《JavaWeb课程》教案

第3章 js

[1. 回顾上一章: [10分钟] 3](#_Toc489947987)

[2. 课程知识点讲解： 3](#_Toc489947988)

[2.1. 知识点1：js引入和变量[70分钟] 4](#_Toc489947989)

[2.2. 知识点2：函数[40分钟] 6](#_Toc489947990)

2.3 知识点3：DOM和常用事件[80分钟].......................................................................................................................9

2.4 知识点4：BOM(浏览器对象模型)和定时器，全局函数[60分钟]...........................................................................12

2.5 知识点5：js对象[80分钟].........................................................................................................................................15

2.5.1 小知识点1：js定义..............................................................................................................................................16

2.5.1 小知识点2：数组.................................................................................................................................................18

2.5.1 小知识点3：正则表达式.....................................................................................................................................19

[3. 本章总结 [20分钟] 2](#_Toc489947993)0

[4. 测试题 [30分钟] 2](#_Toc489947994)0

[5. 作业 [30分钟] 2](#_Toc489947995)0

[6. 扩展部分：](#_Toc489947996) 20

[7. 学员问题汇总： 2](#_Toc489947997)1

编创教师:

课时：420分钟

本章主要目标

掌握js简介： ECMAscript DOM BOM

熟练运用js的语法：数据类型、变量、函数

js的使用场景：打开网页查看

掌握js的ＤＯＭ操作及常用事件：

onload事件

onclick事件

onmouseover和onmouseout事件

onfocus、onblur和onsubmit事件

掌握BOM对象及引入方式和定时器

了解全局函数：

1.parseInt(String,radix):返回转换成整数的值。

2.parseFloat(string):返回转换成浮点型的值。

3.isFinite(value):检测某个是是否是无穷值。

4.isNaN(value):检测某个值是否是NaN。

5.encodeURI(uri):将字符串编码为URI。

6.decodeURI(uri):解码某个编码的URI。

7.encodeURIComponent(uri):将字符串编码为URI组件

8.decodeURIComponent():解码一个编码的URI组件

9.escape():对字符串进行编码

10.unescape():解码由escape()编码的字符串

11.eval():将JavaScript字符串当作脚本来执行

12.Number(object):把对象的值转换为数字

13.String():把对象的值转换为字符串

了解本地对象和内置对象：

Array

Boolean

Date

Math

Number

String

RegExp

Global

完成任务：

使用JS完成首页轮播效果

使用JS完成页面定时广告

使用JS完成表格高亮显示

使用JS完成表格的隔行换

使用JS完成注册表单校验

使用JS完成复选框的操作

使用JS完成省市联动效果

本章重点

重点1：变量声明，函数，事件和定时器的使用

重点2：理解js对象

本章难点

难点1：函数的理解

难点2：事件的运用

本章授课注意事项

1、本章采用由浅入深，逐层深入的的授课思路，教学过程中注意控制进度，避免出现前一阶段没有跟上，后一阶段只能放弃学习的情况

2、本章知识点讲解中主要使用先讲解理论再结合代码，并与之前学过的java做对比，帮助学员理解js语法和对js的运用。

# 回顾上一章: [10分钟]

作业点评和上章主要内容回顾

回顾html标签和css样式，并引入js。可以结合上网，浏览网页。分析哪些是html，哪些是css，哪些是js。

# 课程知识点讲解：

知识点1：[70分钟]

知识点1文字说明

What why where how

**JS的组成**：

ECMAScript：提供核心语言功能。

DOM：文档对象模型：提供访问和操作页面内容的方法和接口。

BOM：浏览器对象模型：提供与浏览器交互的方法和接口。

**数据类型：** 5 种简单数据类型（也称为基本数据类型）：Undefined、Null、Boolean、Number和 String。

还有 1种复杂数据类型——Object，Object 本质上是由一组无序的名值对组成的。

**变量：**使用关键字 var 。可以把变量看做存储数据的容器。

**函数：**匿名函数function （） { }

**引入JS**：1、把 JavaScript 插入到 HTML 页面中要使用<script>元素 2、引入外部JavaScript 文件

**ECMAScript 定义变量函数的语法**：

1、变量名 函数名 避免关键字和保留字

2、ECMAScript 中的一切（变量、函数名和操作符）都区分大小写。

变量名 test 和变量名 Test 分别表示两个不同的变量

3、标识符，就是指变量、函数、属性的名字，或者函数的参数。

标识符可以是按照下列格式规则组合起来的一或多个字符：

 **第一个字符**必须是一个字母、下划线（\_）或一个美元符号（$）；

 其他字符可以是字母、下划线、美元符号或数字。

标识符中的字母也可以包含扩展的 ASCII 或 Unicode 字母字符（如 À 和 Æ），但我们不推荐这样做。

按照惯例，ECMAScript 标识符采用**驼峰大小写格式**，也就是第一个字母小写，剩下的每个单词的首字母大写，

例如：firstSecond myCar doSomethingImportant

**JS注释**： 单行注释// 多行注释/\* \*/

typeof 操作符：

对一个值使用 typeof 操作符可能返回下列某个字符串：

 "undefined"——如果这个值未定义；

 "boolean"——如果这个值是布尔值；

 "string"——如果这个值是字符串；

 "number"——如果这个值是数值；

 "object"——如果这个值是对象或 null；

 "function"——如果这个值是函数。



**案例描述**：

任务描述

**案例代码**：（源文件名称）

1、先感受一下js代码：

<script>

document.write("<h1>这是一个标题</h1>");

</script>

2、ECMAScript 中的语句以一个分号结尾;

var num; // 声明变量

console.log(num); // undefined未定义

var num = 123; // 声明变量并赋值

console.log(num); // 123

3、定义不同的变量，在控制台打印。来理解不同类型的值

var a = true;

var b = "123";

var c = 123;

var d = null;

var e;

var f = function () {

}

console.log(typeof(a));

console.log(typeof(b));

console.log(typeof(c));

console.log(typeof(d));

console.log(typeof(e));

console.log(typeof(f));

console.log(typeof(g));

**案例说明**：

1）案例说明：简单的代码体验一下。

2）案例说明：理解声明变量。例如：我们有一个盒子，盒子里我们可以放任意的东西，当我们需要这个东西时就回去这个盒子里去取。这个盒子我们给他命名，通过名字来找对应的盒子。如果盒子里是空的，那得到的就是undefined未定义。Null是一个空对象指针，本质是一个对象。typeof 是一个操作符而不是函数。有可能会和函数相混淆。

3）案例说明：通过typeof操作符来查看对应的数据类型。定义方法部分还会再细讲。



提问或设问

1）问题1：变量声明的关键字是？

2）问题2: 标识符的命名规则？



1）关于本知识点的注意事项1：避免使用关键字和保留字声明变量。

2）关于本知识点的注意事项2：变量名区分大小写



练习1：

参考代码：将刚才的自己设置变量的值，进行尝试

知识点2：[40分钟]

### 知识点2

知识点1文字说明：函数：匿名函数、有名函数、函数的调用、函数的实参、形参、带有返回值的函数

**函数**是由事件驱动的或者当它被调用时执行的可重复使用的代码块。

// 自己不会主动执行，在调用的时候才会执行。调用函数使用函数名 + 小括号()。

// 方法的执行顺序：不是写代码时的顺序，而是调用方法的顺序。



**案例描述**：

任务描述

**案例代码**：（源文件名称）

1、// 匿名函数

function () {

// 这里写要执行的代码

}

// 有名字的函数

function abc() {

alert('123');

}

// 匿名函数的调用

(function () { // 代码 })();

abc(); // 弹窗，显示123。这里是调用函数

2、**function play (time) {**

**alert(time + '点了,去玩耍了！');**

**}**

**function eat (time) {**

**alert(time + '点了,吃早饭了！');**

**}**

**function study (time) {**

**alert(time + '点了,好好学习！');**

**}**

**eat(8); // 函数调用**

**study (10);**

**play (19);**

**3、function myFunction(){**

**var x=5;**

**return x;**

**}**

**var myVar=myFunction();**

**console.log(myVar);**

**案例说明**：

1）案例说明：匿名函数和有名函数的区别。以及函数的调用。

2）案例说明：函数调用时，传入参数。time是参数，形参。在用调用方法时，传入实参。



1. 问题1：定义函数的关键字？

2）问题2：关于函数的参数，形参和实参的区别？



1）关于本知识点的注意事项1： 注意区分形参和实参

2）关于本知识点的注意事项2： 函数执行的部分需要多花时间理解



练习1：

参考代码：结合if判断

var time = 20;

if (7<time && time<9) {

eat(time);

}else if (9<=time && time<=18) {

study (time);

}else if (18< time < 22) {

play (time);

}

function eat (time) {

console.log(time + '点了,吃早饭了！');

}

function study (time) {

console.log(time + '点了,好好学习！');

}

function play (time) {

console.log(time + '点了,去玩耍了！');

}

知识点1：[80分钟]

当网页被加载时，浏览器会创建页面的文档对象模型（Document Object Model）。

HTML DOM 模型被构造为对象的树：



函数是由事件驱动的或者当它被调用时执行的可重复使用的代码块。

知识点1文字说明:DOM、事件

函数执行是因为被调用。这里我们规定在发生什么事件的时候去触发某个函数。例如：onclick当点击的时候，

onload事件

onmouseover和onmouseout事件：当鼠标移入移除的时候

onfocus、onblur和onsubmit事件：焦点事件、失去焦点事件、提交表单事件



**案例描述**：

任务描述

**案例代码**：（源文件名称）

获取DOM： document.getElementById("myBtn")

document.getElementByTagName("p")

document.getElementByClassName("btn")

1、// 可以在标签上添加事件

**<button onclick="message()">点击这里</button>**

**// 通过document.getElementById("myBtn")获取到DOM元素**

**<body>**

**<button id=“myBtn”></button>**

**<div id=“box”></div>**

**</body>**

**<script>**

**document.getElementById("myBtn").onclick=function(){message()};**

**function message(){**

**document.getElementById("box").innerHTML=‘点击事件’;**

**}**

**</script>**

1. **onload事件**

**<body>**

**<img id="myImg" width="1000" src="http://img2.niutuku.com/desk/1208/1401/ntk-1401-8806.jpg" />**

**</body>**

**<script>**

**window.onload = function(){**

**alert("页面dom加载完成");**

**}**

**document.getElementById("myImg").onload = function(){**

**alert("图片已经加载完成");**

**}**

**</script>**

1. **onmouseover和onmouseout事件**

**// this代表当前对象。这里就是div**

**<div onmouseover="mOver(this)" onmouseout="mOut(this)" style="background-color:#D94A38;width:120px;height:20px;padding:40px;">**

**Mouse Over Me**

**</div>**

**<script>**

**function mOver(obj){**

**obj.innerHTML="Thank You"**

**}**

**function mOut(obj){**

**obj.innerHTML="Mouse Over Me"**

**}**

**</script>**

**4、onfocus 事件 onblur 事件**

**<input id="name" type="text" placeholder="">**

**<script>**

**var oName = document.getElementById("name");**

**oName.onfocus = function(){**

**alert("输入框已获得焦点");**

**};**

**oName.onblur = function(){**

**alert("输入框失去焦点");**

**};**

**</script>**

1. **onsubmit事件**

**<form onsubmit="myFunction()">**

**输入名字: <input type="text" name="fname">**

**<input type="submit" value="提交">**

**</form>**

**<script>**

**function myFunction() {**

**alert("表单已提交");**

**}**

**</script>**

**案例说明**：

1）案例说明：获取页面中的DOM

2）案例说明：列举常见事件的用法



1）问题1：获取页面中的DOM

2）问题2：学习的2种触发事件的方式？

****

1）关于本知识点的注意事项1：js要写在标签之后，否则获取不到DOM。

2）关于本知识点的注意事项2：练习理解常见的事件，以及如何触发事件。



练习1：

参考代码：见 : 练习/DOM事件

练习2：

参考代码：

知识点1：[60分钟]

2.4.1 知识点4内的第1小知识点

知识点1文字说明:BOM 定时器 全局函数

window对象：所有浏览器都支持 window 对象。它表示浏览器窗口。

所有 JavaScript 全局对象、函数以及变量均自动成为 window 对象的成员。

全局变量是 window 对象的属性。

全局函数是 window 对象的方法。

window.screen 对象：包含有关用户屏幕的信息。

window.location 对象：用于获得当前页面的地址 (URL)，并把浏览器重定向到新的页面。

window.history 对象：包含浏览器的历史。

定时器：JavaScript 设定一定时间间隔之后来执行代码。也称计时事件。

setInterval() - 间隔指定的毫秒数不停地执行指定的代码。

setTimeout() - 暂停指定的毫秒数后执行指定的代码

window.setInterval()这是window对象的方法，可以省略window前缀。直接使用setInterval()。

setInterval()方法的第一个参数是函数（function），第二个参数间隔的毫秒数。

全局函数：parseInt()，parseFloat()，isNaN()



**案例描述**：

任务描述

**案例代码**：（源文件名称）

代码：

**window的尺寸：**

有三种方法能够确定浏览器窗口的尺寸。

对于Internet Explorer、Chrome、Firefox、Opera 以及 Safari：

window.innerHeight - 浏览器窗口的内部高度(包括滚动条)

window.innerWidth - 浏览器窗口的内部宽度(包括滚动条)

对于 Internet Explorer 8、7、6、5：

document.documentElement.clientHeight

document.documentElement.clientWidth

或者

document.body.clientHeight

document.body.clientWidth

**screen屏幕：**

screen.availWidth - 可用的屏幕宽度

screen.availHeight - 可用的屏幕高度

<script>

document.write("可用宽度: " + screen.availWidth);

</script>

**location:**

location.hostname 返回 web 主机的域名

location.pathname 返回当前页面的路径和文件名

location.port 返回 web 主机的端口 （80 或 443）

location.protocol 返回所使用的 web 协议（http:// 或 <https://）>

<script>

document.write(location.href);

</script>

<script>

function newDoc()

{

window.location.assign("http://www.baidu.com")； // 加载新文档

}

</script>

**history:**

history.back() - 与在浏览器点击后退按钮相同,加载历史列表中的前一个 URL。

history.forward() - 与在浏览器中点击按钮向前相同,加载历史列表中的下一个 URL。

<script>

function goBack(){

window.history.back()

}

</script>

<script>

function goForward(){

window.history.forward()

}

</script>

**开启setInterval：**

<p id="demo"></p>

<script>

var myVar=setInterval(function(){myTimer()},1000);

function myTimer(){

var d=new Date();

var t=d.toLocaleString();

document.getElementById("demo").innerHTML=t;

}

</script>

**停止setInterval：**

<p id="demo"></p>

<button onclick="myStopFunction()">停止时钟</button>

<script>

var myVar=setInterval(function(){myTimer()},1000);

function myTimer(){

var d=new Date();

var t=d.toLocaleTimeString();

document.getElementById("demo").innerHTML=t;

}

function myStopFunction(){

clearInterval(myVar);

}

</script>

**开启setTimeout()：**

3秒后弹出‘Hello’

<button onclick="myFunction()">点击</button>

<script>

function myFunction(){

setTimeout(function(){alert("Hello")},3000);

}

</script>

**停止setTimeout()：**

<p>点击第二个按钮来阻止第一个函数运行。（你必须在3秒之前点击它）。</p>

<button onclick="myFunction()">点我</button>

<button onclick="myStopFunction()">停止弹框</button>

<script>

var myVar;

function myFunction(){

myVar=setTimeout(function(){alert("Hello")},3000);

}

function myStopFunction(){

clearTimeout(myVar);

}

</script>

**parseInt():**

<script>

document.write(parseInt("10") + "<br>");

document.write(parseInt("10.33") + "<br>");

document.write(parseInt("34 45 66") + "<br>");

document.write(parseInt(" 60 ") + "<br>");

document.write(parseInt("40 years") + "<br>");

document.write(parseInt("He was 40") + "<br>");

</script>

**parseFloat()：**

注意： 字符串中只返回第一个数字。开头和结尾的空格是允许的。如果字符串的第一个字符不能被转换为数字，那么 parseFloat() 会返回 NaN非数字。

使用 parseFloat() 来解析不同的字符串：

<script>

document.write(parseFloat("10") + "<br>");

document.write(parseFloat("10.33") + "<br>");

document.write(parseFloat("34 45 66") + "<br>");

document.write(parseFloat(" 60 ") + "<br>");

document.write(parseFloat("40 years") + "<br>");

document.write(parseFloat("He was 40") + "<br>");

</script>

**isNaN()：**

isNaN() 函数用于检查其参数是否是非数字值。

如果参数值为 NaN 或字符串、对象、undefined等非数字值则返回 true, 否则返回 false。

<script>

document.write(isNaN(123)+ "<br>");

document.write(isNaN(-1.23)+ "<br>");

document.write(isNaN(5-2)+ "<br>");

document.write(isNaN(0)+ "<br>");

document.write(isNaN("Hello")+ "<br>");

document.write(isNaN(NaN)+ "<br>");

</script>

**案例说明**：

1）案例说明： window对象的属性

2）案例说明： window对象的方法。以及讲解3个常用全局函数



1）问题1：window对象的属性有？

2）问题2：window对象的方法可以如何简写？

****

1. 关于本知识点的注意事项1： window对象的几个属性和方法，可以省略 window. ,直接使用。

2）关于本知识点的注意事项2：两种定时器的区别。



练习1：

参考代码：见 练习/定时器

知识点1：[80分钟]

2.5.1 知识点5内的第1小知识点

知识点1文字说明:对象、属性、方法

举例：汽车就是现实生活中的对象。

汽车的属性：

car.name=Fiat

car.model=500

car.weight=850kg

car.color=white

汽车的方法：

car.start()

car.drive()

car.brake()

汽车的属性包括名称、型号、重量、颜色等。

所有汽车都有这些属性，但是每款车的属性都不尽相同。

汽车的方法可以是启动、驾驶、刹车等。

所有汽车都拥有这些方法，但是它们被执行的时间都不尽相同。

在 JavaScript 中，对象拥有属性和方法。

js中对象的属性和方法：对象.属性 对象.方法

var txt = "Hello"; // JavaScript 字符串对象

属性：

txt.length=5

方法：

txt.indexOf()

txt.replace()

txt.search()



**案例描述**：

任务描述

**案例代码**：（源文件名称）

代码

**定义js对象：**

1、 person=new Object();

person.firstname="Bill";

person.lastname="Gates";

person.age=56;

person.eyecolor="blue";

// 控制台打印每一个属性

2、<script>

function person(firstname,lastname,age,eyecolor){

this.firstname=firstname;

this.lastname=lastname;

this.age=age;

this.eyecolor=eyecolor;

this.changeName=changeName;

function changeName(name){

this.lastname=name;

}

}

myMother=new person("Sally","Rally",48,"green");

myMother.changeName("Doe");

document.write(myMother.lastname);

</script>

3、 var obj = {a: 123， b: ‘abc’}；

**案例说明**：

1）案例说明: 定义js对象

2）案例说明： 学习js的内置对象



1）问题1：js对象的方式有？

2）问题2：几种js内置对象？

****

1）关于本知识点的注意事项1：js对象内的属性值可以是任意值。

2）关于本知识点的注意事项2



练习1：

参考代码：学员自己定义对象，并在控制台输出查看。

### 2.5.2 知识点5内的第2小知识点

知识点2文字说明 数组



**案例描述**：

任务描述：数组中可以存储一系列的值

**案例代码**：（源文件名称）

代码

创建一个数组

创建一个数组，有三种方法。

1: 常规方式:

var week = new Array();

week [0]="Monday";

week [1]="Tuesday";

week [2]="Wednesday";

2: 简洁方式:

var week =new Array("Monday","Tuesday","Wednesday");

3: 字面:

var week =["Monday","Tuesday","Wednesday"];

数组的属性和方法：

var x=week.length // week中元素的数量

var y=week.indexOf("Tuesday") // "Tuesday" 值的索引值

删除数组的最后一个元素 - pop()

数组的末尾添加新的元素 - push()

删除数组的第一个元素 - shift()

在数组的开头添加新元素 - unshift()

**案例说明**：

1）案例说明：定义数组、数组的属性

2）案例说明：数组的方法



1）问题1：数组内可以存放的值可以是哪些？

2）问题2：shift()方法的返回值是？

****

1）关于本知识点的注意事项1：理解数组的下标。

2）关于本知识点的注意事项2：数组的length属性结合for使用



练习1：

参考代码：(文件路径：练习/数组方法/彩色条纹)

练习2：

参考代码：(文件路径：练习/数组方法/切换彩色条纹)

### 2.5.3 知识点5内的第3小知识点

知识点3文字说明

**正则表达式**：正则表达式描述了字符的模式对象。当您检索某个文本时，可以使用一种模式来描述要检索的内容。



**案例描述**：

任务描述: 认识并运用正则表达式验证表单

**案例代码**：（源文件名称）

代码

以下这两种写法是等价的。

var re = new RegExp("\\w+");

var re = /\w+/;

RegExp 修饰符：

修饰符用于执行不区分大小写和全文的搜索。

i - 修饰符是用来执行不区分大小写的匹配。

g - 修饰符是用于执行全文的搜索（而不是在找到第一个就停止查找,而是找到所有的匹配）。

<script>

var str="Is this all there is?";

var patt1=/is/g;

document.write(str.match(patt1));

</script>

test()：test()方法搜索字符串指定的值，根据结果并返回真或假。

var patt1=new RegExp("e");

document.write(patt1.test("The best things in life are free"));

exec()：exec() 方法检索字符串中的指定值。返回值是被找到的值。如果没有发现匹配，则返回 null。

var patt1=new RegExp("e");

document.write(patt1.exec("The best things in life are free"));

**案例说明**：

1）案例说明

2）案例说明



1）问题1：正则表达式的定义方式？修饰符的作用？

2）问题2：正则的两个方法？如何使用

****

1）关于本知识点的注意事项1

2）关于本知识点的注意事项2



练习1：

参考代码：(文件路径：练习/正则表达式/邮箱格式)

练习2：

参考代码：(文件路径：练习/正则表达式/02表单验证)

# 本章总结 [20分钟]

用导图做总结（见：其他参考资料）

# 测试题 [分钟]

Key1 测试题1：

Key1 测试题2：

# 作业 [30分钟]

writer01 习题1：轮播图 （见： 案例/轮播图）

writer01 习题2：省市县三级联动 （见： 案例/省市县联动）

# 扩展部分（选讲内容）：

* 扩展部分1：
* 扩展部分2：

# 学员问题汇总（经验积累）：

* 学员听课中提问的问题：
* 学员练习中遇到的问题：