

张景元

选择器

- 1.: `first` 整个查询结果中的第一个元素
- 2.: `last` 整个查询结果中的最后一个元素
- 3.: `even` 整个查询结果中的偶数下标位置元素
- 4.: `odd` 整个查询结果中的奇数下标位置元素
- 5.: `eq(i)` 整个查询结果中的下标为 `i` 位置的元素
- 6.: `gt(i)` 整个查询结果中下标大于 (`>`) `i` 位置的元素
- 7.: `lt(i)` 整个查询结果中小标小于 (`<`) `i` 位置的元素
- 8.: `first-child` 父元素下的第一个元素
- 9.: `last-child` 父元素下的最后一个元素
- 10.: `nth-child(i)` 父元素下 `i` 位置的元素
- 11.: `nth-child(2n)` 父元素下偶数的元素
- 12.: `nth-child(2n+1)` 父元素下奇数的元素
- 13.: `contains(关键词)` 匹配元素内容包含指定关键词的元素
- 14.: `has(选择器)` 匹配子元素中包含符合选择器要求的元素的父元素 //根据孩子选爹 张景元
- 15.: `parent` 匹配所有内容不为空的元素
- 16.: `empty` 匹配内容为空的元素//空字符也不能有
- 17.: `visible` 匹配可见的元素

18. : `hidden` 匹配不可见的元素//只能选择 `display: none` 和 `type= "hidden"` 隐藏元素

19. : `input` 匹配所有文本框的元素

20. : `button` 匹配所有按钮的元素

21. : `textarea` 匹配所有多行文本域的元素

22. : `select` 匹配所有下拉列表的元素

张景元

23. : `text` 专门选择 `input type= "text"` 元素//文本框 //所有按钮都 `disabled` 禁用

24. : `password` 专门选择 `input type= "password"` 元素//密码框

25. : `radio` 专门选择 `input type= "radio"` 元素//单选按钮 //checked 默认选中

26. : `checkbox` 专门选择 `input type= "checkbox"` 元素//多选按钮 //checked 默认选中

27. : `submit` 专门选择 `input type= "submit"` 元素//提交按钮

28. : `reset` 专门选择 `input type= "reset"` 元素//重置按钮

29. : `file` 专门选择 `input type= "file"` 元素//上传文件

张景元

函数库

1. `$元素.html(“新内容”)` 修改或获取 html 内容
2. `$元素.text(“新内容”)` 修改或获取纯文本内容
3. `$元素.val(“新 值”)` 修改或获取表单元素的值
4. `$元素.trim()` 去掉空字符
5. `$元素.attr(“属性名”,“新值”)` 修改或替换 html 中开始标签中的属性
6. `$元素.prop(“属性名”,“新值”)` 修改或替换内存中元素对象上的属性
7. `$元素.css(“css 属性名”,“属性值”)` 修改和获取 css 属性
8. `$元素.addClass(“class 名”)` 添加 class
9. `$元素.removeClass(“class 名”)` 移除 class
10. `$元素.hasClass(“class 名”)` 判断是否包含某个 class
11. `$元素.toggleClass(“class 名”)` 切换有没有一个 class
12. `$元素.parent()` 获得一个元素的父元素
13. `$元素.children()` 获得一个元素下所有直接子元素
14. `$元素.children(选择器)` 获得一个元素下符合条件的所有子元素

15. `$元素.find(选择器)` 在所有后代元素中查找符合要求的元素
16. `$元素.children(":first-child")` 获得一个元素下第一个直接子元素 **张景元**
17. `$元素.children(":last-child")` 获得一个元素下最后一个直接子元素
18. `$元素.children(":nth-child(i)")` 获得一个元素下第 i 个直接子元素
19. `$元素.children(":nth-child(2n)")` 获得一个元素下所有偶数的直接子元素
20. `$元素.children(":odd")` 获得一个元素下所有下标奇数的直接子元素//所有偶数的直接子元素
21. `$元素.children(":nth-child(2n+1)")` 获得一个元素下所有奇数直接子元素
22. `$元素.children(":even")` 获得一个元素下所有下标偶数的直接子元素//所有奇数的直接子元素
23. `$元素.prev()` 当前元素的前一个兄弟
24. `$元素.prevAll(选择器)` 当前元素的之前所有兄弟
25. `$元素.next()` 当前元素的下一个兄弟
26. `$元素.nextAll(选择器)` 当前元素的之后所有

兄弟

27. `$元素.siblings(选择器)` 除当前元素之外其余所有兄弟元素（不论前后）
28. `$(`html 代码片段`)` 创建一个或多个元素
29. `$父元素.append($新元素)` 父元素末尾追加新元素
`|| $新元素.appendTo($父元素)`
30. `$父元素.prepend($新元素)` 父元素开头追加新元素
`|| $新元素.prependTo($父元素)`
31. `$现有元素.before($新元素)` 插入到一个现有元素之前
`|| $新元素.insertBefore($现有元素)`
32. `$现有元素.after($新元素)` 插入到一个现有元素之后
`|| $新元素.insertAfter($现有元素)`
33. `$现有元素.replaceWith($新元素)` 替换父元素下一个现有子元素
`|| $新元素.replaceAll($现有元素)`
34. `$元素.remove()` 删除
35. `$新元素副本=$原元素.clone()` 克隆
36. `$元素.on(“事件名”, 事件处理函数)` 添加事件监听
37. `$元素.off(“事件名”, 原事件处理函数)` 移除事件监听
38. 可简化常用事件名

- 39. `$元素.click(事件处理函数)` 单击事件
- 40. `$元素.blur(事件处理函数)` 失去焦点事件
- 41. `$元素.change(事件处理函数)` 下拉列表选中项
改变事件
- 42. `$元素.dblclick(事件处理函数)` 双击事件
- 43. `$元素.focus(事件处理函数)` 获得焦点事件
- 44. `$元素.keydown(事件处理函数)` 键盘按键按下
事件
- 45. `$元素.keyup(事件处理函数)` 键盘按键抬起事
件
- 46. `$元素.mousedown(事件处理函数)` 鼠标按键按
下事件
- 47. `$元素.mouseenter(事件处理函数)` 鼠标进入事
件//jq
- 48. `$元素.mouseleave(事件处理函数)` 鼠标移出事
件//jq
- 49. `$元素.mousemove(事件处理函数)` 鼠标移动事
件
- 50. `$元素.mouseout(事件处理函数)` 鼠标移出事件
//dom
- 51. `$元素.mouseover(事件处理函数)` 鼠标进入事
件//dom

- 52. `$元素.mouseup(事件处理函数)` 鼠标按键抬起事件
- 53. `$元素.resize(事件处理函数)` 窗口大小改变事件
- 54. `$元素.scroll(事件处理函数)` 网页滚动事件
- 55. `Window.onload=function() {}` 整个页面加载完成时，浏览器会自动触发一个 load 事件
- 56. `$(function() {})` 网页中 html+js 都加载完成，才自动执行
- 57. `$元素.hover(事件处理函数 1, 事件处理函数 2)` 同时绑定鼠标进入和鼠标移除事件
- 58. `$元素.trigger(事件名)` 模拟触发 || `$元素.事件名()` 简写
- 59. `$元素.show()` 显示 `//display: block`
- 60. `$元素.hide()` 隐藏 `//display: none`
- 61. `$元素.toggle()` 在显示和隐藏之间来回切换
- 62. `$元素.slideUp()` 上滑
- 63. `$元素.slideDown()` 下滑
- 64. `$元素.slideToggle()` 在上滑和下滑之间来回切换
- 65. `$元素.fadeIn()` 淡入
- 66. `$元素.fadeOut()` 淡出

- 67. `$元素.fadeToggle()` 在淡入和淡出之间来回切换
- 68. `$元素.animate({css 属性: 目标值}, 动画持续时间)` 万能动画
- 69. `$元素.stop()` 停止动画
- 70. `$元素.stop(true)` 即停止当前正在播放动画，有清空后续队列
- 71. `$元素.each(function(I, elem) {})` 遍历类数组对象
- 72. `$元素.index(dom 元素)` 查找类数组 dom 元素出现的下标位置，如果找不到返回-1
- 73. `e.stopPropagation()` 取消冒泡//停止蔓延
- 74. `e.screenX e.screenY` 鼠标位置相对屏幕左上角的距离
- 75. `e.clientX e.clientY` 鼠标位置相对浏览器文档显示区左上角的距离
- 76. `e.offsetX e.offsetY` 鼠标位置相对事件所在元素左上角的距离
- 77. `jQuery.prototype.自定义函数=function(形参列表) {}` 添加自定义函数
- 78. `$.ajax({ //发送 ajax 请求的函数`
`url: 服务器端接口地址`

type: get 或 post

data: {变量: 值, 变量: 值} // 发送到服务器端的参数变量和值

dataType: "json" // 因为几乎所有服务器端返回的结果都是 json 字符串, 都必须反复调用 JSON.parse() 将 json 字符串, 转为 js 中的对象和数组, 程序才能用

success: function(result) {

 // 只要请求响应成功, 成功获得服务器端返回的结果时, 自动调用 success 回调函数。并自动将服务器端返回的结果, 经过 JSON.parse() 后翻译为对象, 交给 result 变量。

 }

})