

01-1 제이쿼리 기본 다지기

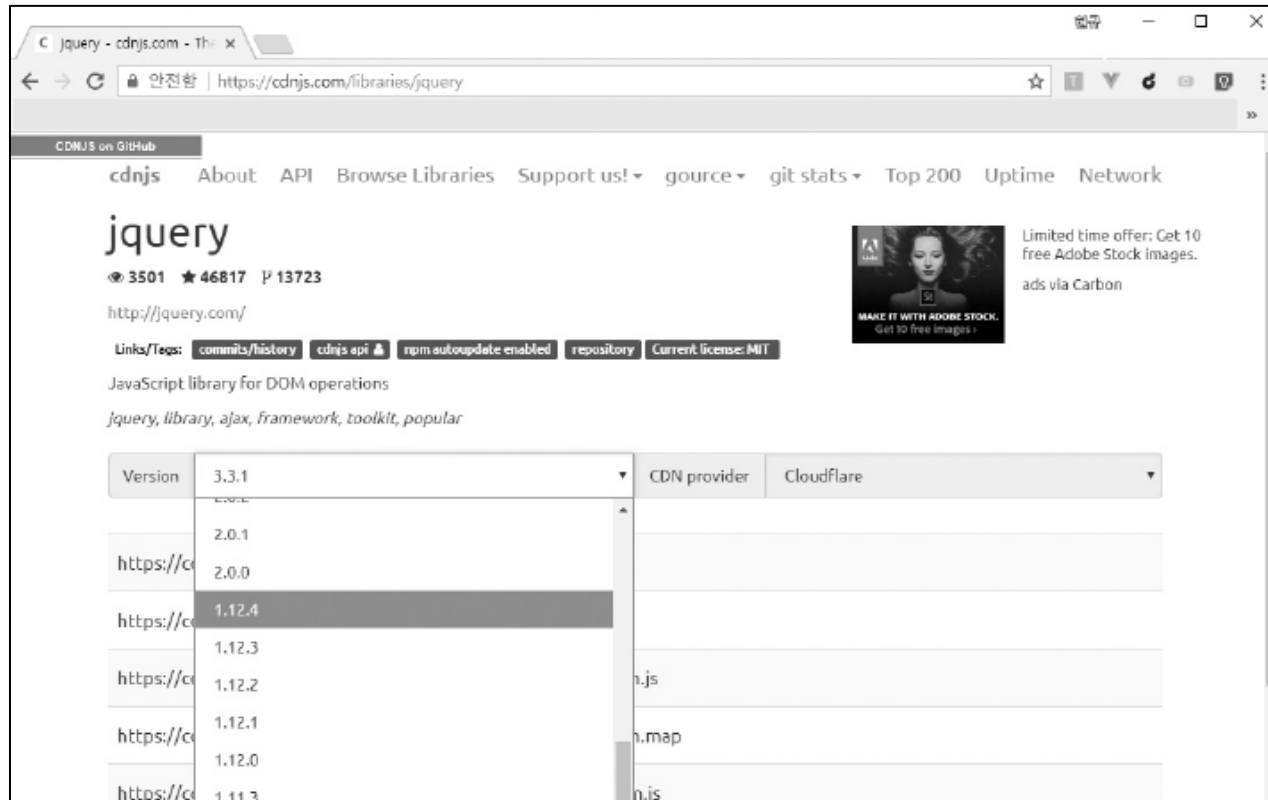
제이쿼리란?



쉽고 편리한 애니메이션 효과 기능 구현

존 레식(John Resig)
자바스크립트를 이용해 만든 라이브러리 언어

제이쿼리 라이브러리 연동하기 — 다운로드 방식



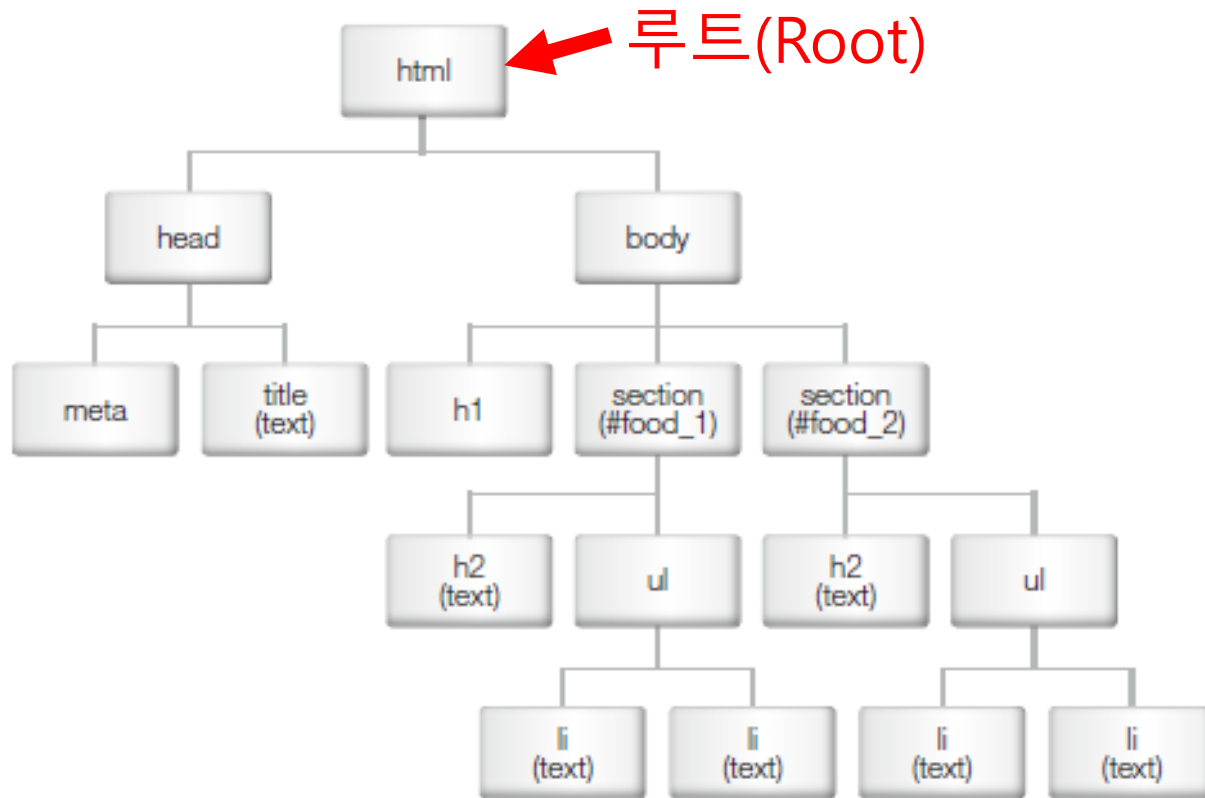
<https://cdnjs.com/libraries/jquery>

제이쿼리 라이브러리 연동하기 — 네트워크 전송 방식

```
<head>  
  <meta charset = "UTF-8">  
  <title> 제이쿼리 연동 </title>  
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/1.12.4/jquery.js"></script>  
</head>
```

01-2 선택자

선택자 — 문서 객체 모델



문서 객체 모델에 대해 알아두는 것이 좋습니다

선택자 — 선택자 사용하기

기본형

1. 선택한 요소에 지정한 스타일을 적용합니다.

```
$("#CSS 선택자").css("스타일 속성명", "값");
```

2. 선택한 요소에 지정한 속성을 적용합니다.

```
$("#CSS 선택자").attr("속성명", "값");
```

기본 선택자 — 직접 선택자

구분	종류	사용법	설명
직접 선택자	전체 선택자	<code>\$("*")</code>	모든 요소를 선택합니다.
	아이디 선택자	<code>\$("#아이디명")</code>	id 속성에 지정한 값을 가진 요소를 선택합니다.
	클래스 선택자	<code>\$(".클래스명")</code>	class 속성에 지정한 값을 가진 요소를 선택합니다.
	요소 선택자	<code>\$("요소명")</code>	지정한 요소명과 일치하는 요소들만 선택합니다.
	그룹 선택자	<code>\$("#선택 1, 선택 2, 선택 3, ...선택 n")</code>	선택 1, 선택 2, 선택 3, ...선택 n에 지정된 요소들을 한 번에 선택합니다.
	종속 선택자	<code>\$("#p.txt_1") \$("#p#txt_1")</code>	<code><p></code> 요소 중 class 값이 txt_1인 요소 또는 id 값이 txt_1인 요소를 선택합니다.

기본 선택자 — 인접 관계 이해하기

문서 객체 구조와 관계

<body>

<div id="wrap"> — a

<div id="inner_wrap"> — b

<h1>제목1</h1> — d

c — <h2>제목2</h2> — e

<p>내용1</p> — f

<p>내용2</p>

<p>내용3</p>

</div>

</div>

</body>

a는 c의 상위(closest) 요소

c는 a의 하위 요소

b는 c의 부모(parent) 요소

c는 b의 자식(children) 요소

d는 e의 형(prev) 요소

f는 e의 동생(next) 요소

기본 선택자 — 인접 관계 선택자

인접 관계 선택자	부모 요소 선택자	<code>\$(요소 선택).parent()</code>	선택한 요소의 부모 요소를 선택합니다.
	상위 요소 선택자	<code>\$(요소 선택).parents()</code>	선택한 요소의 상위 요소를 모두 선택합니다.
	가장 가까운 상위 요소 선택자	<code>\$(요소 선택).closest("div")</code>	선택한 요소의 상위 요소 중 가장 가까운 <div>만 선택합니다.
	하위 요소 선택자	<code>\$(요소 선택 하위 요소)</code>	선택한 요소에 지정한 하위 요소를 선택합니다.
	자식 요소 선택자	<code>\$(요소 선택)자식 요소)</code>	선택한 요소를 기준으로 자식 관계에 지정한 요소 만 선택합니다.
	자식 요소들 선택자	<code>\$(요소 선택).children()</code>	선택한 요소의 모든 자식 요소를 선택합니다.
	형(이전) 요소 선택자	<code>\$(요소 선택).prev()</code>	선택한 요소의 바로 이전 요소를 선택합니다.
	형(이전) 요소들 선택자	<code>\$(요소 선택).prevAll()</code>	선택한 요소의 이전 요소 모두를 선택합니다.
	지정 형(이전) 요소들 선택자	<code>\$(요소 선택).prevUntil ("요소명")</code>	선택한 요소부터 지정한 요소의 이전 요소까지 모 두 선택합니다.
	동생(다음) 요소 선택자	<code>\$(요소 선택).next()</code> <code>\$(요소 선택+다음 요소)</code>	선택한 요소의 다음 요소를 선택합니다.
	동생(다음) 요소들 선택자	<code>\$(요소 선택).nextAll()</code>	선택한 요소의 다음 요소 모두를 선택합니다.
	지정 동생(다음) 요소들 선택자	<code>\$(요소 선택).nextUntil("h2")</code>	선택한 요소부터 지정한 요소의 다음 요소까지 모 두 선택합니다.
	전체 형제 요소 선택자	<code>\$(".box_1").siblings()</code>	class 값이 box_1인 요소의 형제 요소 전체를 선 택합니다.

01-3 제이쿼리 탐색 선택자

위치 탐색 선택자

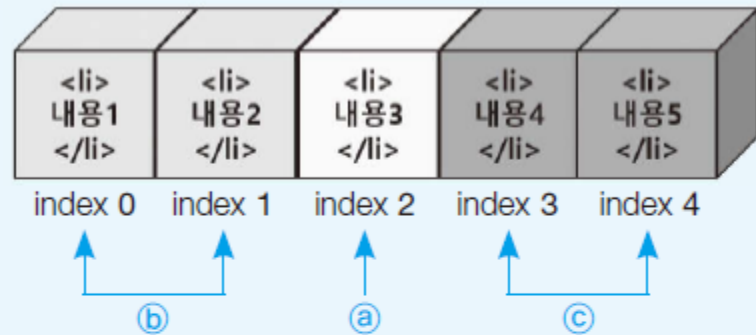
종류	사용법	설명
\$(요소 선택:first) \$(요소 선택).first()	\$(li:first) \$(li).first()	전체 요소 중 첫 번째 요소만 선택합니다.
\$(요소 선택:last) \$(요소 선택).last()	\$(li:last) \$(li).last()	전체 요소 중 마지막 요소만 선택합니다.
\$(요소 선택:odd)	\$(li:odd)	 요소 무리 중 짝수 번째(홀수 인덱스) 요소만 선택합니다.
\$(요소 선택:even)	\$(li:even)	 요소 무리 중 홀수 번째(짝수 인덱스) 요소만 선택합니다.
\$(요소 선택:first-of-type)	\$(li:first-of-type)	 요소 무리 중 첫 번째 요소만 선택합니다.
\$(요소 선택:last-of-type)	\$(li:last-of-type)	 요소 무리 중 마지막 요소만 선택합니다.
\$(요소 선택:nth-child(숫자))	\$(li:nth-child(3))	 요소 무리 중 세 번째 요소만 선택합니다.
\$(요소 선택:nth-child(숫자n))	\$(li:nth-child(3n))	 요소 무리 중 3의 배수 번째에 있는 요소만 선택합니다.
\$(요소 선택:eq(index)) \$(요소 선택).eq(index)	\$(li:eq(2)) \$(li).eq(2)	 요소 중 인덱스 2가 참조하는 요소를 불러옵니다.
\$(요소 선택:gt(index))	\$(li:gt(1))	 요소 중 인덱스 1보다 큰 인덱스가 참조하는 요소를 불러옵니다.
\$(요소 선택:lt(index))	\$(li:lt(1))	 요소 중 인덱스 1보다 작은 인덱스가 참조하는 요소를 불러옵니다.

위치 탐색 선택자(4) — eq(index) / lt(index) / gt(index) 탐색 선택자

```
<script>
$(function() {
  $("#menu li").eq(2) — ㉠
    .css({"background-color": "#ff0"});

  $("#menu li:lt(2)") — ㉡
    .css({"background-color": "#0ff"});

  $("#menu li:gt(2)") — ㉢
    .css({"background-color": "#f0f"});
});
</script>
```



제이쿼리 배열 관련 메서드

종류	사용법	설명
<code>each() / \$.each()</code>	<code>\$("요소 선택").each(function)</code> <code>\$.each(\$ "요소 선택", function)</code>	배열에 저장된 문서 객체만큼 메서드가 반복 실행됩니다. 배열에 저장된 객체의 인덱스 순서대로 하나씩 접근하여 객체를 선택하고 인덱스를 구합니다.
<code>\$.map()</code>	<code>\$.map(Array, function)</code>	배열에 저장된 데이터 수만큼 메서드가 반복 실행됩니다. 함수에서 반환된 데이터는 새 배열에 순서대로 저장됩니다. 새로 저장된 배열 객체를 반환합니다.
<code>\$.grep()</code>	<code>\$.grep(Array, function)</code>	배열에 저장된 데이터 수만큼 메서드가 반복 실행됩니다. 반환값이 true인 경우에만 배열의 데이터가 인덱스 오름차순으로 새 배열에 저장되며 그 배열을 반환합니다.
<code>\$.inArray()</code>	<code>\$.inArray(data, Array, start index)</code>	배열 안에서 데이터를 찾습니다. 데이터를 찾으면 가장 맨 앞 데이터의 인덱스를 반환하고, 찾지 못하면 -1을 반환합니다. start index의 값을 지정하면 해당 위치부터 데이터를 찾습니다.
<code>\$.isArray()</code>	<code>\$.isArray(object)</code>	입력한 객체가 배열 객체라면 true를, 아니면 false를 반환합니다.
<code>\$.merge()</code>	<code>\$.merge(Array1, Array2)</code>	인자값으로 입력한 2개의 배열 객체를 하나로 그룹화합니다.
<code>index()</code>	<code>\$("요소 선택").index("지정 요소 선택")</code>	선택자로 요소를 먼저 선택합니다. 그런 다음 지정한 요소의 인덱스 정보를 가져옵니다.

속성 탐색 선택자의 종류

종류	사용법	설명
<code>\$(요소 선택[속성])</code>	<code>\$(li[title])</code>	〈li〉 요소 중 title 속성이 포함된 요소만 선택합니다.
<code>\$(요소 선택[속성=값])</code>	<code>\$(li[title='리스트'])</code>	〈li〉 요소 중 title 속성값이 '리스트'인 요소만 선택합니다.
<code>\$(요소 선택[속성^=텍스트])</code>	<code>\$(a[href^='http://'])</code>	〈li〉 요소 중 href 속성값이 'http://'로 시작하는 요소만 선택합니다.
<code>\$(요소 선택[속성\$=텍스트])</code>	<code>\$(a[href\$='.com'])</code>	〈li〉 요소 중 href 속성값이 '.com'으로 끝나는 요소만 선택합니다.
<code>\$(요소 선택[href*=텍스트])</code>	<code>\$(a[href*='easyspub'])</code>	〈li〉 요소 중 href 속성값 중에서 'easyspub'을 포함하는 요소만 선택합니다.
<code>\$(요소 선택:hidden)</code>	<code>\$(li:hidden)</code>	〈li〉 요소 중 숨겨져 있는 요소만 선택합니다.
<code>\$(요소 선택:visible)</code>	<code>\$(li:visible)</code>	〈li〉 요소 중 보이는 요소만 선택합니다.
<code>\$(":text")</code>	<code>\$(":text")</code>	〈input〉 요소 중 type 속성값이 "text"인 요소만 선택합니다.
<code>\$(":selected")</code>	<code>\$(":selected")</code>	selected 속성이 적용된 요소만 선택합니다.
<code>\$(":checked")</code>	<code>\$(":checked")</code>	checked 속성이 적용된 요소만 선택합니다.

그 외의 선택자 — 콘텐츠 탐색 선택자

종류	사용법	설명
<code>\$(요소 선택:contains(텍스트))</code>	<code>\$(li:contains('내용2'))</code>	⟨li⟩ 요소 중 '내용2'라는 텍스트를 포함하는 요소만 선택합니다.
<code>\$(요소 선택).contents()</code>	<code>\$(p).contents()</code>	선택한 요소의 하위 요소 중 가장 가까운 하위 요소를 선택합니다.
<code>\$(요소 선택:has(요소명))</code> <code>\$(요소 선택).has(요소명)</code>	<code>\$(li:has('span'))</code> <code>\$(li).has('span')</code>	⟨li⟩ 요소 중 ⟨span⟩을 포함하는 요소만 선택합니다.
<code>\$(요소 선택:not(:제외 요소))</code> <code>\$(요소 선택).not(:제외 요소)</code>	<code>\$(li:not(:first))</code> <code>\$(li)</code>	⟨li⟩ 요소 중 첫 번째 요소만 제외하고 선택합니다.
<code>\$(요소 선택).filter(필터 요소)</code>	<code>\$(li).filter('.list2')</code>	⟨li⟩ 요소 중 class 값이 'list2'인 요소만 선택합니다.
<code>\$(요소 선택1).find(요소 선택2)</code>	<code>\$(li).find('strong')</code>	⟨li⟩ 요소의 하위 요소인 ⟨strong⟩만 선택합니다.
<code>\$(요소 선택1).closest(요소 선택2)</code>	<code>\$(strong).closest('div')</code>	⟨strong⟩을 감싸는 상위 ⟨div⟩ 요소 중 가장 가까운 상위 요소를 선택합니다.

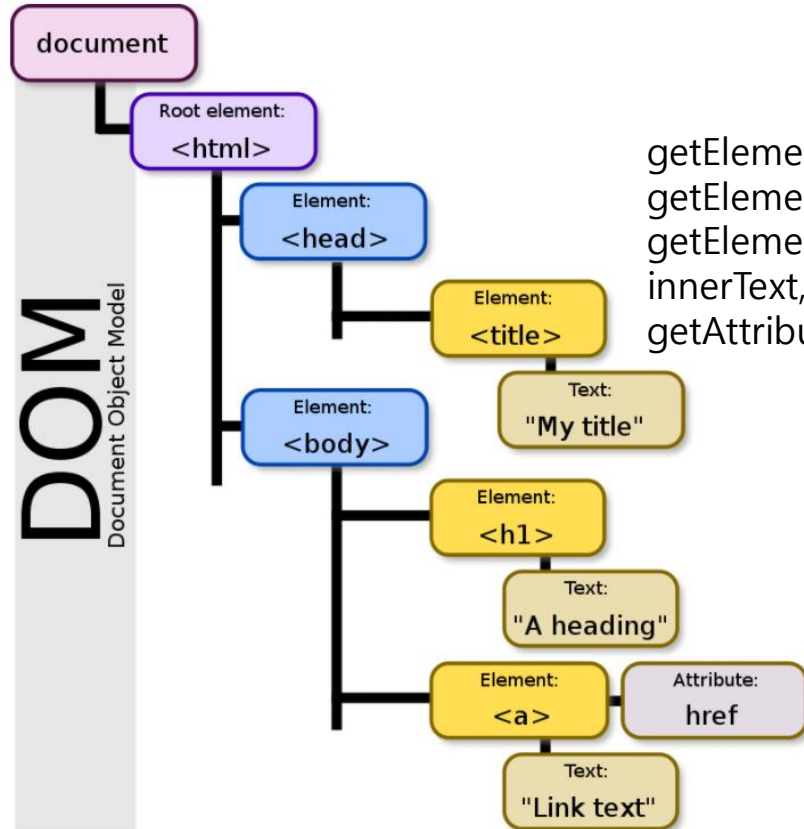
종류	사용법	설명
is("요소 상태")	\$(".txt1").is(":visible")	선택한 요소가 보이면 true를 반환합니다.

기본형 : \$(".요소 선택").is(":"^①checked | ^②selected | ^③visible | ^④hidden | ^⑤animated"]")

- ① 선택한 체크 박스(checkbox) 또는 라디오(radio) 버튼 요소가 체크된(checked) 상태면 true를, 아니면 false를 반환합니다.
- ② 선택한 선택 상자(select)에 옵션(option) 요소가 선택된(selected) 상태면 true를, 아니면 false를 반환합니다.
- ③ 선택한 요소가 보이면(visible) true를, 아니면 false를 반환합니다.
- ④ 선택한 요소가 숨겨진(hidden) 상태면 true를, 아니면 false를 반환합니다.
- ⑤ 선택한 요소가 애니메이션 동작(animated) 상태면 true를, 아니면 false를 반환합니다.

노드 : DOM트리에서 가지가 갈라져 나간 항목
루트 노드 : DOM트리의 시작 부분인 html 노드
부모 노드 안의 요소: 자식 노드
부모 노드가 동일 : 형제 노드

getElementById() 메서드=> id선택자로 DOM에 접근
getElementsByClassName() 메서드=> class값으로 DOM에 접근
getElementsByTagName() 메서드=> 태그 이름으로 DOM 요소에 접근하기
innerText, innerHTML 프로퍼티=> 웹 요소 내용 수정
getAttribute(), setAttribute() 메서드=> 속성을 가져오거나 수정하는 메서드



01-4 객체 조작 메서드

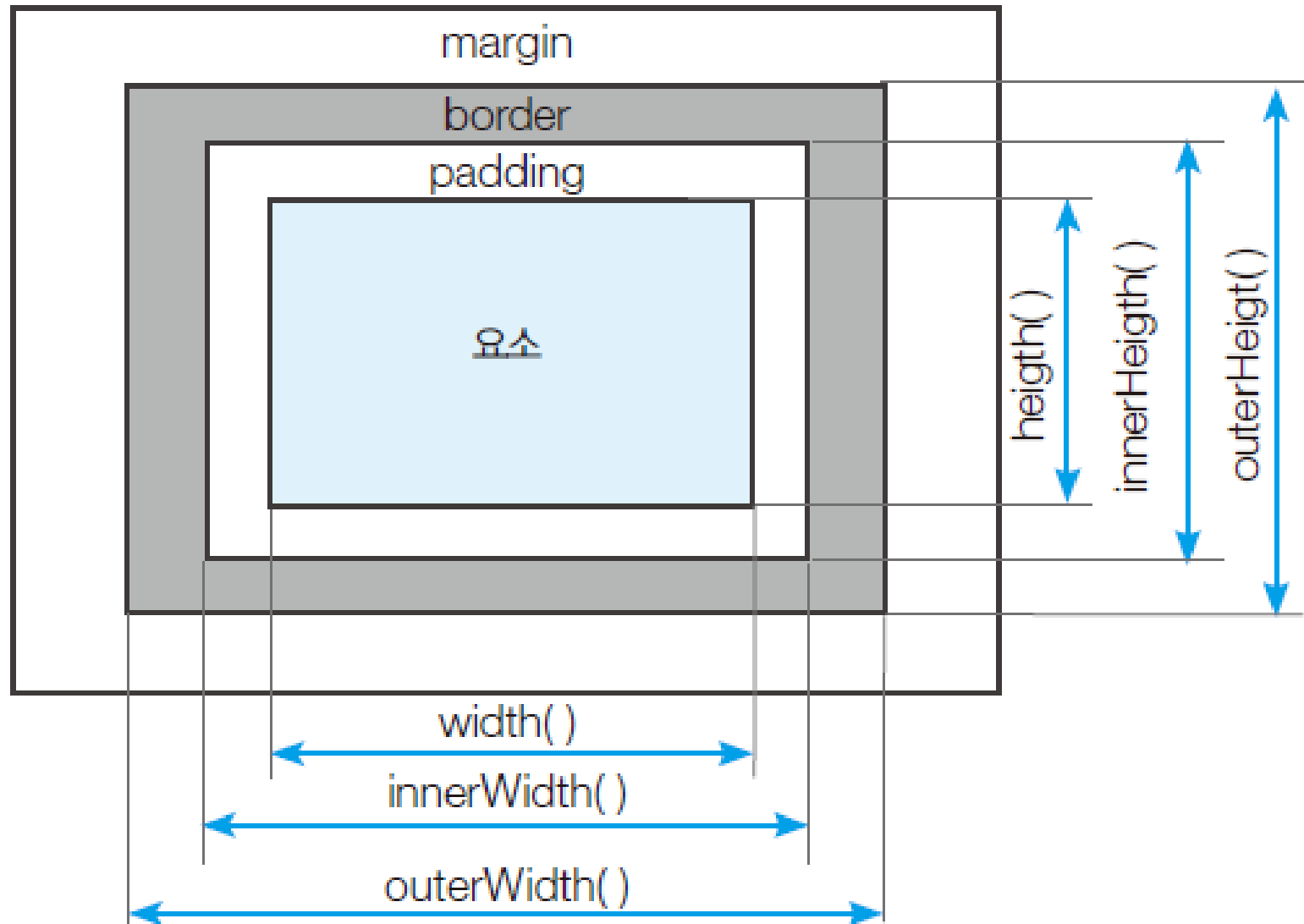
종류	사용법	설명
html() html("새 요소")	\$(요소 선택).html(); \$(요소 선택).html("새 요소");	선택한 요소의 하위 요소를 가져옵니다. 선택한 요소의 하위 요소를 모두 제거하고, 그 위치에 지정한 새 요소를 생성합니다.
text() text("새 텍스트")	\$(요소 선택).text(); \$(요소 선택).text("새 텍스트");	선택한 요소가 감싸는 모든 텍스트를 가져옵니다. 선택한 요소의 하위 요소를 모두 제거하고, 그 위치에 지정한 새 텍스트를 생성합니다.
attr("속성명") attr("속성명","새 값")	\$(요소 선택).attr("속성명"); \$(요소 선택).attr("속성명","새 값");	선택한 요소의 지정한 속성(attribute)값을 가져옵니다. 요소를 선택하여 지정한 속성에 새 값을 적용합니다.
removeAttr("속성명")	\$(요소 선택).removeAttr("속성명");	선택한 요소의 지정한 속성만 제거합니다.

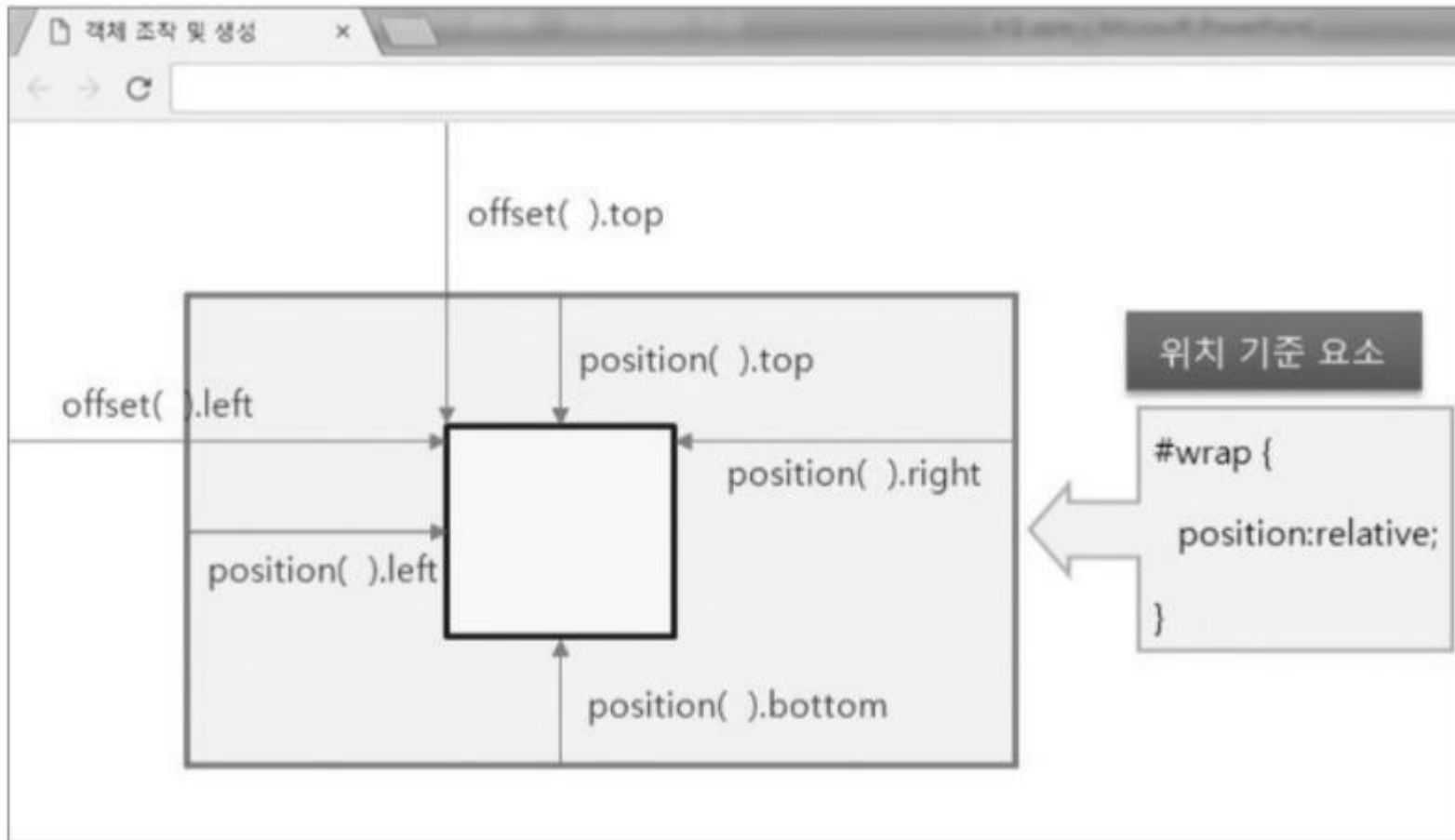
종류	사용법	설명
prop("상태 속성명") prop("상태 속성명", 새 값)	\$("#요소 선택").prop("상태 속성명"); \$("#요소 선택").prop("상태 속성명", "새 값");	선택한 요소의 상태 속성값을 가져옵니다. 요소를 선택하여 상태 속성에 새 값을 적용합니다.
val() val("새 값")	\$("#요소 선택").val(); \$("#요소 선택").val("새 값");	선택한 폼 요소의 value 값을 가져옵니다. 폼 요소를 선택하여 value 속성에 새 값을 적용합니다.
css("속성명") css("속성명", "새 값")	\$("#요소 선택").css("속성명"); \$("#요소 선택").css("속성명", "새 값");	선택한 요소의 지정한 스타일(CSS) 속성값을 가져옵니다. 요소를 선택하여 지정한 스타일(CSS) 속성에 새 값을 적용합니다.
addClass("class 값")	\$("#요소 선택").addClass("class 값");	선택한 요소의 class 속성에 새 값을 추가합니다.
removeClass("class 값")	\$("#요소 선택").removeClass("class 값");	선택한 요소의 class 속성에서 지정한 값만 제거합니다.
toggleClass("class 값")	\$("#요소 선택").toggleClass("class 값")	선택한 요소의 class 값에 지정한 값이 포함되어 있으면 제거하고 속성값이 없으면 추가합니다.
hasClass("class 값")	\$("#요소 선택").hasClass("class 값")	선택한 요소의 class 값에 지정한 클래스 값이 포함되어 있으면 true를, 없으면 false를 반환합니다.

수치 조작 메서드

종류	사용법	설명
height()	\$(요소 선택).height(); \$(요소 선택).height(100);	안쪽 여백과 선을 제외한 높이를 반환하거나 변환합니다.
width()	\$(요소 선택).width(); \$(요소 선택).width(100);	안쪽 여백과 선을 제외한 너비를 반환하거나 변환합니다.
innerHeight()	\$(요소 선택).innerHeight(); \$(요소 선택).innerHeight(300);	안쪽 여백을 포함한 높이를 반환하거나 변환합니다.
innerWidth()	\$(요소 선택).innerWidth(); \$(요소 선택).innerWidth(100);	안쪽 여백을 포함한 너비를 반환하거나 변환합니다.
outerHeight()	\$(요소 선택).outerHeight(); \$(요소 선택).outerHeight(100);	선과 안쪽 여백을 포함한 높이를 반환하거나 변환합니다.
outerWidth()	\$(요소 선택).outerWidth(); \$(요소 선택).outerWidth(100);	선과 안쪽 여백을 포함한 너비를 반환하거나 변환합니다.
position()	\$(요소 선택).position().left; \$(요소 선택).position().top;	선택한 요소의 포지션 위치값을 반환합니다.
offset()	\$(요소 선택).offset().left; \$(요소 선택).offset().top;	선택한 요소가 문서에서 수평/수직으로 얼마나 떨어져 있는지에 대한 값을 반환합니다.
scrollLeft()	\$(window).scrollLeft();	브라우저의 수평 스크롤 이동 높이를 반환합니다.
scrollTop()	\$(window).scrollTop();	브라우저의 수직 스크롤 이동 너비를 반환합니다.

수치 조작 메서드 — 요소 너비 / 높이 메서드의 계산 범위





객체 편집 메서드(1)

종류	사용법	설명
before()	<code>\$("요소 선택").before("새 요소");</code>	선택한 요소의 이전 위치에 새 요소를 추가합니다.
after()	<code>\$("요소 선택").after("새 요소");</code>	선택한 요소의 다음 위치에 새 요소를 추가합니다.
append()	<code>\$("요소 선택").append("새 요소");</code>	선택한 요소의 마지막 위치에 새 요소를 추가합니다.
appendTo()	<code>\$("새 요소").appendTo("요소 선택");</code>	선택한 요소의 마지막 위치에 새 요소를 추가합니다.
prepend()	<code>\$("요소 선택").prepend("새 요소");</code>	선택한 요소의 맨 앞 위치에 새 요소를 추가합니다.

객체 편집 메서드(2)

prependTo()	\$("새 요소").prependTo("요소 선택");	선택한 요소의 맨 앞 위치에 새 요소를 추가합니다.
insertBefore()	\$("새 요소").insertBefore("요소 선택");	선택한 요소의 이전 위치에 새 요소를 추가합니다.
insertAfter()	\$("새 요소").insertAfter("요소 선택");	선택한 요소의 다음 위치에 새 요소를 추가합니다.
clone()	\$("요소 선택").clone(true or false);	선택한 문서 객체를 복제합니다. 이때 인자값이 true일 경우 하위 요소까지 모두 복제하고, false일 경우에는 선택한 요소만 복제합니다.
empty()	\$("요소 선택").empty();	선택한 요소의 하위 내용들을 모두 삭제합니다.
remove()	\$("요소 선택").remove();	선택한 요소를 삭제합니다.
replaceAll() / replaceWith()	\$("새 요소").replaceAll("요소 선택"); \$("요소 선택").replaceWith("새 요소");	선택한 요소들을 새 요소로 교체합니다.
unwrap()	\$("요소 선택").unwrap();	선택한 요소의 부모 요소를 삭제합니다.
wrap()	\$("요소 선택").wrap(새 요소);	선택한 요소를 새 요소로 각각 감쌉니다.
wrapAll()	\$("요소 선택").wrapAll();	선택한 요소를 새 요소로 한꺼번에 감쌉니다.
wrapInner()	\$("요소 선택").wrapInner(새 요소);	선택한 요소의 내용을 새 요소로 각각 감쌉니다.