

jQuery 快速参考

选择器、DOM、事件、AJAX、动效、遍历

选择器

基本选择器

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

常用选择器

<code>\$('*')</code>	所有元素
<code>\$('a, span')</code>	多选择器
<code>\$(':first')</code>	第一个匹配元素
<code>\$(':last')</code>	最后一个匹配元素
<code>\$(':even') / \$(':odd')</code>	偶数/奇数索引元素
<code>\$(':visible') / \$(':hidden')</code>	可见性过滤器
<code>\$(':contains(text)')</code>	包含文本的元素
<code>\$(':has(selector)')</code>	包含选择器匹配的元素

DOM 操作

内容与结构

```
$( "#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
```

插入与删除

```
$( "<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
```

方法

<code>.html()</code>	获取/设置内部 HTML
<code>.text()</code>	获取/设置文本内容
<code>.append()</code> / <code>.prepend()</code>	在末尾/开头添加子元素
<code>.after()</code> / <code>.before()</code>	在后面/前面插入兄弟元素
<code>.wrap()</code>	用新父元素包裹元素
<code>.clone()</code>	深度复制元素
<code>.replaceWith()</code>	替换元素

事件

绑定事件

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

简写方法

<code>.click(fn)</code>	点击事件处理器
<code>.hover(fnIn, fnOut)</code>	鼠标进入/离开
<code>.focus()</code> / <code>.blur()</code>	获得/失去焦点
<code>.change(fn)</code>	值改变 (input/select)
<code>.submit(fn)</code>	表单提交
<code>.keydown()</code> / <code>.keyup()</code>	键盘事件

文档就绪

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

动效与动画

显示 / 隐藏 / 切换

```
$( "#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
```

自定义动画

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

动效方法

<code>.fadeIn()</code> / <code>.fadeOut()</code>	透明度动画
<code>.fadeToggle()</code>	淡入淡出切换
<code>.slideDown()</code> / <code>.slideUp()</code>	高度动画
<code>.animate(props, ms)</code>	自定义 CSS 动画
<code>.stop()</code>	停止当前动画
<code>.delay(ms)</code>	延迟下一个排队的动效

AJAX

简写方法

```
$.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
```

完整 \$.ajax

```
$.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)
});
```

AJAX 选项

url	请求 URL
method	HTTP 方法 (GET、POST 等)
data	发送的数据
dataType	期望的响应类型
contentType	请求内容类型
success / error	回调函数
headers	自定义请求头

遍历

树形导航

```
$( "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
```

遍历方法

<code>.parent()</code> / <code>.parents()</code>	直接父元素 / 所有祖先
<code>.children()</code> / <code>.find()</code>	直接子元素 / 所有后代
<code>.siblings()</code>	所有兄弟元素
<code>.next()</code> / <code>.prev()</code>	下一个/上一个兄弟
<code>.closest(sel)</code>	最近的匹配祖先
<code>.first()</code> / <code>.last()</code>	匹配集中的第一个/最后一个
<code>.eq(index)</code>	指定索引处的元素
<code>.filter(sel)</code>	缩小到匹配的元素

属性与 CSS

属性

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

CSS 与类

```
$( ".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
```

表单

值与序列化

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

表单选择器

<code>\$(":input")</code>	所有表单元素
<code>\$(":checked")</code>	已选中的复选框/单选框
<code>\$(":selected")</code>	已选中的 option 元素
<code>\$(":disabled")</code>	已禁用的元素
<code>\$(":text")</code>	文本输入框

工具方法

数组与对象辅助

```
$.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)
```

工具方法列表

<code>\$.each(arr, fn)</code>	遍历数组或对象
<code>\$.extend(target, ...)</code>	合并对象 (浅拷贝)
<code>\$.inArray(val, arr)</code>	值的索引 (不存在返回 -1)
<code>\$.isArray(obj)</code>	判断是否为数组
<code>\$.trim(str)</code>	去除首尾空白
<code>\$.parseJSON(str)</code>	解析 JSON 字符串
<code>\$.map(arr, fn)</code>	转换数组元素

jQuery 快速参考

常见模式

链式调用

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

插件模式

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

冲突解决

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