의 포지션 위치값을 반환
offset()	<code>\$("요소선택").offset().left;</code> <code>\$("요소선택").offset().top;</code>	선택한 요소가 문서에서 수평/수직으로 얼마나 떨어져 있는 지에 대한 값을 반환
scrollLeft()	<code>\$("요소선택").scrollLeft();</code>	브라우저의 수평 스크롤 이동 높이를 반환
scrollTop()	<code>\$("요소선택").scrollTop();</code>	브라우저의 수직 스크롤 이동 너비를 반환

## ❖ jQuery 객체 조작 Method

### ◆ 객체 편집 method

종류	사용법	설 명
before()	<code>\$("요소선택").before("새 요소");</code>	선택한 요소 이전 위치에 새 요소 추가
after()	<code>\$("요소선택").after("새 요소");</code>	선택한 요소 다음 위치에 새 요소 추가
append()	<code>\$("요소선택").append("새 요소");</code>	선택한 요소 마지막 위치에 새 요소 추가
appendTo()	<code>\$("새 요소").appendTo("요소선택");</code>	선택한 요소 마지막 위치에 새 요소 추가
prepend()	<code>\$("요소선택").prepend("새 요소");</code>	선택한 요소 맨 앞 위치에 새 요소 추가
prependTo()	<code>\$("요소선택").prependTo("요소선택");</code>	선택한 요소 맨 앞 위치에 새 요소 추가
insertBefore()	<code>\$("요소선택").insertBefore("새 요소");</code>	선택한 요소 이전 위치에 새 요소 추가
insertAfter()	<code>\$("요소선택").insertAfter("새 요소");</code>	선택한 요소 다음 위치에 새 요소 추가
clone()	<code>\$("요소선택").clone(true   false);</code>	선택한 요소 복제( true : 하위요소까지 복제 / false : 선택 요소만 복제 )
empty()	<code>\$("요소선택").empty();</code>	선택한 요소의 하위 내용들 모두 삭제
remove()	<code>\$("요소선택").remove();</code>	선택한 요소를 삭제
replaceAll() replaceWith()	<code>\$("요소선택").replaceAll("요소선택");</code> <code>\$("요소선택").replaceWith("새 요소");</code>	선택한 요소들을 새 요소로 교체
unwrap()	<code>\$("요소선택").unwrap();</code>	선택한 요소의 부모 요소를 삭제
wrap()	<code>\$("요소선택").wrap(새 요소);</code>	선택한 요소를 새 요소로 각각 감싼다.
wrapAll()	<code>\$("요소선택").wrapAll();</code>	선택한 요소를 새 요소로 한꺼번에 감싼다.
wrapInner()	<code>\$("요소선택").wrapInner(새 요소);</code>	선택한 요소 내에 내용을 새 요소로 각각 감싼다.

## ❖ jQuery 이벤트

### ◆ 이벤트 등록 method 종류

구 분	종 류	설 명
로딩 이벤트	load()	선택한 이미지 또는 프레임 요소에 연동된 소스가 로딩이 완료된 후 이벤트가 발생
	ready()	지정한 HTML 문서 객체가 로딩이 완료된 후 이벤트가 발생
마우스 이벤트	click()	선택한 요소를 클릭했을 때 이벤트가 발생
	dblclick()	선택한 요소를 연속으로 두 번 클릭했을 때 이벤트가 발생
	mouseout()	선택한 요소의 영역에서 마우스가 벗어났을 때 이벤트 발생. 이때 하위 요소의 영향을 받는다.
	mouseover()	선택한 요소의 영역에서 마우스를 올렸을 때 이벤트가 발생
	hover()	선택한 요소에 마우스를 올렸을 때와 벗어났을 때 각각 이벤트 발생
	mousedown()	선택한 요소에서 마우스 버튼을 눌렀을 때 이벤트 발생
	mouseup()	선택한 요소에서 마우스 버튼을 눌렀다 떼었을 때 이벤트가 발생
	mouseenter()	선택한 요소 범위 내에서 마우스를 올렸을 때 이벤트가 발생
	mouseleave()	선택한 요소 범위 내에서 마우스가 벗어났을 때 이벤트가 발생
	mousemove()	선택한 요소 범위에서 마우스를 움직였을 때 이벤트가 발생
포커스 이벤트	focus()	선택한 요소에 포커스가 생성되었을 때 이벤트 발생하거나 선택한 요소에 강제로 포커스 생성
	focusin()	선택한 요소에 포커스가 생성되었을 때 이벤트가 발생
	focusout()	선택한 요소에서 포커스가 다른 요소로 이동되었을 때 이벤트가 발생
	blur()	선택한 요소에서 포커스가 다른 요소로 이동되었을 때 이벤트 발생 또는 선택한 요소에 강제로 포커스가 사라지도록 한다.

## ❖ jQuery 이벤트

### ◆ 이벤트 등록 method 종류

구 분	종 류	설 명
키보드 이벤트	keypress()	선택한 요소를 키보드를 눌렀을 때 이벤트 발생, 문자 키를 제외한 키코드값을 반환
	keydown()	선택한 요소에서 키보드를 눌렀을 때 이벤트 발생, 자판의 모든 키코드값 반환
	keyup()	선택한 요소에서 키보드에서 손을 떼었을 때 이벤트가 발생
기 타 이벤트	scroll()	스크롤 바를 움직일 때 이벤트 발생
	change()	선택한 요소에 입력 요소의 값이 변경되었을 때 이벤트가 발생
	error()	선택한 요소에서 에러가 발생했을 때 이벤트가 발생
포커스 이벤트	focus()	선택한 요소에 포커스가 생성되었을 때 이벤트 발생하거나 선택한 요소에 강제로 포커스 생성
	focusin()	선택한 요소에 포커스가 생성되었을 때 이벤트가 발생
	focusout()	선택한 요소에서 포커스가 다른 요소로 이동되었을 때 이벤트가 발생



## ❖ jQuery 그룹 이벤트 등록 및 삭제

### ◆ 그룹 이벤트 등록 method 종류

종 류	설 명
on()	이벤트 대상 요소에 2개 이상의 이벤트를 등록. 사용방식에 따라 이벤트를 등록한 이후에도 동적으로 생성되거나 복제된 요소에도 이벤트가 적용
bind()	이벤트 대상 요소에 2개 이상의 이벤트를 등록
delegate()	선택한 요소의 지정한 하위 요소에 이벤트를 등록 이벤트 등록 이후 동적으로 생성된 이벤트를 등록한 요소와 동일한 새 요소에도 이벤트가 등록
live()	선택한 요소에 한 개 이상의 이벤트를 등록 이벤트 등록된 이후 동적으로 생성된 이벤트를 등록한 요소와 동일한 새 요소에도 이벤트가 등록 제이쿼리 1.9 버전부터 지원안함
one()	선택한 요소에 이벤트를 등록하며, 단 한번만 이벤트가 발생

## ❖ jQuery 효과 및 애니메이션

### ◆ 효과 method 종류

종 류	설 명
<b>hide()</b>	노출되어 있는 요소를 숨김
<b>show()</b>	숨겨져 있던 요소를 노출시킴
<b>toggle()</b>	숨겨진 요소는 노출시키고 노출된 요소는 숨김
<b>fadeIn()</b>	숨겨져 있던 요소가 점점 선명해지면서 노출됨
<b>fadeOut()</b>	노출된 요소가 점점 투명해 지면서 숨겨짐
<b>fadeToggle()</b>	노출된 요소는 점점 투명해 지면서 사라지고 사라진 요소는 선명해지면서 노출됨
<b>fadeTo()</b>	노출 되어 있던 요소에 투명도를 지정한 만큼 숨김
<b>slideDown()</b>	숨겨져 있던 요소가 밑으로 펼쳐지면서 노출됨
<b>slideUp()</b>	노출되어 있던 요소를 위로 접으며 숨김
<b>slideToggle()</b>	숨겨져 있던 요소는 아래로 펼쳐지며 노출되고, 노출되어 있던 요소는 위로 접으면서 숨김
<b>animate()</b>	선택한 요소에 애니메이션을 적용

## ❖ jQuery 효과 및 애니메이션

### ◆ 효과 제어 method 종류

종 류	설 명
<b>stop()</b>	현재 실행 중인 효과를 모두 정지
<b>delay()</b>	스택에 대기 중인 애니메이션 효과를 지연시킴, 대기 중인 효과가 지정한 시간만큼 지연된 후 발생
<b>queue()</b>	선택된 요소의 스택에 대기 중인 큐를 반환하거나 함수에 실행문을 큐로 추가시킬 수 있음
<b>clearQueue()</b>	첫 번째 큐에 해당하는 효과 또는 애니메이션만 실행하고 대기 중인 큐를 모두 삭제
<b>dequeue()</b>	스택에 쌓인 큐를 모두 제거
<b>finish()</b>	선택한 요소의 진행 중인 애니메이션이 강제로 완료 시점으로 간 후 종료