

Referencia Rápida de jQuery

Selectores, DOM, eventos, AJAX, efectos, recorrido

Selectores

Selectores básicos

```
$("#p") // all <p> elements
$(".btn") // class selector
$("#main") // ID selector
$("#ul > li") // direct children
$("#input[type='text']") // attribute selector
```

Selectores comunes

\$('.*')	Todos los elementos
\$('.a, span')	Selectores múltiples
\$('.:first')	Primer elemento coincidente
\$('.:last')	Último elemento coincidente
\$('.:even') / \$('.:odd')	Elementos con índice par/impar
\$('.:visible') / \$('.:hidden')	Filtros de visibilidad
\$('.:contains(text)')	Elementos que contienen texto
\$('.:has(selector)')	Elementos que contienen una coincidencia del selector

Manipulación del DOM

Contenido y estructura

```
$("#el").html("<b>Bold</b>"); // set inner HTML
$("#el").text("Plain text"); // set text content
$("#ul").append("<li>New</li>"); // add at end
$("#ul").prepend("<li>First</li>"); // add at start
```

Insertar y eliminar

```
$("#<p>Hello</p>").insertAfter("#el"); // insert after
$("#<p>Hi</p>").insertBefore("#el"); // insert before
$("#el").remove(); // remove element from DOM
$("#el").empty(); // remove all children
```

Métodos

.html()	Obtener/establecer HTML interno
.text()	Obtener/establecer contenido de texto
.append() / .prepend()	Agregar hijo al final/inicio
.after() / .before()	Insertar hermano después/antes
.wrap()	Envolver el elemento con un nuevo padre
.clone()	Copia profunda del elemento
.replaceWith()	Reemplazar el elemento

Eventos

Vincular eventos

```
$("#.btn").on("click", function () {
    $(this).toggleClass("active");
});
$(document).on("click", ".dynamic", handler); // delegation
$("#.btn").off("click"); // remove handler
```

Métodos abreviados

.click(fn)	Manejador de clic
.hover(fnIn, fnOut)	Entrada/salida del ratón
.focus() / .blur()	Foco ganado/perdido
.change(fn)	Valor cambiado (input/select)
.submit(fn)	Envío del formulario
.keydown() / .keyup()	Eventos de teclado

Document ready

```
$(document).ready(function () { /* DOM loaded */ });
$(function () { /* shorthand */ });
```

Efectos y animación

Mostrar / Ocultar / Alternar

```
$("#el").hide(400); // hide with animation
$("#el").show(400); // show with animation
$("#el").toggle(); // toggle visibility
$("#el").fadeIn(600); // fade in
$("#el").slideDown(400); // slide down
```

Animación personalizada

```
$("#el").animate({
    opacity: 0.5,
    left: "+=50px",
    height: "toggle"
}, 1000, function () { /* complete */ });
```

Métodos de efecto

.fadeIn() / .fadeOut()	Animar opacidad
.fadeToggle()	Alternar con fundido
.slideDown() / .slideUp()	Animar altura
.animate(props, ms)	Animación CSS personalizada
.stop()	Detener animación actual
.delay(ms)	Retrasar el siguiente efecto en cola

AJAX

Métodos abreviados

```
$.get("/api/data", function (data) { console.log(data); });
$.post("/api/data", { name: "Alice" }, function (res) {});
$.getJSON("/api/items", function (json) {});
$("#el").load("/fragment.html"); // load HTML into element
```

\$.ajax completo

```
$.ajax({
    url: "/api/data",
    method: "POST",
    data: JSON.stringify({ key: "val" }),
    contentType: "application/json",
    success: (res) => console.log(res),
    error: (xhr) => console.error(xhr.status)
});
```

Opciones de AJAX

url	URL de la petición
method	Método HTTP (GET, POST, ...)
data	Datos a enviar
dataType	Tipo de respuesta esperado
contentType	Tipo de contenido de la petición
success / error	Funciones de callback
headers	Cabeceras personalizadas de la petición

Recorrido del DOM

Navegación por el árbol

```
$("#li").parent(); // direct parent
$("#li").parents("ul"); // all matching ancestors
$("#ul").children("li"); // direct children
$("#ul").find(".active"); // all matching descendants
$("#li").siblings(); // sibling elements
```

Métodos de recorrido

.parent() / .parents()	Padre directo / todos los ancestros
.children() / .find()	Hijos directos / todos los descendientes
.siblings()	Todos los elementos hermanos
.next() / .prev()	Hermano siguiente/anterior
.closest(sel)	Ancstro más cercano que coincide con el selector
.first() / .last()	Primero/último del conjunto coincidente
.eq(index)	Elemento en el índice
.filter(sel)	Reducir a los elementos coincidentes

Atributos y CSS

Atributos

```
$("#img").attr("src"); // get attribute
$("#img").attr("src", "new.jpg"); // set attribute
$("#img").removeAttr("title"); // remove attribute
$("#input").prop("checked", true); // set property
```

CSS y clases

```
$(".box").css("color", "red"); // set single style
$(".box").css({ color: "red", fontSize: "14px" }); // multi
$("#p").addClass("highlight");
$("#p").removeClass("highlight");
$("#p").toggleClass("active");
$("#p").hasClass("active"); // returns boolean
```

Formularios

Valores y serialización

```
$("#input").val(); // get value
$("#input").val("new value"); // set value
$("#form").serialize(); // "name=Alice&age=30"
$("#form").serializeArray(); // [{name, value}, ...]
```

Selectores de formulario

\$("#:input")	Todos los elementos de formulario
\$("#:checked")	Checkboxes/radios marcados
\$("#:selected")	Opciones seleccionadas
\$("#:disabled")	Elementos deshabilitados
\$("#:text")	Inputs de texto

Utilidades

Helpers de array y objeto

```
$.each([1, 2, 3], function (i, val) { /* iterate */ });
$.each(obj, function (key, val) { /* iterate object */ });
$.extend({}, defaults, options); // merge objects
$.inArray("b", ["a", "b", "c"]); // returns 1 (index)
```

Métodos de utilidad

\$.each(arr, fn)	Iterar array u objeto
\$.extend(target, ...)	Fusionar objetos (superficial)
\$.inArray(val, arr)	Índice del valor (-1 si no existe)
\$.isArray(obj)	Verificar si es array
\$.trim(str)	Eliminar espacios al inicio/final
\$.parseJSON(str)	Parsear cadena JSON
\$.map(arr, fn)	Transformar elementos del array

Referencia Rápida de jQuery

Patrones comunes

Encadenamiento

```
$( "ul" )
  .find( "li" )
  .addClass( "item" )
  .first()
  .css( "font-weight", "bold" );
```

Patrón de plugin

```
$.fn.highlight = function (color) {
  return this.each(function () {
    $(this).css("background", color || "yellow");
  });
};
$("p").highlight("pink");
```

Resolución de conflictos

```
// Avoid $ conflicts with other libraries
jQuery.noConflict();
jQuery(document).ready(function ($) {
  $(".btn").click(handler); // $ safe inside
});
```