



jQuery 01

❖ 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	http://code.jquery.com/jquery-3.4.1.min.js
	http://code.jquery.com/jquery-latest.js
Google CDN	http://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js
Microsoft CDN	http://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.4.1.min.js

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

❖ jQuery 선택자

◆ 속성 탐색 선택자 종류

종류	사용법	설 명
<code>\$("요소선택[속성]")</code>	<code>\$("li[title]")</code>	 요소 중 title 속성이 포함된 요소만 선택
<code>\$("요소선택[속성=값]")</code>	<code>\$("li[title='리스트']")</code>	 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[속성*=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>	 요소 중 숨겨진 요소만 선택
<code>\$("요소선택:visible")</code>	<code>\$("li:visible")</code>	 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></code> 요소 중 '내용2'라는 텍스트를 포함하는 요소만 선택
<code>\$("요소선택").content()</code>	<code>\$("p").contains()</code>	<code><p></code> 요소의 하위 요소 중 가장 가까운 하위 요소 선택
<code>\$("요소선택:has(요소명)")</code>	<code>\$("li:has('span')")</code>	<code></code> 요소 중 <code></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></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></code> 요소 중 class 속성값이 'list2'인 요소만 선택
<code>\$("요소선택1").find(요소선택2)</code>	<code>\$("li").find("strong")</code>	<code></code> 요소의 하위 요소인 <code></code> 요소만 선택
<code>\$("요소선택1").closest(요소선택2)</code>	<code>\$("strong").closest("div")</code>	<code></code> 을 감싸는 상위 <code><div></code> 요소 중 가장 가까운 상위 요소 선택
<code>end()</code>	<code>\$("li").children("a").end()</code>	필터링이 일어나기 이전의 요소인 <code></code> 선택

❖ jQuery 선택자

◆ 선택자와 함께 사용하면 편리한 method

종류	사용법	설 명
is()	<code>\$(“li”).is(“visible”)</code>	선택한 요소가 보이면 true 반환
noConflict()	<code>var m = \$.noConflict(); m(“h1”).width(100);</code>	선택자 메서드를 사용할 때 '\$'를 앞에 붙여 사용한다. \$.onConflict()를 사용하면 '\$' 메서드를 변수에 참조시켜 '\$' 문자대신 임의의 문자를 지정해서 사용할 수 있다.
get()	<code>\$(“li”).get(0).style.color=“red”;</code>	jQuery로 선택한 요소에 get(0) 메서드를 적용하면 자바스크립트에 DOM 선택자로 선택한 태그로 인식되어 이전 자바스크립트의 DOM 방식을 따른다.
each() / \$.each()	<code>\$(“li”).each(function(매개변수1,매개변수2){}); \$.each(\$(“li”), function(매개변수1,매개변수2){});</code>	선택한 요소들을 각각 순차적으로 접근해 선택해 온다. 매개변수1 : 순차적으로 접근해 선택된 요소의 인덱스 저장 매개변수2 : 순차적으로 선택된 요소가 저장
map()	<code>배열객체.map(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>	선택한 요소의 하위 요소들을 반환하거나 새 요소로 반환
width()	<code>\$("요소선택").width();</code> <code>\$("요소선택").width(100);</code>	선택한 요소에 포함된 텍스트를 반환하거나 새 텍스트로 변환
innerHeight()	<code>\$("요소선택").innerHeight();</code> <code>\$("요소선택").innerHeight(300);</code>	선택한 요소의 스타일(CSS) 속성값을 반환 또는 변환
innerWidth()	<code>\$("요소선택").innerWidth();</code> <code>\$("요소선택").innerWidth(100);</code>	선택한 요소의 속성값을 반환 또는 변환
outerHeight()	<code>\$("요소선택").outerHeight();</code> <code>\$("요소선택").outerHeight(300);</code>	선택한 요소의 지정 속성을 삭제
outerWidth()	<code>\$("요소선택").outerWidth();</code> <code>\$("요소선택").outerWidth(100);</code>	선택한 요소에 지정한 클래스를 생성 또는 추가
position()	<code>\$("요소선택").position().left;</code> <code>\$("요소선택").position().top;</code>	선택한 요소의 지정한 클래스 값을 삭제
offset()	<code>\$("요소선택").toggleClass("abc");</code>	선택한 요소의 지정 클래스 값을 생성, 삭제를 반복
scrollLeft()	<code>\$("요소선택").hasClass("abc");</code>	선택한 요소의 지정 클래스 값의 포함 여부 검사 있으면 true / 없으면 false
scrollTop()	<code>\$("요소선택").prop("tagName");</code>	선택한 태그의 속성들을 반환