## Day10\_javaScript高级

### 0.1学习目标

1、能够掌握js 的Bom对象

2、能够掌握j s的Dom对象

3、独立完成四个案例

### 0.2 学习指南

* Js的Bom对象（重要）
* Js的Dom对象（重点）
* 四个案例(重要)

### 第一堂课

本节知识点

1、今日内容

2、DOM简单学习

3、事件简单学习

4、案例1\_电灯开关

本节目标

了解什么是Dom和事件

#### 今日内容

##### 1.1.1知识概述

今天主要是js高级

##### 1.1.2视频详情



##### 1.1.3总结与补充

无

##### 1.1.4课堂提问与练习

##### 1.1.5习题答案

##### 1.1.6视频缺陷

无

##### 1.1.7视频扩展

无

#### DOM简单学习

##### 1.2.1知识概述

 Dom的功能。

##### 1.2.2视频详情



##### 1.2.3总结与补充

##### 1.2.4课堂提问与练习

* Dom的功能？

##### 1.2.5习题答案

* 控制html文档的内容
* 获取页面标签(元素)对象：Element

##### 1.2.6视频缺陷

无

##### 1.2.7视频扩展

无

#### 事件简单学习

##### 1.3.1知识概述

js中事件比如点击,敲打键盘都是事件等,用途就是触发函数的

##### 1.3.2视频详情



##### 1.3.3总结与补充

##### 1.3.4课堂提问与练习

##### 1.3.5习题答案

##### 1.3.6视频缺陷

无

##### 1.3.7视频扩展

无

#### 案例1\_电灯开关

##### 1.4.1知识概述

通过dom对象,来切换图片,用两张图片表示开和关

##### 1.4.2视频详情



##### 1.4.3总结与补充

##### 1.4.4课堂提问与练习

##### 1.4.5习题答案

##### 1.4.6视频缺陷

无

##### 1.4.7视频扩展

无

### 第二堂课

本节知识点：

1、BOM\_概述

2、BOM\_Window\_弹出方法

3、BOM\_Window\_打开关闭方法

4、BOM\_Window\_定时器方法

本节目标 ：

window对象常用的方法

#### BOM\_概述

##### 2.1.1知识概述

bom指的就是浏览器对象,来操作浏览器的

##### 2.1.2视频详情



##### 2.1.3总结与补充

##### 2.1.4课堂提问与练习

* 什么是BOM?

##### 2.1.5习题答案

1. 概念：Browser Object Model 浏览器对象模型 \* 将浏览器的各个组成部分封装成对象。
2. 组成：

\* Window：窗口对象

\* Navigator：浏览器对象

\* Screen：显示器屏幕对象

\* History：历史记录对象

\* Location：地址栏对象

##### 2.1.6视频缺陷

无

##### 1.1.7视频扩展

无

#### BOM\_Window\_弹出方法

##### 2.2.1知识概述

window对象常见的弹出框有警告框, 输入框和确认框

##### 2.2.2视频详情



##### 2.2.3总结与补充

##### 2.2.4课堂提问与练习

##### 2.2.5习题答案

##### 2.2.6视频缺陷

无

##### 2.2.7视频扩展

无

#### BOM\_Window\_打开关闭方法

##### 2.3.1知识概述

window对象打开和关闭窗口的方法也很常见,一个是open,一个是close

##### 2.3.2视频详情



##### 2.3.3总结与补充

##### 2.3.4课堂提问与练习

无

##### 2.3.5习题答案

无

##### 2.3.6视频缺陷

无

##### 2.3.7视频扩展

无

#### BOM\_Window\_定时器方法

##### 2.4.1知识概述

window对象的定时器 ,通过定时来间隔时间调用函数

##### 2.4.2视频详情



##### 2.4.3总结与补

##### 2.4.4课堂提问与练习

无

##### 2.4.5习题答案

无

##### 2.4.6视频缺陷

无

##### 2.4.7视频扩展

无

### 第三堂课

本节知识点

1、案例2\_轮播图

2、BOM\_Window\_属性

3、BOM\_Location

4、案例3\_自动跳转首页

本节目标

熟练使用定时器和location属性

#### 案例2\_轮播图

##### 3.1.1知识概述

通过js的定时器,间隔时间来切换图片

##### 3.1.2视频详情



##### 3.1.3总结与补充

##### 3.1.4课堂提问与练习

* 练习轮播图

##### 3.1.5习题答案

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>轮播图</title>

</head>

<body>

<img id="img" src="img/banner\_1.jpg" width="100%">

<script>

/\*

分析：

1.在页面上使用img标签展示图片

2.定义一个方法，修改图片对象的src属性

3.定义一个定时器，每隔3秒调用方法一次。

\*/

//修改图片src属性

var number = 1;

function fun(){

number ++ ;

//判断number是否大于3

if(number > 3){

number = 1;

}

//获取img对象

var img = document.getElementById("img");

img.src = "img/banner\_"+number+".jpg";

}

//2.定义定时器

setInterval(fun,3000);

</script>

</body>

</html>

##### 3.1.6视频缺陷

无

##### 3.1.7视频扩展

无

#### BOM\_Window\_属性

##### 3.2.1知识概述

window对象的属性常见的有五个,基本都是操作浏览器的

##### 3.2.2视频详情



##### 3.2.3总结与补充

##### 3.2.4课堂提问与练习

##### 3.2.5习题答案

##### 3.2.6视频缺陷

无

##### 3.2.7视频扩展

无

#### BOM\_Location

##### 3.3.1知识概述

location翻译过来就是地址,作用获取地址或者设置地址

##### 3.3.2视频详情



##### 3.3.3总结与补充

##### 3.3.4课堂提问与练习

##### 3.3.5习题答案

##### 3.3.6视频缺陷

无

##### 3.3.7视频扩展

无

#### 案例3\_自动跳转首页

##### 3.4.1知识概述

通过location对象和定时器完成自动跳转页面的效果.

##### 3.4.2视频详情



##### 3.4.3总结与补充

##### 3.4.4课堂提问与练习

* 练习自动跳转页

##### 3.4.5习题答案

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>自动跳转</title>

<style>

p{

text-align: center;

}

span{

color:red;

}

</style>

</head>

<body>

<p>

<span id="time">5</span>秒之后，自动跳转到首页...

</p>

<script>

/\*

分析：

1.显示页面效果 <p>

2.倒计时读秒效果实现

2.1 定义一个方法，获取span标签，修改span标签体内容，时间--

2.2 定义一个定时器，1秒执行一次该方法

3.在方法中判断时间如果<= 0 ，则跳转到首页

\*/

// 2.倒计时读秒效果实现

var second = 5;

var time = document.getElementById("time");

//定义一个方法，获取span标签，修改span标签体内容，时间--

function showTime(){

second -- ;

//判断时间如果<= 0 ，则跳转到首页

if(second <= 0){

//跳转到首页

location.href = "https://www.baidu.com";

}

time.innerHTML = second +"";

}

//设置定时器，1秒执行一次该方法

setInterval(showTime,1000);

</script>

</body>

</html>

##### 3.3.6视频缺陷

##### 3.3.7视频扩展

### 第四堂课

本节知识点：

1、BOM\_History

2、DOM\_概述

3、DOM\_Document对象\_获取Element方法

4、DOM\_Document对象\_创建DOM对象

本节目标 ：

掌握Dom对象的操作

#### BOM\_History

##### 4.1.1知识概述

history翻译过来就是历史,通过history可以完成浏览器地址栏的跳转效果

##### 4.1.2视频详情



##### 4.1.3总结与补充

##### 4.1.4课堂提问与练习

##### 4.1.5习题答案

##### 4.1.6视频缺陷

##### 4.1.7视频扩展

#### DOM\_概述

##### 4.2.1知识概述

Dom就是文档对象,用来操作文档的

##### 4.2.2视频详情



##### 4.2.3总结与补充

##### 4.2.4课堂提问与练习

* 什么是Dom

##### 4.2.5习题答案

Document Object Model 文档对象模型

\* 将标记语言文档的各个组成部分，封装为对象。可以使用这些对象，对标记语言文档进行CRUD的动态操作

##### 4.2.6视频缺陷

##### 4.2.7视频扩展

#### DOM\_Document对象\_获取Element方法

##### 4.3.1知识概述

在html文档里面,要想操作html标签,首先要获取标签对象,获取标签对象常用的三种方式.

##### 4.3.2视频详情



##### 4.3.3总结与补充

##### 4.3.4课堂提问与练习

* 练习获取Elment对象的方法

##### 4.2.5习题答案

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Document对象</title>

</head>

<body>

<div id="div1">div1</div>

<div id="div2">div2</div>

<div id="div3">div3</div>

<div class="cls1">div4</div>

<div class="cls1">div5</div>

<input type="text" name="username">

<script>

/\*

Document：文档对象

1. 创建(获取)：在html dom模型中可以使用window对象来获取

1. window.document

2. document

2. 方法：

1. 获取Element对象：

1. getElementById() ： 根据id属性值获取元素对象。id属性值一般唯一

2. getElementsByTagName()：根据元素名称获取元素对象们。返回值是一个数组

3. getElementsByClassName():根据Class属性值获取元素对象们。返回值是一个数组

4. getElementsByName(): 根据name属性值获取元素对象们。返回值是一个数组

2. 创建其他DOM对象：

createAttribute(name)

createComment()

createElement()

createTextNode()

3. 属性

\*/

//2.根据元素名称获取元素对象们。返回值是一个数组

var divs = document.getElementsByTagName("div");

//alert(divs.length);

//3.根据Class属性值获取元素对象们。返回值是一个数组

var div\_cls = document.getElementsByClassName("cls1");

// alert(div\_cls.length);

//4.根据name属性值获取元素对象们。返回值是一个数组

var ele\_username = document.getElementsByName("username");

//alert(ele\_username.length);

var table = document.createElement("table");

alert(table);

</script>

</body>

</html>

##### 4.2.6视频缺陷

##### 4.2.7视频扩展

无

#### DOM\_Document对象\_创建DOM对象

##### 4.4.1知识概述

Dom就是文档对象,用来操作文档的,那如何创建Dom呢,创建有四种方式,创建不同的Dom

##### 4.4.2视频详情



##### 4.4.3总结与补充

##### 4.4.4课堂提问与练习

##### 4.4.5习题答案

##### 4.4.6视频缺陷

##### 4.4.7视频扩展

### 第五堂课

本节知识点：

1、DOM\_Element对象

2、DOM\_Node对象

3、案例4\_动态表格\_添加

本节目标 ：

理解Node和Element的关系

#### DOM\_Element对象

##### 5.1.1知识概述

通过Element对象,来操作标签的属性,有两种方式,添加属性和移除属性.

##### 5.1.2视频详情



##### 5.1.3总结与补充

##### 5.1.4课堂提问与练习

* Element对象的操作属性的方法

##### 5.1.5习题答案

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Element对象</title>

</head>

<body>

<a>点我试一试</a>

<input type="button" id="btn\_set" value="设置属性">

<input type="button" id="btn\_remove" value="删除属性">

<script>

//获取btn

var btn\_set =document.getElementById("btn\_set");

btn\_set.onclick = function(){

//1.获取a标签

var element\_a = document.getElementsByTagName("a")[0];

element\_a.setAttribute("href","https://www.baidu.com");

}

//获取btn

var btn\_remove =document.getElementById("btn\_remove");

btn\_remove.onclick = function(){

//1.获取a标签

var element\_a = document.getElementsByTagName("a")[0];

element\_a.removeAttribute("href");

}

</script>

</body>

</html>

##### 5.1.6视频缺陷

##### 5.1.7视频扩展

#### DOM\_Node对象

##### 5.2.1知识概述

Node对象,又称节点对象,是所有对象的父对象,包括Dom对象.

##### 5.2.2视频详情



##### 5.2.3总结与补充

##### 5.2.4课堂提问与练习

##### 5.2.5习题答案

##### 5.2.6视频缺陷

无

##### 5.2.7视频扩展

无

#### 案例4\_动态表格\_添加

##### 5.3.1知识概述

 通过刚才所学的node对象等知识点,来完成动态添加表格的效果

##### 5.3.2视频详情



##### 5.3.3总结与补充

##### 5.3.4课堂提问与练习

##### 5.3.5习题答案

##### 5.3.6视频缺陷

无

##### 5.3.7视频扩展

无

### 第六堂课

本节知识点：

1. 案例4\_动态表格\_删除
2. HTMLDOM\_innerHTML
3. HTMLDOM\_样式控制

本节目标 ：

掌握innerHTML属性的使用

#### 案例4\_动态表格\_删除

##### 6.1.1知识概述

和上面讲的基本样,下面通过父标签来删除子标签

##### 6.1.2视频详情



##### 6.1.3总结与补充

##### 6.1.4课堂提问与练习

* 练习动态添加和删除表格

##### 6.1.5习题答案

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>动态表格</title>

<style>

table{

border: 1px solid;

margin: auto;

width: 500px;

}

td,th{

text-align: center;

border: 1px solid;

}

div{

text-align: center;

margin: 50px;

}

</style>

</head>

<body>

<div>

<input type="text" id="id" placeholder="请输入编号">

<input type="text" id="name" placeholder="请输入姓名">

<input type="text" id="gender" placeholder="请输入性别">

<input type="button" value="添加" id="btn\_add">

</div>

<table>

<caption>学生信息表</caption>

<tr>

<th>编号</th>

<th>姓名</th>

<th>性别</th>

<th>操作</th>

</tr>

<tr>

<td>1</td>

<td>令狐冲</td>

<td>男</td>

<td><a href="javascript:void(0);" onclick="delTr(this);">删除</a></td>

</tr>

<tr>

<td>2</td>

<td>任我行</td>

<td>男</td>

<td><a href="javascript:void(0);" onclick="delTr(this);">删除</a></td>

</tr>

<tr>

<td>3</td>

<td>岳不群</td>

<td>?</td>

<td><a href="javascript:void(0);" onclick="delTr(this);" >删除</a></td>

</tr>

</table>

<script>

/\*

分析：

1.添加：

1. 给添加按钮绑定单击事件

2. 获取文本框的内容

3. 创建td，设置td的文本为文本框的内容。

4. 创建tr

5. 将td添加到tr中

6. 获取table，将tr添加到table中

2.删除：

1.确定点击的是哪一个超链接

<a href="javascript:void(0);" onclick="delTr(this);" >删除</a>

2.怎么删除？

removeChild():通过父节点删除子节点

\*/

//1.获取按钮

/\* document.getElementById("btn\_add").onclick = function(){

//2.获取文本框的内容

var id = document.getElementById("id").value;

var name = document.getElementById("name").value;

var gender = document.getElementById("gender").value;

//3.创建td，赋值td的标签体

//id 的 td

var td\_id = document.createElement("td");

var text\_id = document.createTextNode(id);

td\_id.appendChild(text\_id);

//name 的 td

var td\_name = document.createElement("td");

var text\_name = document.createTextNode(name);

td\_name.appendChild(text\_name);

//gender 的 td

var td\_gender = document.createElement("td");

var text\_gender = document.createTextNode(gender);

td\_gender.appendChild(text\_gender);

//a标签的td

var td\_a = document.createElement("td");

var ele\_a = document.createElement("a");

ele\_a.setAttribute("href","javascript:void(0);");

ele\_a.setAttribute("onclick","delTr(this);");

var text\_a = document.createTextNode("删除");

ele\_a.appendChild(text\_a);

td\_a.appendChild(ele\_a);

//4.创建tr

var tr = document.createElement("tr");

//5.添加td到tr中

tr.appendChild(td\_id);

tr.appendChild(td\_name);

tr.appendChild(td\_gender);

tr.appendChild(td\_a);

//6.获取table

var table = document.getElementsByTagName("table")[0];

table.appendChild(tr);

}\*/

//使用innerHTML添加

document.getElementById("btn\_add").onclick = function() {

//2.获取文本框的内容

var id = document.getElementById("id").value;

var name = document.getElementById("name").value;

var gender = document.getElementById("gender").value;

//获取table

var table = document.getElementsByTagName("table")[0];

//追加一行

table.innerHTML += "<tr>\n" +

" <td>"+id+"</td>\n" +

" <td>"+name+"</td>\n" +

" <td>"+gender+"</td>\n" +

" <td><a href=\"javascript:void(0);\" onclick=\"delTr(this);\" >删除</a></td>\n" +

" </tr>";

}

//删除方法

function delTr(obj){

var table = obj.parentNode.parentNode.parentNode;

var tr = obj.parentNode.parentNode;

table.removeChild(tr);

}

</script>

</body>

</html>

##### 6.1.6视频缺陷

无

##### 6.1.7视频扩展

无

#### HTMLDOM\_innerHTML

##### 6.2.1知识概述

大家一定要明白innerHTML属性,这个属性只有块级标签才有,比如div和span

##### 6.2.2视频详情



##### 6.2.3总结与补充

##### 6.2.4课堂提问与练习

##### 6.2.5习题答案

##### 6.2.6视频缺陷

无

##### 6.2.7视频扩展

无

#### HTMLDOM\_样式控制

##### 6.3.1知识概述

通过操作Dom对象的style属性,来控制html里面标签的样式

##### 6.3.2视频详情



##### 6.3.3总结与补充

##### 6.3.4课堂提问与练习

##### 6.3.5习题答案

##### 6.3.6视频缺陷

##### 6.3.7视频扩展