

# JQUERY 빠른 참조

선택어, DOM, 이벤트, AJAX, 효과, 탐색

## 선택어

기본 선택어	
<code>\$( "p" )</code>	// all <p> elements
<code>\$( ".btn" )</code>	// class selector
<code>\$( "#main" )</code>	// ID selector
<code>\$( "ul &gt; li" )</code>	// direct children
<code>\$( "input[type='text']" )</code>	// attribute selector
주요 선택어	
<code>\$( "*" )</code>	모든 요소
<code>\$( "a, span" )</code>	여러 선택어
<code>\$( ":first" )</code>	첫 번째로 매칭된 요소
<code>\$( ":last" )</code>	마지막으로 매칭된 요소
<code>\$( ":even" ) / \$( ":odd" )</code>	짝수/홀수 인덱스 요소
<code>\$( ":visible" ) / \$( ":hidden" )</code>	가시성 필터
<code>\$( ":contains(text)" )</code>	텍스트를 포함하는 요소
<code>\$( ":has(selector)" )</code>	선택어와 매칭되는 하위 요소를 가진 요소

## DOM 조작

콘텐츠 및 구조	
<code>\$( "#el" ).html( "&lt;b&gt;Bold&lt;/b&gt;" );</code>	// set inner HTML
<code>\$( "#el" ).text( "Plain text" );</code>	// set text content
<code>\$( "ul" ).append( "&lt;li&gt;New&lt;/li&gt;" );</code>	// add at end
<code>\$( "ul" ).prepend( "&lt;li&gt;First&lt;/li&gt;" );</code>	// add at start
삽입 및 제거	
<code>\$( "&lt;p&gt;Hello&lt;/p&gt;" ).insertAfter( "#el" );</code>	// insert after
<code>\$( "&lt;p&gt;Hi&lt;/p&gt;" ).insertBefore( "#el" );</code>	// insert before
<code>\$( "#el" ).remove();</code>	// remove element from DOM
<code>\$( "#el" ).empty();</code>	// remove all children
메서드	
<code>\$.html()</code>	내부 HTML 가져오기/설정
<code>\$.text()</code>	텍스트 콘텐츠 가져오기/설정
<code>\$.append()</code> / <code>\$.prepend()</code>	끝/처음에 자신 추가
<code>\$.after()</code> / <code>\$.before()</code>	뒤/앞에 형제 요소 삽입
<code>\$.wrap()</code>	새 부모로 요소 감싸기
<code>\$.clone()</code>	요소 깊은 복사
<code>\$.replaceWith()</code>	요소 교체

## 이벤트

이벤트 바인딩	
<code>\$( ".btn" ).on( "click", function () {</code>	
<code>  \$( this ).toggleClass( "active" );</code>	
<code>});</code>	
<code>\$( document ).on( "click", ".dynamic", handler );</code>	// delegation
<code>\$( ".btn" ).off( "click" );</code>	// remove handler
단축 메서드	
<code>\$.click(fn)</code>	클릭 핸들러
<code>\$.hover(fnIn, fnOut)</code>	마우스 진입/이탈
<code>\$.focus()</code> / <code>\$.blur()</code>	포커스 획득/해제
<code>\$.change(fn)</code>	값 변경 (input/select)
<code>\$.submit(fn)</code>	폼 제출
<code>\$.keydown()</code> / <code>\$.keyup()</code>	키보드 이벤트
문서 준비	
<code>\$( document ).ready( function () { /* DOM loaded */ });</code>	
<code>\$( function () { /* shorthand */ });</code>	

## 효과 및 애니메이션

보이기 / 숨기기 / 토글	
<code>\$( "#el" ).hide( 400 );</code>	// hide with animation
<code>\$( "#el" ).show( 400 );</code>	// show with animation
<code>\$( "#el" ).toggle();</code>	// toggle visibility
<code>\$( "#el" ).fadeIn( 600 );</code>	// fade in
<code>\$( "#el" ).slideDown( 400 );</code>	// slide down
커스텀 애니메이션	
<code>\$( "#el" ).animate( {</code>	
<code>  opacity: 0.5,</code>	
<code>  left: "+=80px",</code>	
<code>  height: "toggle"</code>	
<code>}, 1000, function () { /* complete */ });</code>	
효과 메서드	
<code>\$.fadeIn()</code> / <code>\$.fadeOut()</code>	투명도 애니메이션
<code>\$.fadeToggle()</code>	페이드로 토글
<code>\$.slideDown()</code> / <code>\$.slideUp()</code>	높이 애니메이션
<code>\$.animate( props, ms )</code>	커스텀 CSS 애니메이션
<code>\$.stop()</code>	현재 애니메이션 중지
<code>\$.delay( ms )</code>	다음 대기 효과 지연

## AJAX

단축 메서드	
<code>\$.get( "/api/data", function ( data ) { console.log( data ); } );</code>	
<code>\$.post( "/api/data", { name: "Alice" }, function ( res ) {} );</code>	
<code>\$.getJSON( "/api/items", function ( json ) {} );</code>	
<code>\$( "#el" ).load( "/fragment.html" );</code>	// load HTML into element
\$.ajax 전체 사용	
<code>\$.ajax( {</code>	
<code>  url: "/api/data",</code>	
<code>  method: "POST",</code>	
<code>  data: JSON.stringify( { key: "val" } ),</code>	
<code>  contentType: "application/json",</code>	
<code>  success: ( res ) =&gt; console.log( res ),</code>	
<code>  error: ( xhr ) =&gt; console.error( xhr.status )</code>	
<code>};</code>	
AJAX 옵션	
<code>url</code>	요청 URL
<code>method</code>	HTTP 메서드 (GET, POST, ...)
<code>data</code>	전송할 데이터
<code>dataType</code>	기대하는 응답 타입
<code>contentType</code>	요청 콘텐츠 타입
<code>success / error</code>	콜백 함수

## headers

커스텀 요청 헤더	
탐색	
트리 탐색	
<code>\$( "li" ).parent();</code>	// direct parent
<code>\$( "li" ).parents( "ul" );</code>	// all matching ancestors
<code>\$( "ul" ).children( "li" );</code>	// direct children
<code>\$( "ul" ).find( ".active" );</code>	// all matching descendants
<code>\$( "li" ).siblings();</code>	// sibling elements
탐색 메서드	
<code>\$.parent()</code> / <code>\$.parents()</code>	직전 부모 / 모든 상위 요소
<code>\$.children()</code> / <code>\$.find()</code>	직전 자식 / 모든 하위 요소
<code>\$.siblings()</code>	모든 형제 요소
<code>\$.next()</code> / <code>\$.prev()</code>	다음 이전 형제
<code>\$.closest(sel)</code>	선택어와 매칭되는 가장 가까운 상위 요소
<code>\$.first()</code> / <code>\$.last()</code>	매칭된 집합의 첫 번째/마지막
<code>\$.eq(index)</code>	인덱스의 요소
<code>\$.filter(sel)</code>	매칭되는 요소로 축소

## 속성 및 CSS

속성	
<code>\$( "img" ).attr( "src" );</code>	// get attribute
<code>\$( "img" ).attr( "src", "new.jpg" );</code>	// set attribute
<code>\$( "img" ).removeAttr( "title" );</code>	// remove attribute
<code>\$( "input" ).prop( "checked", true );</code>	// set property
CSS 및 클래스	
<code>\$( ".box" ).css( "color", "red" );</code>	// set single style
<code>\$( ".box" ).css( { color: "red", fontSize: "14px" } );</code>	// multi
<code>\$( "p" ).addClass( "highlight" );</code>	
<code>\$( "p" ).removeClass( "highlight" );</code>	
<code>\$( "p" ).toggleClass( "active" );</code>	
<code>\$( "p" ).hasClass( "active" );</code>	// returns boolean

## 폼

값 및 직렬화	
<code>\$( "input" ).val();</code>	// get value
<code>\$( "input" ).val( "new value" );</code>	// set value
<code>\$( "form" ).serialize();</code>	// "name=Alice&age=30"
<code>\$( "form" ).serializeArray();</code>	// [{name, value}, ...]
폼 선택어	
<code>\$( ":input" )</code>	모든 폼 요소
<code>\$( ":checked" )</code>	체크된 체크박스/라디오
<code>\$( ":selected" )</code>	선택된 option 요소
<code>\$( ":disabled" )</code>	비활성화된 요소
<code>\$( ":text" )</code>	텍스트 입력

## 유틸리티

배열 및 객체 헬퍼	
<code>\$.each( [ 1, 2, 3 ], function ( i, val ) { /* iterate */ });</code>	
<code>\$.each( obj, function ( key, val ) { /* iterate object */ });</code>	
<code>\$.extend( {}, defaults, options );</code>	// merge objects
<code>\$.inArray( "b", [ "a", "b", "c" ] );</code>	// returns 1 (index)
유틸리티 메서드	
<code>\$.each( arr, fn )</code>	배열 또는 객체 순회
<code>\$.extend( target, ... )</code>	객체 병합 (얕은 복사)
<code>\$.inArray( val, arr )</code>	값의 인덱스 (-1이면 없음)
<code>\$.isArray( obj )</code>	배열인지 확인
<code>\$.trim( str )</code>	앞뒤 공백 제거
<code>\$.parseJSON( str )</code>	JSON 문자열 파싱
<code>\$.map( arr, fn )</code>	배열 요소 변환

## 주요 패턴

체이닝	
<code>\$( "ul" )</code>	
<code>  .find( "li" )</code>	
<code>  .addClass( "item" )</code>	
<code>  .first();</code>	
<code>  .css( "font-weight", "bold" );</code>	
플러그인 패턴	
<code>\$.fn.highlight = function ( color ) {</code>	
<code>  return this.each( function () {</code>	
<code>    \$( this ).css( "background", color    "yellow" );</code>	
<code>  });</code>	
<code>\$( "p" ).highlight( "pink" );</code>	
충돌 해결	
<code>// Avoid \$ conflicts with other libraries</code>	
<code>jQuery.noConflict();</code>	
<code>jQuery( document ).ready( function (\$) {</code>	
<code>  \$( ".btn" ).click( handler );</code>	// \$ safe inside
<code>});</code>	