

课程回顾

1 标准DOM*

```
1  **** 获取节点: getElementById()  byName()  byClassName() byTagName()
2
3  **** 新建: createElement() createTextNode
4      追加   父.appendChild()  父.insertBefore()
5      替换   父.replaceChild()
6      删除   父.removeChild()
7      克隆   要克隆的节点.cloneNode(bool:deep):
8              默认: false 表示只克隆本身, 不可克隆其后代
9              true: 克隆自己及其后代
```

2 innerText和innerHTML

```
1  innerText: 获取双标签的文本节点
2  innerHTML: 模拟网页节点追加、新增。。。
```

3 BOM

```
1  window:
2      弹框方法:
3          alert()  prompt()  confirm(): 弹出一个确认框: 确定和取消
4      定时器:
5          setInterval()  clearInterval()
6
7          setTimeout()  clearTimeout()
8
9      窗口开启和关闭:
10         open(string url, "新窗口名称", "新窗口特征")
11         close():
```

课程目标

1 location和history ==== 掌握

2 JQuery概念

3 JQuery常用选择器 ==== 掌握

课程实施

1 window方法

setTimeout应用案例

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title></title>
6   </head>
7   <script type="text/javascript">
8     function sayHi(){
9       alert("hello");
10      //推荐使用setTimeout模拟定时器
11      setTimeout(sayHi,5000); //间隔5m后执行sayHi,然后过期
12    }
13    // setInterval(sayHi,5000); //每隔5m就执行一次sayHi()
14    sayHi();
15  </script>
16  <body>
17  </body>
18 </html>
19
```

面试题：setInterval()和setTimeout()区别

- 1 setInterval(): 间隔指定的时间后重复执行函数代码。即：函数会重复执行
- 2 setTimeout(): 间隔指定时间后执行一次函数代码，然后过期，即：不再执行代码了

时钟效果的两种实现方式

- setInterval()和clearInterval()

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title></title>
6   </head>
7   <script type="text/javascript">
8     function showNow(){
9       document.getElementById("div1").innerText=new
Date().toLocaleString();
10    }
11
12    var id; //保存定时的编号
13    function start(){
14      showNow(); //开始按钮一点击，就显示当前时间
15      id=setInterval(showNow,1000);
16    }
17
18    function stop(){
19      // clearInterval(停止的intervalID);
20      clearInterval(id);
21    }
22  </script>
23  <body>
```

```

24     <div id="div1">
25
26     </div>
27     <button onclick="start()">开始</button>
28     <button onclick="stop()">停止</button>
29 </body>
30 </html>

```

- setTimeout()和clearTimeout()

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8" />
5          <title></title>
6      </head>
7      <script type="text/javascript">
8          function showNow(){
9              document.getElementById("div1").innerText=new
Date().toLocaleString();
10             id=setTimeout(showNow,1000);//间隔1s，执行一次showNow()
11         }
12
13         var id;//保存定时的编号
14         function start(){
15             showNow();//开始按钮一点击，就显示当前时间
16         }
17
18         function stop(){
19             //      clearTimeout(要停止的timeoutID);
20             clearTimeout(id);
21         }
22     </script>
23     <body>
24         <div id="div1">
25
26         </div>
27         <button onclick="start()">开始</button>
28         <button onclick="stop()">停止</button>
29     </body>
30 </html>

```

2 location对象

Location 对象属性

属性	描述	IE	F	O
hash	设置或返回从井号 (#) 开始的 URL (锚)。	4	1	9
host	设置或返回主机名和当前 URL 的端口号。	4	1	9
hostname	设置或返回当前 URL 的主机名。	4	1	9
href	设置或返回完整的 URL。	4	1	9
pathname	设置或返回当前 URL 的路径部分。	4	1	9
port	设置或返回当前 URL 的端口号。	4	1	9
protocol	设置或返回当前 URL 的协议。	4	1	9
search	设置或返回从问号 (?) 开始的 URL (查询部分)。	4	1	9

Location 对象方法

属性	描述	IE	F	O
assign()	加载新的文档。	4	1	9
reload()	重新加载当前文档。	4	1	9
replace()	用新的文档替换当前文档。	4	1	9

课堂案例

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8" />
5          <title></title>
6      </head>
7      <script type="text/javascript">
8          function openBaiDu(){
9              //href设置浏览器地址栏访问的网址
10             location.href="http://www.baidu.com";//获取当前浏览器窗口的url 设置
url地址
11         }
12         function openJingDong(){
13             // location.replace(新网页的地址);//替换
14             //用新的url地址替换当前url
15             // location.replace("http://www.jd.com:80");
16             //设置url的新地址
17             location.assign("http://www.jd.com:80");//加载新的文档。
18         }
19         function showTime(){
20             document.getElementById("div1").innerText="当前时间: "+new
Date().toLocaleString();
21         }
22     </script>
23     <body onload="showTime()">
24     <!--
25         location
```

```

26     -->
27     <div id="div1">
28         当前时间:
29     </div>
30     <button onclick="openBaiDu()">打开百度</button>
31     <button onclick="openJingDong()">打开京东</button>
32     <button onclick="location.reload()">刷新网页</button>
33 </body>
34 </html>

```

小结：打开新网页的实现方式

```

1  1. window.open("url","", "窗口特征")
2  2. location.href=url;
3  3. location.assign(url);
4  4. location.replace(url);

```

3 history对象

History 对象包含用户（在浏览器窗口中）访问过的 URL。

3-1 常用方法

```

1  go()
2  back()
3  forward()

```

3-2 课堂案例

- 1.html

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <title>1.html</title>
6      </head>
7      <body>
8          图书管理系统首页
9          <a href="2.html">新增</a>
10         <a href="3.html">修改</a>
11
12         <button onclick="history.forward()">前进</button>
13         <!--
14             go(1):等价于forward()
15         -->
16         <button onclick="history.go(1)">前进修改页</button>
17     </body>
18 </html>

```

- 2.html

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title></title>
6   </head>
7   <body>
8     新增图书信息
9     <button onclick="history.back()">后退</button>
10  </body>
11 </html>

```

- 3.html

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title></title>
6   </head>
7   <body>
8     修改图书信息
9     <!--
10      go(-1):等价于back()
11
12      -->
13     <button onclick="history.go(-1)">后退</button>
14   </body>
15 </html>

```

4 补充：课后作业大侠名称显示

- 1.html

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title></title>
6   </head>
7   <script type="text/javascript">
8     // window是一个对象
9     //定义全局变量，都归属于window对象
10    var input;
11    function showwindow(){
12      //input是用户输入的数据，如果用户没有输入任何数据，就是默认值“大侠”
13      input=prompt("大侠，请留下您的大名：", "大侠");
14      //打开新窗口
15      // window.open("info.html");//open()打开新窗口，在新窗口通过opener属性获
      取当前窗口的window对象
16      // location.href="info.html";
17    }
18  </script>
19  <body>

```

```

20
21     <button onclick="showwindow()">click me</button>
22 </body>
23 </html>

```

- 2.html

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <title></title>
6      </head>
7      <script type="text/javascript">
8          window.onload=function(){
9              //window.open() 打开新窗口,
10             var objwindow=window.opener;//获取是哪个窗口打开了当前窗口
11             //div1里面显示大侠的名称
12             document.getElementById("div1").innerText="大侠，你的名字
是："+objwindow.input;
13         }
14     </script>
15     <body>
16         <div id="div1">
17
18         </div>
19     </body>
20 </html>

```

5 JQuery概述

为了简化 JavaScript 的开发, 一些 JavaScript 库诞生了。JavaScript 库封装了很多预定义的对象和实用函数。能帮助使用者建立有难度交互的 Web2.0 特性的富客户端页面, 并且兼容各大浏览器。Jquery就是这些封装的类库之一。

5-1 JQuery的入门案例HelloWorld的开发步骤

- 1 • jquery的官方网站上面下载jquery提供的js类库文件
- 2 • 自己的项目引入jquery的js文件
- 3 <script src="jqueryjs文件url"></script>
- 4 • 按照jquery的语法格式, 书写代码即可

参考代码

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <title></title>
6          <!--引入jquery的类库文件
7
8              min:压缩版js, 阅读性不好
9              -->
10         <script type="text/javascript" src="js/jquery-3.1.1.min.js" >
</script>

```

```

11     </head>
12     <script type="text/javascript">
13         //$(function);
14         $(function(){
15             alert("helloworld!");
16         });
17     </script>
18     <body>
19     </body>
20 </html>
21

```

6 JQueryの\$详解

	window.onload	\$(document).ready()
执行时机	必须等待网页中所有的内容加载完毕后（包括图片、flash、视频等）才能执行	网页中所有DOM文档结构绘制完毕后即刻执行，可能与DOM元素关联的内容（图片、flash、视频等）并没有加载完
编写个数	同一页面不能同时编写多个	同一页面能同时编写多个
简化写法	无	\$(function){ //执行代码 };

6-1 \$和window.onload的区别

- window.onload绑定函数只能有一次，且以最后一次赋值为主，但是\$()可以绑定多个函数

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <title></title>
6          <!--引入jquery的类库文件
7
8              min:压缩版js，阅读性不好
9              -->
10         <script type="text/javascript" src="js/jquery-3.1.1.min.js" >
11             <!--//浏览器不识别的-->
12         </script>
13     </head>
14
15     <script type="text/javascript">
16         //$(function): JQuery() 等价于window.onload
17         $(function(){
18             alert("helloworld!");//弹出helloworld
19         });
20
21         $(function(){
22             alert("你好Jquery!");//弹出你好JQuery
23         });
24     </script>
25     <body>
26     </body>
27 </html>
28
29

```



```

30 <!DOCTYPE html>
31 <html>
32   <head>
33     <meta charset="UTF-8">
34     <title></title>
35   </head>
36
37   <script type="text/javascript">
38     window.onload=function(){
39       alert("helloWorld");//不弹出
40     }
41
42     window.onload=function(){
43       alert("你好jQuery");//弹出 你好jQuery
44     }
45   </script>
46   <body>
47   </body>
48 </html>

```

- window.onload没有简化方式，但是\$()有很多书写方式，更灵活

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="UTF-8">
5      <title></title>
6      <!--引入jquery的类库文件
7
8      min:压缩版js，阅读性不好
9      -->
10     <script type="text/javascript" src="js/jquery-3.1.1.min.js" >
11       <!--//浏览器不识别的-->
12     </script>
13   </head>
14
15   <script type="text/javascript">
16     //jQuery需要对上述代码进行封装
17     $(document).ready(function(){
18       alert("helloWorld");
19     });
20     //简化版本一：
21     $().ready(function(){
22       alert("helloWorld");
23     });
24     //最简化，也是最常用的格式
25     $(function(){
26       alert("helloWorld");
27     });
28   </script>
29   <body>
30   </body>
31 </html>
32
33
34 <!DOCTYPE html>
35 <html>

```

```

36     <head>
37         <meta charset="UTF-8">
38         <title></title>
39     </head>
40     <script type="text/javascript">
41         //没有其他的书写方式
42         window.onload=function(){
43
44         }
45     </script>
46     <body>
47     </body>
48 </html>

```

✖ Uncaught ReferenceError: \$ is not defined
at [掌握]helloworld.html:17:3

没有导入外部的jquery的js

7 选择器

获取html网页上一个标签的方法。等价于getElementBy。。。

7-1 选择器语法

```

1  $("#id值"): 等价于getElementById
2  $(".class值"): 等价于getElementsByClassName
3  $("p"): 等价于getElementsByTagName("p")
4  $("*"):

```

7-2 基本选择器

```

1  ID选择器: $("#id值")
2  class类选择器: $(".class值")
3  标签选择器: $("HTML标签")
4  通配符选择器: $("*")

```

课堂案例

- ID选择器

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <title></title>
6          <!--引入jquery的类库文件-->
7          <script type="text/javascript" src="js/jquery-3.1.1.min.js" >
8      </script>
9      </head>

```

```

9      <script type="text/javascript">
10      $(function(){//当document.ready开始执行内部代码
11          //处理自己的业务需求
12          //1.获取id=div1的标签，内部的内容
13          //1-1 获取div1对应的对象（即div）
14          var objDiv=$('#div1');//jQuery对象
15          //1-2 获取div里面的文字内容
16          var strText=objDiv.text();//innerText学习js的dom补充属性
17          //jquery获取innerText对应方法text()  innerHTML对应方法html()
18          //1-3 弹框显示
19          alert(strText);
20      });
21      </script>
22      <body>
23          <div id="div1">Hello JQuery</div>
24          <span class="s1">你好，JQuery</span>
25      </body>
26  </html>
27

```

- class选择器

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <title></title>
6          <!--引入jquery的类库文件-->
7          <script type="text/javascript" src="js/jquery-3.1.1.min.js" >
8      </script>
9      </head>
10     <script type="text/javascript">
11         //class选择器
12         $(function(){
13             //1.获取对象 span class=s1
14             var objJQspan=$(".s1");
15             //2.获取span之间的文本
16             var str=objJQspan.text();
17             alert(str);
18         // console.log(str);
19     });
20     </script>
21     <body>
22         <div id="div1">Hello JQuery</div>
23         <span class="s1">你好，JQuery</span>
24     </body>
25 </html>

```

7-3 基本选择器的使用案例：全选

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <title></title>

```

```

6      <!--引入jquery的类库文件-->
7      <script type="text/javascript" src="js/jquery-3.1.1.min.js" >
</script>
8      </head>
9
10     <script type="text/javascript">
11         $(function(){
12             //给全选添加一个click的监听器
13             //jquery如何处理事件监听器 所有的on事件=function() 改成 选择器.事件
            (function(){})(
14                 $("#chkAll").click(function(){
15                     //实现全选
16                     /**
17                      * 1. 获取所有的要选中的复选框
18                      * 2. 遍历设置每一个复选框，checked属性与全选一致
19                      */
20                     var arrJQ=$(".subject");
21                     //jquery中循环的方式：each()，等价于foreach()
22                     /*for(var i=0;i<arrJQ.length;i++){
23                         arrJQ[i].checked=this.checked;//this是js对象
24                     }*/
25                     //each()实现数组的循环
26                     $(arrJQ).each(function(index,objDom){
27                         //this不是事件源，不能再同事件注册的对象使用
28                         //设置每个元素选中的状态
29                         //objDom.checked=$("#chkAll")[0].checked;
30                         //1. 获取全选的值得
31                         var v1=$("#chkAll").prop("checked");
32                         //2. 设置全选
33                         $(objDom).prop("checked",v1);
34                     });
35                 });//调用或设置方法
36             });
37     </script>
38     <body>
39         <input type="checkbox" id="chkAll"/>全选
40         <input type="button" id="btnCheck" value="反选" />
41         <br />
42         <input type="checkbox" class="subject" />java
43         <input type="checkbox" class="subject"/>html
44         <input type="checkbox" class="subject"/>sql
45         <input type="checkbox" class="subject"/>vue
46         <input type="checkbox" class="subject"/>spring
47         <input type="checkbox" class="subject"/>springboot
48         <input type="checkbox" class="subject"/>springmvc
49     </body>
50 </html>

```

8 JQuery对象和JavaScript对象互相转换

- JQuery-->js对象（也称为dom对象）

```

1 $(选择器)[下标]
2 $(选择器).get(下标)
3 举例:
4 var objJQDiv=$("#div1");
5 //jquery--->js
6 var objDomDiv=objJQDiv[0]
7 objDomDiv.innerHTML;//正确

```

- js对象-->Jquery对象

```

1 var objJQ=$(js对象)

```

8-2 课堂案例

- jquery-->dom 案例

```

1 <!DOCTYPE html>
2 <html>
3     <head>
4         <meta charset="UTF-8">
5         <title></title>
6         <!--引入jquery的类库文件
7
8             min:压缩版js, 阅读性不好
9             -->
10        <script type="text/javascript" src="js/jquery-3.1.1.min.js" >
11    </script>
12    </head>
13    <script type="text/javascript">
14        $(function(){//当document.ready开始执行内部代码
15            //处理自己的业务需求
16            //1.获取id=div1的标签, 内部的内容
17            //1-1 获取div1对应的对象 (即div)
18            var objDiv=$("#div1");//JQuery对象
19            //1-2 获取div里面的文字内容 常见程序问题: jquery对象和js属性混用, 不支持
20            的!!!
21            //jquery对象和js对象互相转换,
22            var strText=objDiv[0].innerHTML;//innerHTML学习js的dom补充属性
23            //jquery获取innerHTML对应方法text() innerHTML对应方法html()
24            //1-3 弹框显示
25            alert(strText);
26        });
27    </script>
28    <body>
29        <div id="div1">Hello JQuery</div>
30        <span class="s1">你好, JQuery</span>
31    </body>
32 </html>
33

```

- dom-->Jquery 案例

```

1 <!DOCTYPE html>
2 <html>
3     <head>

```

```

4      <meta charset="UTF-8">
5      <title></title>
6      <!--引入jquery的类库文件
7
8          min:压缩版js, 阅读性不好
9      -->
10     <script type="text/javascript" src="js/jquery-3.1.1.min.js" >
</script>
11 </head>
12 <script type="text/javascript">
13     $(function(){//当document.ready开始执行内部代码
14         //处理自己的业务需求
15         //1.获取id=div1的标签, 内部的内容
16         //1-1 获取div1对应的对象 (即div)
17         var objDomDiv=document.getElementById("div1");
18         //1-2 获取div里面的文字内容 常见程序问题: jquery对象和js属性混用, 不支持
    的!!!
19         //jquery对象和js对象互相转换,
20         var strText=$(objDomDiv).text();//innerText学习js的dom补充属性
21         //jquery获取innerText对应方法text() innerHTML对应方法html()
22         //1-3 弹框显示
23         alert(strText);
24     });
25 </script>
26 <body>
27     <div id="div1">Hello JQuery</div>
28     <span class="s1">你好, JQuery</span>
29 </body>
30 </html>

```

课程总结

1 window(setInterval setTimeout) location (打开新网页) history常用方法

2 JQuery是一个大佬提供的外部js

3 JQuery写代码的步骤, 基本骨架

4 基本选择器, each() 事件监听器注册方式 prop()

预习安排

mysql5.7+

解压版配置

