**선택자**

**기본선택자**

|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 종류 | 사용법 | 설명 |
| 직접 선택자 | 전체선택자 | $(“\*”) | 모든 요소를 선택 |
| 아이디 선택자 | $(“#아이디명”) | id 속성에 지정한 값을 가진 요소를 선택 |
| 클래스 선택자 | $(“.클래스명”) | class 속성에 지정한 값을 가진 요소를 선택 |
| 요소 선택자 | $(“요소명”) | 지정한 요소명과 일치하는 요소들만 선택 |
| 그룹 선택자 | $(“선택1, 선택2, 선택3”) | 선택1, 선택2, 선택3 ... 선택 n에 지정된 요소들을 한꺼번에 선택 |
| 종속 선택자 | $(“p.txt\_1”) $(“p#txt\_1“) | <p> 요소 중 class 값이 txt\_1인 요소 또는 id 값이 txt\_1인 요소를 선택 |
| 인접  관계 선택자 | 부모요소 선택자 | $(“요소 선택”).parent() | 선택한 요소의 부모요소를 선택 |
| 상위요소 선택자 | $(“요소 선택”).parents() | 선택한 요소의 상위 요소를 모두 선택 |
| 가장 가까운 상위 요소 선택자 | $(“요소 선택”).closest(“div”) | 선택한 요소의 상위 요소 중 가장 가까운 <div>만 선택 |
| 하위요소 선택자 | $(“요소 선택 하위요소”) | 선택한 요소에 지정한 하위 요소를 선택 |
| 자식요소 선택자 | $(“요소 선택 > 자식요소”) | 선택한 요소를 기준으로 자식 관계에 지정한 요소만 선택 |
| 자식요소들 선택자 | $(“요소 선택”).children() | 선택한 요소의 모든 자식 요소를 선택 |
| 형(이전) 요소 선택자 | $(“요소 선택”).prev() | 선택한 요소의 바로 이전 요소를 선택 |
| 형(이전) 요소들 선택자 | $(“요소 선택”).prevAll() | 선택한 요소의 바로 이전 요소 모두를 선택 |
| 지정 형(이전) 요소들 선택자 | $(“요소 선택”).prevUntil(“요소명”) | 선택한 요소부터 지정한 요소의 이전 요소까지 모두 선택 |
| 동생(다음)요소 선택자 | $(“요소 선택”).next()  $(“요소 선택+다음요소”) | 선택한 요소의 다음 요소를 선택 |
| 동생(다음)요소들 선택자 | $(“요소 선택”).nextAll() | 선택한 요소의 다음 요소 모두를 선택 |
| 지정동생(다음) 요소들 선택자 | $(“요소 선택”).nextUntil(“h2”) | 선택한 요소부터 지정한 요소의 다음 요소까지 모두 선택 |
| 전체 형제 요소 선택자 | $(“.box\_1”).siblings() | class 값이 box\_1인 요소의 형제 요소 전체를 선택 |

**위치 탐색 선택자**

|  |  |  |
| --- | --- | --- |
| 종류 | 사용법 | 설명 |
| $(“요소선택:first”) $(“요소선택”).fist() | $(“li:first”) $(“li“).first() | 전체 <li> 요소 중 첫 번째 요소만 선택합니다. |
| $(“요소선택:last”) $(“요소선택”).last() | $(“li:last”)  $(“li”).last() | 전체 <li> 요소 중 마지막 요소만을 선택합니다. |
| $(“요소선택:odd”) | $(“li:odd”) | <li>요소 무리 중 짝수 번째(홀수 인덱스)요소만 선택 |
| $(“요소선택:even”) | $(“li:even”) | <li>요소 무리 중 홀수 번째(짝수 인덱스)요소만 선택 |
| $(“요소선택:first-of-type”) | $(“li:first-of-type”) | <li>요소 무리 중 첫 번째 요소만 선택 |
| $(“요소선택:last-of-type”) | $(“li:last-of-type”) | <li>요소 무리 중 마지막 요소만 선택 |
| $(“요소선택:nth-child(숫자)”) | $(“li:nth-child(3)”) | <li>요소 무리 중 세번째 요소만 선택 |
| $(“요소선택:nth-child(숫자n)”) | $(“li:nth-child(3n)) | <li>요소 무리 중 3의 배수 번째 있는 요소만 선택 |
| $(“요소선택:nth-last-of-type(숫자)”) | $(“li:nth-last-of-type(2)”) | <li>요소 무리 중 마지막 위치로부터 두 번째 있는 요소만 선택 |
| $(“요소선택:only-child”) | $(“li:only-child”) | 부모 요소 내에 <li> 요소가 1개 뿐인 <li>요소만 선택 |
| $(“요소선택:eq(index)”)  $(“요소선택”).eq(index) | $(“li:eq(2)”)  $(“li”).eq(2) | <li>요소 중 인덱스 2가 참조하는 요소를 불러옵니다 |
| $(“요소선택:gt(index)”) | $(“li:gt(1)”) | <li>요소 중 인덱스 1보다 큰 인덱스가 참조하는 요소를 불러옵니다. |
| $(“요소선택:lt(index)”) | $(“li:lt(1)”) | <li>요소 중 인덱스 1보다 작은 인덱스가 참조하는 요소를 불러 옵니다. |
| $(“요소선택”).slice(index) | $(“li”).slice(2) | <li>요소 중 인덱스 2부터 참조하는 요소를 불러옵니다. |

**속성 탐색 선택자**

|  |  |  |
| --- | --- | --- |
| 종류 | 사용법 | 설명 |
| $(“요소선택[속성]”) | $(“li[title]“) | <li>요소 중 title 속성이 포함된 요소만 선택 |
| $(“요소선택[속성=‘값’]”) | $(“li[title=’리스트‘]”) | <li>요소 중 title 속성 값이 ‘리스트’인 요소만 선택 |
| $(“요소선택[속성^=‘텍스트’]”) | $(“a[href^=’http://‘]”) | <li>요소 중 href 속성값이 ‘http://’로 시작하는 요소만 선택 |
| $(“요소선택[속성$=‘텍스트’]”) | $(“a[href$=’.com’]”) | <li>요소 중 href 속성값이 ‘.com’으로 끝나는 요소만 선택 |
| $(“요소선택[속성\*=‘텍스트’]”) | $(“a[href\*=’easypub’]”) | <li>요소 중 href 속성값 중에서 ‘easyspuyb’을 포함하는 요소만 선택 |
| $(“요소선택:hidden”) | $(“li:hidden”) | <li>요소 중 숨겨져 있는 요소만 선택 |
| $(“요소선택:visible”) | $(“li:visible”) | <li>요소 중 보이는 요소만 선택 |
| $(“:text”) | $(“:text”) | <input> 요소 중 type 속성값이 “text”인 요소만 선택 |
| $(“:selected”) | $(“:selected”) | selected 속성이 적용된 요소만 선택 |
| $(“:checked”) | $(“:checked”) | checked 속성이 적용된 요소만 선택 |

**컨텐츠 탐색 선택자**

|  |  |  |
| --- | --- | --- |
| 종류 | 사용법 | 설명 |
| $(“요소선택:contains(‘텍스트’)”) | $(“li:contains(‘내용2’)”) | <li> 요소 중 ‘내용2’라는 텍스트를 포함하는 요소만 선택 |
| $(“요소선택”).contents() | $(“p”).contents() | 선택한 요소의 하위 요소 중 가장 가까운 하위 요소를 선택 |
| $(“요소선택:has(‘요소명’)”)  $(“요소선택”).has(“요소명”) | $(“li:has(‘span’)”)  $(“li”).has(“span”) | <li>요소 중 <span>을 포함하는 요소만 선택 |
| $(“요소선택:not(‘제외요소’)”)  $(“요소선택”).not(“제외요소”) | $(“li:not(:first)”) | <li>요소 중 첫 번째 요소만 제외하고 선택 |
| $(“요소선택”).filter(“필터요소”) | $(“li”).filter(“.list2”) | <li>요소 중 class 값이 “list2”인 요소만 선택 |
| $(“요소선택1”).find(“요소선택2”) | $(“li”).find(“strong”) | <li>요소 중 하위 요소인 <strong>만 선택합니다. |
| $(“요소선택1”).closest(“요소선택2”) | $(“strong”).closest(“div”) | <strong>을 감싸는 <div> 요소 중 가장 까까운 상위 요소를 선택합니다. |
| end() | $(“li”).children(“a”).end() | 필터링이 실행되기 이전의 요소인 <li>가 선택됩니다. |

**메서드**

**제이쿼리 배열관련 메서드**

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

**선택자와 함께 알아두면 유용한 메서드**

|  |  |  |
| --- | --- | --- |
| 종류 | 사용법 | 설명 |
| is(“요소 상태”) | $(“.txt1”).is(:visible“) | 선택한 요소가 보이면 true를 반환 |
| $.noConflict() | var 변수 = $.noConflict()  변수(“요소 선택”) | $.noConflict() 함수를 이용하면 현재 제이쿼리에서 사용 중인 $ 메서드 사용을 중단하고 새로 지정한 변수명 메서드를 사용 |
| get() | $(“요소 선택”).get(0)  var el = $(“요소선택).get(0);  el.style.color = “#f00”; | 선택자에 get(0)을 적용하면 자바스크립트 DOM 방식의 스타일을 적용할 수 있다  **(제이쿼리로 선택한 요소는 자바스크립트 DOM 방식의 스타일을 사용할 수 없습니다.)** |

**속성 조작 메서드**

|  |  |  |
| --- | --- | --- |
| 종류 | 사용법 | 설명 |
| html()  html(“새요소”) | $(“요소선택”).html();  $(“요소선택”).html(“새 요소”) | 선택한 요소의 하위 요소를 가져옵니다.  선택한 하위 요소를 모두 제거하고, 그 위치에 지정한 새 요소를 생성합니다. |
| text()  text(“새 텍스트”) | $(“요소선택”).text();  $(“요소선택”).text(“새 텍스트”); | 선택한 요소가 감싸는 모든 텍스트를 가져옵니다.  선택한 요소의 하위요소를 모두 제거하고 그 위치에 지정한 새 텍스트를 생성합니다. |
| attr(“속성명”)  attr(“속성명”, “새 값”) | $(“요소선택”).attr(“속성명”);  $(“요소선택”).attr(“속성명”, “새 값”); | 선택한 요소의 지정한 속성(attribute)값을 가져옵니다.  요소를 선택하여 지정한 속성에 새 값을 적용합니다. |
| removeAttr(“속성명”) | $(“요소선택”).removeAttr(“속성명”); | 선택한 요소의 지정한 속성만 제거합니다. |
| prop(“상태 속성명”)  prop(“상태 속성명”, 새값) | $(“요소선택”).prop(“상태속성명”);  $(“요소선택”).prop(“상태속성명”,“새 값”); | 선택한 요소의 상태 속성값을 가져옵니다.  폼 요소를 선택하여 value 속성에 새 값을 적용 |
| val()  val(“새 값”) | $(“요소선택”).val();  $(“요소선택”).val(“새 값”); | 선택한 폼 요소의 value 값을 가져옵니다.  폼 요소를 선택하여 value 속성에 새 값을 적용 |
| css(“속성명”)  css(“속성명”, “새 값”) | $(“요소선택”).css(“속성명”);  $(“요소선택”).css(“속성명”, “새 값”); | 선택한 요소의 지정한 스타일(CSS) 속성값을 가져옵니다.  요소를 선택하여 지정한 스타일(CSS) 속성에 새 값을 적용 |
| addClass(“class 값”) | $(“요소선택”).addClass(“class값”); | 선택한 요소의 class 속성에 새 값을 추가합니다. |
| removeClass(“class 값”) | $(“요소선택”).removeClass(“class값”); | 선택한 요소의 class 속성에 지정한 값만 제거합니다. |
| toggleClass(“class 값”) | $(“요소선택”).toggleClass(“class값”); | 선택한 요소에 class값에 지정한 값이 포함되어 있으면 제거하고 속성값이 없으면 추가합니다. |
| hasClass(“class 값”) | $(“요소선택”).hasClass(“class 값”); | 선택한 요소의 class 값에 지정한 클래스 값이 포함되어 있으면 true, 없으면 false를 반환 |

**수치 조작 메서드**

|  |  |  |
| --- | --- | --- |
| 종류 | 사용법 | 설명 |
| height() | $(“요소선택”).height();  $(“요소선택”).height(100); | 안쪽 여백과 선을 제외한 높이값을 반환하거나 변환합니다. |
| width() | $(“요소선택”).width();  $(“요소선택”).width(100) | 안쪽 여백과 선을 제외한 너빗값을 반환하거나 변환합니다. |
| innerHeight() | $(”요소선택”).innerHeight();  $(”요소선택”).innerHeight(300); | 안쪽 여백을 포함한 높이값을 반환하거나 변환합니다. |
| innerWidth() | $(”요소선택”).innerWidth();  $(”요소선택”).innerWidth(100); | 안쪽 여백을 포함한 너비값을 반환하거나 변환합니다. |
| outerWidth() | $(”요소선택”).outerWidth();  $(”요소선택”).innerWidth(100); | 선과 안쪽 여백을 포함한 너비값을 반환하거나 변환합니다. |
| outerHeight() | $(”요소선택”).outerHeight();  $(”요소선택”).outerHeight(100); | 선과 안쪽 여백을 포함한 높이값을 반환하거나 변환합니다. |
| position() | $(”요소선택”).position().left;  $(”요소선택”).position().top; | 선택한 요소의 포지션 위칫값을 반환합니다. |
| offset() | $(”요소선택”).offset().left;  $(”요소선태”).offset().top; | 선택한 요소가 문서에서 수평/수직으로 얼마나 떨어져 있는지에 대한 값을 반환합니다. |
| scrollLeft() | $(”요소선택”).scrollLeft(); | 브라우저의 수직 스크롤 이동 너빗값을 반환합니다. |
| scrollTop() | $(”요소선택”).scrollTop(); | 브라우저의 수평 스크롤 이동 높이값을 반환합니다. |

**객체 편집 메서드**

|  |  |  |
| --- | --- | --- |
| 종류 | 사용법 | 설명 |
| before() | $(”요소선택”).before(”새 요소”); | 선택한 요소의 이전 위치에 새 요소를 추가합니다. |
| after() | $(”요소선택”).after(”새 요소”); | 선택한 요소의 다음 위치에 새 요소를 추가합니다. |
| append() | $(”요소선택”).append(”새요소”); | 선택한 요소의 마지막 위치에 새 요소를 추가합니다. |
| appendTo() | $(”새요소”).appendTo(”요소선택”); | 선택한 요소의 마지막 위치에 새 요소를 추가합니다. |
| prepend() | $(”요소선택”).prepend(”새 요소”); | 선택한 요소의 맨 앞 위치에 새 요소를 추가합니다. |
| prependTo() | $(”새요소”).prependTo(”요소선택”) | 선택한 요소의 맨 앞 위치에 새 요소를 추가합니다. |
| insertBefore() | $(”새요소”).insertBefore(”요소선택”) | 선택한 요소의 이전 위치에 새 요소를 추가합니다. |
| insertAfter() | $(”새요소”).insertAfter(”요소선택”); | 선택한 요소의 다음 위치에 새 요소를 추가합니다. |
| clone() | $(”요소선택”).clone(true or false); | 선택한 문서 객체를 복사합니다. 이때 인자값이 true일 경우 하위 요소까지 모두 복제하고, false일 경우에는 선택한 요소만 복제합니다. |
| empty() | $(”요소선택”).empty(); | 선택한 요소의 하위내용들을 모두 삭제합니다. |
| remove() | $(”요소선택”).remove(); | 선택한 요소를 삭제합니다. |
| replaceAll()  replaceWith() | $(”새요소”).replaceAll(”요소 선택”);  $(”요소선택”).replaceWith(”새요소”); | 선택한 요소들을 새 요소로 교체합니다, |
| unwrap() | $(”요소선택”).unwrap(); | 선택한 요소의 부모요소를 삭제합니다. |
| wrap() | $(”요소선택”).wrap(”새요소”); | 선택한 요소를 새 요소로 각각 감쌉니다. |
| wrapAll() | $(”요소선택”).wrapAll(새 요소); | 선택한 요소를 새 요소로 한꺼번에 감쌉니다. |
| wrapInner() | $(”요소선택”).wrapInner(”새요소”); | 선택한 요소의 내용을 새 요소르 각각 감쌉니다. |

**이벤트 등록 메서드**

|  |  |  |
| --- | --- | --- |
| 구분 | 종류 | 설명 |
| 로딩 이벤트 | load() | 선택한 이미지 또는 프레임 요소에 연동된 소스의 로딩이 완료된 후 이벤트가 발생합니다. |
| ready() | 지정된 HTML 문서 객체의 로딩이 완료된 후 이벤트가 발생합니다. |
| error() | 이벤트 대상 요소에서 오류가 발생하면 이벤트가 발생합니다. |
| 마우스  이벤트 | click() | 선택한 요소를 클릭했을 때 이벤트가 발생합니다. |
| dblclick() | 선택한 요소를 연속해서 두번 클릭했을 때 이벤트가 발생합니다. |
| mouseout() | 선택한 요소의 영역에서 마우스 포인트가 벗어났을 때 이벤트가 밸생합니다. 이때 하위 요소의 영향을 받습니다. |
| mouseover() | 선택한 요소의 영역에 마우스 포인터가 벗어 났을 때 이벤트가 발생합니다. 이때 하위 요소의 영향을 받습니다. |
| hover() | 선택한 요소에 마우스 포인터를 올렸을 때와 벗어났을 때 각각 이벤트가 발생합니다. |
| mousedown() | 선택한 요소에서 마우스 버튼을 눌렀을 때 이벤트가 발생합니다. |
| mouseup() | 선택한 요소에서 마우스 버튼을 눌렀다 떼었을 때 이벤트가 발생합니다. |
| mouseenter() | 선택한 요소 범위에 마우스 포인터를 올렸을 때 이벤트가 발생합니다. |
| mouseleave() | 선택한 요소 범위에서 마우스 포인터가 벗어났을 때 이벤트가 발생합니다. |
| mousemove() | 선택한 요소 범위에서 마우스 포인터를 움직였을 때 이벤트가 발생합니다. |
| scroll() | 가로, 세로 스크롤바를 움직일 때마다 이벤트가 발생합니다. |
| 포커스  이벤트 | focus() | 선택한 요소에 포커스가 생성되었을 때 이벤트를 발생하거나 선택한 요소에 강제로 포커스를 생성합니다. |
| focusin() | 선택한 요소에 포커스가 생성되었을 때 이벤트가 발생합니다. |
| focusout() | 포커스가 선택한 요소에서 다른 요소로 이동되었을 때 이벤트가 발생합니다. |
| blur() | 포커스가 선택한 요소에서 다른 요소로 이동되었을 때 이벤트가 발생하거나 선택한 요소의 포커스가 강제로 사라지도록 합니다. |
| change() | 이벤트 대상인 입력 요소의 값이 변경되고, 포커스가 이동하면 이벤트가 발생합니다. 그리고 강제로 change 이벤트를 발생시킬 때도 사용합니다. |
| 키보드  이벤트 | keypress() | 선택한 요소에서 키보드를 눌렀을 때 이벤트가 발생합니다. 그리고 문자 키를 제외한 키의 코드 값을 반환합니다. |
| keydown() | 선택한 요소에서 키보드를 눌렀을 때 이벤트가 발생합니다. 키보드의 모든 키의 코드값을 반환합니다. |
| keyup() | 선택한 요소에서 키보드에서 손을 떼었을때 발생합니다. |

**이벤트 객체와 종류**

|  |  |  |
| --- | --- | --- |
| 구분 | 종류 | 설명 |
| 마우스 이벤트 | clientX | 마우스 포인터의 X좌푯값 반환(스크롤 이동 거리 무시) |
| clientY | 마우스 포인터의 Y좌표값 반환(스크롤 이동 거리 무시) |
| pageX | 스크롤 X축의 이동한 거리를 계산하여 마우스 포인터의 X좌표값을 반환 |
| pageY | 스크롤 Y축의 이동한 거리를 계산하여 마우스 포인터의 Y기본값을 반환 |
| screenX | 화면 모니터를 기준으로 마우스 포인터의 X 좌표값을 반환 |
| screenY | 화면 모니터를 기준으로 마우스 포인터의 Y 좌표값을 반환 |
| layerX | position을 적용한 요소를 기준으로 마우스 포인터의 X 좌표값을 반환 |
| layerY | position을 적용한 요소를 기준으로 마우스 포인터의 Y 좌표값을 반환 |
| button | 마우스 버튼의 종류에 따라 값을 반환(왼쪽 0, 휠 1, 오른쪽 2) |
| 키보드 이벤트 | keyCode | 키보드의 아스키 코드값을 반환 |
| altKey | 이벤트 발생 시 Alt키가 눌렸으면 true, 아니면 false를 반환 |
| ctrlKey | 이벤트 발생 시 Ctrl키가 눌렸으면 true, 아니면 false를 반환 |
| shiftKey | 이벤트 발생 시 Shift키가 눌렸으면 true, 아니면 false를 반환 |
| 전체 이벤트 | target | 이벤트가 전파된 마지막 요소를 가리킵니다. |
| cancelBubble | 이벤트 전파를 차단하는 속성으로 기본값은 false며, true로 설정하면 전파가 차단됩니다. |
| stopPropagation | 이벤트 전파를 차단합니다. |
| preventDefault() | 기본 이벤트를 차단합니다. 예를 들어 <a>에 클릭 이벤트를 적용하고 사용자가 이벤트를 발생시키면 기본 이벤트가 등록되어 있어 링크 주소로 이용하는데, 이런 기본 이벤트를 차단할 수 있습니다. |

**그룹 이벤트 등록 메서드**

|  |  |
| --- | --- |
| 종류 | 설명 |
| on() | 이벤트 대상 요소에 2개 이상의 이벤트를 등록합니다. 사용 방식에 따라 이벤트를 등록한 이후에도 동적으로 생성되거나 복제된 요소에도 이벤트가 적용됩니다.  새롭게 생성, 복제된 요소에 이벤트 등록  $([document|”이벤트 대상의 상위 요소”]).on(”이벤트 종류”, “이벤트 대상 선택”, function() {  자바스크립트 코드; }); |
| bind() | 이벤트 대상 요소에 2개 이상의 이벤트를 등록합니다. |
| delegate() | 선택한 요소의 하위 요소에 이벤트를 등록합니다. 이벤트를 등록한 이후에도 동적으로 생성되거나 복제된 요소에도 이벤트가 적용됩니다.  $([document|”이벤트 대상의 상위 요소 선택”]).delegate(”이벤트 대상 요소”, “이벤트 종류”, function() {  자바스크립트 코드;  }); |
| one() | 이벤트 대상 요소에 1개 이상의 이벤트를 등록합니다. 지정한 이벤트가 1회 발생하고 자동으로 해제 됩니다.  \* 기본 이벤트 등록 방식  $(”이벤트 대상 선택”).one(”이벤트 종류”, function() {  자바스크립트 코드; });  \* one() 라이브 이벤트 등록 방식  $([document | “이벤트 대상의 상위 요소 선택”]).one(”이벤트 종류”, “이벤트 대상요소 선택”, function() {  자바스크립트 코드;  }); |

**이벤트 제거 메서드**

|  |  |
| --- | --- |
| 종류 | 설명 |
| off() | on() 메서드로 등록한 이벤트를 제거합니다.  **\* 기본 이벤트 제거 방식**  $(”이벤트 대상 요소 선택”).off(”이벤트 종류”);  **\* 라이브 이벤트 제거 방식**  $([document|“이벤트 대상 상위 요소 선택”]).off(”이벤트 종류”, “이벤트 대상 요소 선택”); |
| unbind() | bind() 메서드로 등록한 이벤트를 제거합니다.  $(”이벤트 대상 요소 선택”).unbind(”이벤트 종류”); |
| undelegate() | delegate() 메서드로 등록한 이벤트를 제거합니다.  **\* 기본 이벤트 제거 방식**  $(”이벤트 대상 요소 선택”).delegate(”이벤트 종류”);  **\* 라이브 이벤트 제거 방식**  $([document | “이벤트 대상의 상위 요소 선택”]).undelegate(”이벤트 대상 선택”, “이벤트 종류”); |

**애니메이션 효과 메서드**

|  |  |  |
| --- | --- | --- |
| 구분 | 종류 | 설명 |
| 숨김 | hide() | 요소를 숨깁니다. |
| fadeOut() | 요소가 점점 투명해지면서 사라집니다. |
| slideUp() | 요소가 위로 접히면서 숨겨집니다. |
| 노출 | show() | 숨겨진 요소가 노출됩니다. |
| fadeIn() | 숨겨진 요소가 점점 선명해집니다. |
| slideDown() | 숨겨진 요소가 아래로 펼쳐집니다. |
| 노출, 숨김 | toggle() | hide(), show() 효과를 적용합니다. |
| fadeToggle() | fadeIn(), fadeOut() 효과를 적용합니다. |
| slideToggle() | slideUp(), slideDown() 효과를 적용합니다. |
| fadeTo() | 지정한 투명도를 적용합니다 |

**animate() 메서드**

스타일 속성에 따른 동적 효과를 적용할 수 있습니다.

$(”요소 선택”).animate({스타일 속성(1)}, 적용시간(2), 가속도(3), 콜백 함수(4));

**애니메이션 효과 제어 메서드**

|  |  |
| --- | --- |
| 종류 | 설명 |
| stop() | 현재 실행 중인 효과를 모두 정지시킵니다.  (1) $(”요소선택”).stop();  (2) $(”요소선택”).stop(clearQueue, finish);  (1) 진행 중인 첫 번째 애니메이션만 정지시킵니다. 큐에 대기 중인 애니메이션은 계속 실행합니다.  (2) clearQueue가 true면 큐에서 대기 중인 애니메이션을 모두 제거, finish가 true면 진행 중인 애니메이션을 강제로 종료 |
| delay() | 지정한 시간만큼 지연했다가 애니메이션을 진행합니다.  $(”요소 선택”).delay(지연시간).애니메이션 효과 메서드(); |
| queue() | 큐에 사용자 정의 함수를 추가하거나 큐에 대기 중인 함수를 배열에 담아 반환합니다. 그리고 queue() 메서드 이후의 애니메이션 효과 메서드는 모두 취소합니다.  \* 큐(Queue)의 함수 반환  $(”요소 선택”).queue();  \* 큐(Queue)에 함수 추가  $(”요소 선택”).queue(function() { 자바스크립트 코드 }); |
| clearQueue() | 큐에서 처음으로 진행하고 있는 애니메이션만 제외하고 대기 중인 애니메이션은 모두 제거합니다.  $(”요소선택”).clearQueue(); |
| dequeue() | queue() 메서드를 이용하면 대기하고 있는 애니메이션 메서드는 제거됩니다. 하지만 dequeue() 메서드를 이용하면 메서드가 제거되는 것을 막을 수 있습니다.  $(”요소 선택”).dequeue(); |
| finish() | 선택한 요소의 진행 중인 애니메이션을 강제로 완료 시점으로 보낸 후 종료합니다. |

**Ajax 메서드**

|  |  |
| --- | --- |
| 종류 | 설명 |
| load() | 외부 콘텐츠를 가져올 때 사용합니다.  $(요소 선택).load(url, data, 콜백함수); |
| $.ajax() | 데이터 서버에 HTTP, POST, GET 방식으로 전송할 수 있으며, HTML, XML, JSON 텍스트 유형에 데이터를 요청힐 수 있는 통합적인 메서드 입니다. 이 표에 있는 $.post(), $.get(), $.getJSON() 메서드의 기능을 하나로 합쳐 놓은 것이라 보면 됩니다.  $.ajax({  url : “전송 페이지(action url)”,  type : “전송 방식(get, post)”,  data : “전송할 데이터”,  dataType : “요청한 데이터 형식(”html”, “xml”, “json”, “text”, “jsonp”)”,  success : function(data) {  전송에 성공하면 실행될 코드;  },  error : fuiinction() {  전송에 실패하면 실행될 코드;  } }); |
| $.post() | 데이터를 서버에 HTTP POST 방식으로 전송한 후 서버 측의 응답을 받을 때 사용합니다. |
| $.get() | 데이버를 서버에 HTTP GET 방식으로 전송한 후 서버 측의 응답을 받을 때 사용합니다. |
| $.getJSON() | 데이터를 서버에 HTTP GET 방식으로 전송한 후 서버 측의 응답을 JSON 형식으로 받을 때 사용합니다. |
| $.getScript() | Ajax를 이용하여 외부 자바스크립트를 불러옵니다.  $(”button”).click(function() {  $.getScript(”demo\_ajax\_script.js”); }); |
| ajaxStop(function(){...}) | 비동기 방식으로 서버에 응답 요청이 완료되었을 때 함수가 실행됩니다. |
| ajaxSuccess(function(){...}) | ajax 요청이 성공적으로 완료되면 함수가 실행됩니다. |
| ajaxComplete(function(){...}) | Ajax 통신이 완료되면 함수가 실행됩니다. |

**$.ajax() 메서드의 옵션 종류**

|  |  |
| --- | --- |
| 종류 | 설명 |
| async | 통신을 동기 또는 비동기 방식으로 설정하는 옵션입니다. 기본값은 비동기 통신 방식인 true로 설정되어 있습니다. 만일 비동기 방식으로 설정되어 있다면 사용자 컴퓨터에서 서버로 데이터를 전송하고 요청하는 동안에도 다른 작업을 할 수 있습니다. |
| beforeSend | 요청하기 전에 함수를 실행하는 이벤트 핸들러입니다. |
| cache | 요청한 페이지를 인터넷에 캐시(저장)할지의 여부를 설정합니다. 기본값은 true 입니다. |
| complete | Ajax가 완료되었을 때 함수를 실행하는 이벤트 핸들러입니다. |
| data | 서버로 전송할 데이터를 지정합니다. |
| dataType | 서버에서 받아올 데이터의 형식을 지정합니다. 생략하면 요청한 자료에 맞게 자동으로 형식이 설정됩니다. |
| error | 통신에 문제가 발생했을 때 함수를 실행합니다. |
| success | Ajax로 통신이 정상적으로 이뤄지면 이벤트 핸들러를 실행합니다. |
| timeout | 통신 시간을 제한합니다. 시간 단위는 밀리초입니다. |
| type | 데이터를 전송할 방식(get, post)를 설정합니다. |
| url | 데이터를 전송할 페이지를 설정합니다. 기본값은 현재 페이지 입니다. |
| username | HTTP 액세스를 할 떄 인증이 필요한 경우 사용자 이름을 지정합니다. |
|  |  |

**Ajax 전송 데이터 가공 메서드의 종류**

|  |  |  |
| --- | --- | --- |
| 종류 | 사용법 | 설명 |
| serialize() | $(”fom”).serialize() | 사용자가 입력 요소에 값을 입력한 데이터의 전송방식을 'name1=value1 & name2=value2,...'와 같은 쿼리 스트링 형태의 데이터로 변환해 반환합니다. |
| serializeArray() | $(”form”).serializeArray() | 사용자가 입력요소에 값을 입력한 데이터의 전송 방식을[{name1 : value1}, {name2 : value2}]와 같은 배열 객체로 변환해 반환합니다. |
| $.param() | $.param( Object ); | {name1 : value1, name2 : value2}와 같이 작성된 객체를 가공해 'name1=value1&name2=value2, ..'와 같은 쿼리 스트링 형식의 데이터로 변환해 반환합니다. |
| JSON.parse() | JSON.parse('{”name”:”value”}'); | 객체 형태로 작성한 문자열 데이터를 객체로 가공하여 변환합니다, |
| JSON.stringify() | JSON.stringify({”name1”:”value1”, “name2” : “value2”}); | 객체를 문자열 데이터로 가공하여 반환합니다. |