

jQuery

2006年初发布的一个JavaScript库，极大简化了JS编程。

jQuery库包含以下功能：

- HTML 元素选取
- HTML 元素操作
- CSS 操作
- HTML DOM 遍历和修改
- HTML 事件函数
- JavaScript 特效和动画
- **AJAX**
- Utilities

除了上述功能外，它之所以流行，是官方、非官方提供了海量的插件。

安装

从官网下载 <https://jquery.com/download/>，或使用CDN。

开发用的jquery-3.6.0.js或部署用的压缩版jquery-3.6.0.min.js。

```
1 <script src="js/jquery.xxx.js"></script>
```

Flask静态部署

```
1 ('root_path', 'E:\\ClassProjects\\test')
2 ('_static_folder', 'static')
3
4 ('url_map', Map([
5     ...,
6     <Rule '/static/<filename>' (OPTIONS, HEAD, GET) -> static>]))
```

将js文件、css文件一般都是放到root_path下的static目录中。

在root_path目录下创建static目录，在static目录中创建css和js目录，css目录中放置bootstrap.min.css，js目录中放置jquery-3.6.0.min.js和bootstrap.bundle.min.js

修改templates/index.html如下

```

1 <!DOCTYPE html>
2 <html lang="zh-CN">
3 <head>
4     <meta charset="UTF-8">
5     <link rel="stylesheet" href="{{ url_for('static',
filename='css/bootstrap.min.css') }}">
6     <script src="{{ url_for('static', filename='js/jquery-3.6.0.min.js')
}}"></script>
7     <script src="/static/js/bootstrap.bundle.min.js"></script>
8     <title>马哥教育jQuery测试</title>
9 </head>
10 <body>
11     <div id="root">从这里开始</div>
12 </body>
13 </html>

```

`/static/js/bootstrap.bundle.min.js` 表示站点根目录下开始, <http://127.0.0.1:8000/static/js/bootstrap.bundle.min.js>, 找到返回200, 找不到返回404。

但是上面这种写法是固定死的, 可以使用`url_for`处理。

`url_for`第一个参数写函数, 通过函数对象, 反查路径映射中的路径。

```

1 <script src="{{ url_for('static', filename='js/jquery-3.6.0.min.js') }}">
</script>

```

上面这一句的意思就是通过`static`反查路径, `<Rule '/static/<filename>' (OPTIONS, GET, HEAD) -> static>]`

在Flask中, 可以指定静态目录。

```

1 app = Flask('web', static_folder='static', static_url_path='/st')

```

- `static_url_path`, 指的是url中访问路径, 可以不写, 默认就是 `/<static_folder>`, 这个路径出现在URL中, 是一个逻辑的路径
- `static_folder`, 指的是静态文件的物理放置目录。注意这个目录不要写错。如若没有配置, 默认`static`。如果没有配置`static_url_path`, 默认就是 `/static`

基本语法

`$(selector).action()`

- 美元符号定义 jQuery
- 选择符 (selector) “查询”和“查找” HTML 元素
- jQuery 的 `action()` 执行对元素的操作

文档就绪函数

为了防止文档加载完之前就允许执行jQuery代码, 需要一个网页加载完的才调用的函数。

```

1 $(document).ready(function(){
2   /* jQuery functions go here */
3 });

```

可以简写为下面形式

```

1 $(function(){
2   /* jQuery functions go here 准备就绪回调函数 */
3 });

```

jQuery 元素选择器

jQuery 使用 CSS 选择器来选取 HTML 元素。

选择器	含义
\$("p")	选取 <p> 元素
\$("p.intro")	选取所有 class="intro" 的 <p> 元素
\$("p#demo")	选取所有 id="demo" 的 <p> 元素

jQuery 属性选择器

jQuery 使用 XPath 表达式来选择带有给定属性的元素。

选择器	含义
\$("[href]")	选取所有带有 href 属性的元素
\$("[href='#]")	选取所有带有 href 值等于 "#" 的元素
\$("[href!='#]")	选取所有带有 href 值不等于 "#" 的元素
\$("[href\$.jpg]")	选取所有 href 值以 ".jpg" 结尾的元素

元素包装

```

1 console.log($('#root')); // 返回jQuery 对象
2 console.log(document.getElementById('root')); // HTML元素
3 console.log($(document.getElementById('root'))) // 包装成jQuery对象
4 console.log($('div')) // div标签选择器，注意和下面的不同之处
5 console.log($('<div />')) // HTML元素包装成jQuery对象

```

CSS操作

```

1 var r = $('#root') // 选择
2 console.log(r.css('color')); // 读
3 r.css('color', '#F00'); // 写
4 r.css({'background-color': '#f2f2f2', 'color': '#00F'});

```

HTML操作

```
1 var r = $('#root');
2 console.log(r.html()); // 读
3 r.html('<span>'+ r.html() + '</span>'); // 写
4 console.log(r.html()); // 读
5 console.log(r.text()); // 只有文本
```

增加节点

内部追加

```
1 var r = $('#root');
2 r.append('<img />');
```

增加兄弟

after外部增加兄弟, append内部末尾增加

```
1 var r = $('#root');
2 r.append('<img />');
3 r.after('<hr>');
4 r.append('<ul><li></li><li></li><li></li></ul>');
```

attr属性操作

```
1 var r = $('#root');
2 r.append('<img />');
3 r.after('<hr>');
4 r.append('<ul><li>行1</li><li>行2</li><li>行3</li></ul>');
5 $('#root img').attr({
6     height:60, width:120, alt:'magedu logo',
7     src:'http://www.magedu.com/wp-
8     content/uploads/2018/12/2018122312035677.png'
9 });
10 console.log($('#root img').attr('alt'));
```

removeAttr是移除属性函数。

第几个

:odd奇数, :even偶数, :first第一个

```
1 $('#root ul li:first').css('color', '#f00'); //
2 $('#root ul li').first().css('color', '#f00');
3 $('#root ul li:nth-child(2)').css('color', '#0f0'); // 匹配li的父ul下的li的第几个
```

:odd 或 :even 奇偶数, 从0开始计数

jQuery事件函数

为jQuery对象增加事件回调函数

```
1 <!DOCTYPE html>
2 <html lang="zh-CN">
3
4 <head>
5     <meta charset="UTF-8">
6     <script src="{{ url_for('static', filename='js/jquery-3.6.0.min.js') }}"></script>
7     <title>马哥教育jQuery测试</title>
8 </head>
9
10 <body>
11     <div id="root">
12         <h2>测试事件</h2>
13         <div><button class="testclick">点击事件</button></div>
14         <div class="console" style="border: 1px solid #000;"></div>
15     </div>
16 </body>
17
18 </html>
19 <script>
20     $(function () {
21         $('button.testclick').click(function (event) {
22             console.log(event.target);
23             var c = $('<div>hello</div>'); // 包装成jquery对象
24             c.css({ 'background-color': '#F0F0F0', 'margin': '10px' });
25             $('.console').append(c); // 使用jquery对象
26         });
27     });
28
29 </script>
```

注意

```
1 $('button.testclick').click() //调用click事件回调函数
2 $('button.testclick').click(function(){} ) // 设置click事件回调函数
```