

问答

1.说说库和框架的区别？

库：库提供了方法，即API。当开发者引入了库以后，可以直接调用并使用这里面的API。可以当做小工具使用。它的体积较小。常见的库有jQuery.js，three.js等。

框架：顾名思义，框架就是有一个支撑结构。相当于一个已经构建好的实体房子，而用户只需在这个房子里添加自己想要的家具等。而框架对于开发者，开发者只需在这个框架中添加自己的东西而实现具体的功能。常见的框架有Angular, React,Vue等。

2.jquery 能做什么？

jquery是一个丰富的js库，内部对js的很多复杂的方法进行了封装和加工，比如js很多方法名很冗长，jquery封装后方法就变得简便了，也考虑到了跨浏览器兼容问题，这样我们用js时需要考虑的很多问题它都帮我们解决了，这样我们使用时就比js代码要方便，高效的多。

3.jquery 对象和 DOM 原生对象有什么区别？如何转化？

jquery 对象和 DOM 原生对象的区别

- DOM对象是通过原生JavaScript获得的对象，jQuery对象是通过jQuery选择器获得的对象；
- jQuery对象无法使用DOM对象的任何方法，同理DOM对象也不能使用jQuery里的方法。

jQuery对象转成DOM原生对象——两种转换方式：`[index]` 和 `.get(index)`

HTML如下:

```
<div id="ct">
<div class="item">item1</div>
<div class="item">item2</div>
<div class="item">item3</div>
</div>
```

Jquery方法如下:

```
$('.item')[0];//方法1,<div class="item">item1</div>
$('.item').get(2);//方法2,<div class="item">item3</div>
```

原生对象转成jQuery对象:使用\$()将DOM对象包裹起来

```
var ct = document.getElementById('ct');//DOM对象
```

```
var $ct = $(ct); // $(DOM对象) 转化为了jQuery对象$ct
```

4.jquery中如何绑定事件？

bind、unbind、delegate、live、on、off 都有什么作用？推荐使用哪种？使用on绑定事件使用事件代理的写法？

1.最简单常见的是 **bind (type,handle)** ,这种方式的写法和 **type (handle)** 的写法的基本作用是相同的(如 **\$("a").click(function(){}))**),对应的取消事件的绑定方法是 **unbind** : 如果不带参数默认移除所有绑定的事件监听,若带参数则移除指定的事件监听

(如 **\$("p").unbind("click")**),这种是直接在符合条件的元素的事件类型上绑定监听函数,因此每生成一个新的元素,就需要给新元素绑定一次事件,比较麻烦。

2.于是有了 **live** 这种方式 (如 **\$("a").live(type,function(){}))**),这种方式实现了事件的代理,把监听函数绑定在 **\$(document)** 对象上,每次有事件触发时,通过事件冒泡, **\$(document)** 会首先接收到,并且判断target是否是目标元素 (如上例中 **\$("a")**), 事件类型是否是该类型 (**type**), 若都符合条件会执行该监听函数,解决了 **bind** 的弊端,但是又有新的缺点,如果dom树结构比较深,点击的目标需要通过层层遍历经历事件冒泡到达顶端,性能低,开销大,在老版本的jQuery中已经被废弃。

3. **delegate** ,实现指定代理对象 (如 **\$(".div1").delegate("a", type,function(){}))**),这样代理方从 **\$(document)** 对象变成了一个更具体的 **\$(".div1")** 对象,减少了查找范围,缩短了冒泡旅程,比live更精确,效率高。

4. **on** 这种方式把以上三种方法做了统一 (如 **\$(".div1").on(type, "a", ,function(){}))** ,注意第二个参数是子元素选择标签),对应的off是移除事件绑定的方式,在jquery的新的版本上都是支持的,也是现在比较推荐的写法。

5.jquery 如何展示/隐藏元素？

.hide(speed, callback) : 隐藏元素

.show(speed, callback) : 显示元素

.toggle(speed, callback) : 显示隐藏的元素,隐藏显示的元素

可选的 **speed** 参数规定隐藏/显示的速度,可以取以下值: " slow" 、 " fast" 或毫秒数; 可选的 **callback** 参数是隐藏或显示完成后所执行的函数名称。

展示元素：

使用show()

```
$('#btn').on('click', function() {
    $('#div1').show(1000); // 1000表示以1000ms的速度显示div
});
使用fadeIn()

$('#btn').on('click', function() {
```

```
$('.div1').fadeIn(1000);  
});
```

使用slideDown()

```
$('#btn').on('click', function() {  
    $('.div1').slideDown(1000);  
});
```

隐藏元素：

使用hide()

```
$('#btn').on('click', function() {  
    $('.div1').hide(1000);  
});  
使用fadeOut()
```

```
$('#btn').on('click', function() {  
    $('.div1').fadeOut(1000);  
});  
使用slideUp()
```

```
$('#btn').on('click', function() {  
    $('.div1').slideUp(1000);  
});
```

除此之外还可以通过 `addClass()/removeClass()` 控制CSS来展示/隐藏元素.

6.jquery 动画如何使用？

`animate()`方法用于创建自定义动画。

```
.animate({params}, speed, callback);
```

params 定义形成动画的 CSS 属性(动画结束后的CSS)必选

speed 规定效果的时长.值:slow,fast或毫秒,可选

callback 动画完成后所执行的函数,可选

.delay() 设置一个定时器，以延迟 执行队列中后续的项目。

.clearQueue()从队列中删除所有还没有运行过的项目。

.stop()在匹配的元素上停止当前正在运行的动画。

.finish()针对匹配的元素 停止当前 正在运行的动画，删除所有 队列中的动画，并结束所有动画

7.如何设置和获取元素内部 HTML 内容？如何设置和获取元素内部文本？

`$(selector).html()` 设置html内容

`$(selector).text()` 设置文本内容

如果括号为空则获取, 有值则设置

8.如何设置和获取表单用户输入或者选择的内容？如何设置和获取元素属性？

使用Jquery选择器选择input元素后，调用 `.val()` 即可获取输入的内容；在 `.val()` 中传递参数即可设置input的内容。

```
$(selector).val()
```

使用JQuery选择器选择元素后，调用 `attr(attribute,value)` 即可获取元素置顶属性的值，`attribute` 参数为必填，代表了想要定位的属性，而 `value` 参数选填，未填为查找，填写后代表修改为 将属性值改为 `value`

```
$(selector).attr(attribute,value)
```

代码

1.实现如下效果

这一题的核心思想是利用.show()与.hide()

2.实现如下效果

这一题的核心思想是增加与删除元素身上的active与hover类

3.实现如下效果

这一题的核心思想是封装一个输出新增内容的函数



Leanote
Upgrade Account