第二个月

## day-1 js的概述盒基础语法

### 1.1 Javascript概述

### 1.2 JavaScript组成

#### 1.2.1 ECMAScript

#### 1.2.2 文档对象模型(DOM)

#### 1.2.3 浏览器对象模型（BOM）

### 1.3 JavaScript引入

### 1.4 JavaScript语句

### 1.5 变量

#### 1.5.1 变量的概念

#### 1.5.2 变量的声明

### 1.6 typeof操作符

### 1.7 数字（Number）类型

#### 1.7.1 整型

#### 1.7.2 浮点型

#### 1.7.3 数值范围

#### 1.7.4 NaN

### 1.8 文本（String）类型

#### 1.8.1 字符串直接量

#### 1.8.2 转义字符

#### 1.8.3 字符串的使用

### 1.9 布尔（Boolean）类型

### 1.10 Undefined类型

### 1.10 Null类型

### 1.12 对象（Object）类型

### 1.13 数据类型转换

#### 1.13.1 各种数据类型转换为字符串

#### 1.13.2 各种数据类型转换为数值

#### 1.13.3 各种数据类型转换为布尔值

### 1.14 表达式的概念

### 1.15 运算符

#### 1.15.1 算术运算符

#### 1.15.2 关系运算符

#### 1.15.3 逻辑运算符

#### 1.15.4 赋值运算符

#### 1.15.5 条件运算符

#### 1.15.6 运算符优先级

## day-2 流程控制语句

### 2.1 if语句

### 2.2 switch语句

### 2.3 for语句

### 2.4 跳转语句

#### 2.4.1 break语句

#### 2.4.2 continue语句

### 2.5 while语句

### 2.6 do-while语句

### 2.7 案例

## day-3 函数

### 3.1 函数定义

### 3.2 函数调用

### 3.3 函数参数

#### 3.5.1 形参

#### 3.5.2 实参

### 3.4 函数的返回值

### 3.5 局部变量和全局变量（了解）

## day-4 数组的使用

### 4.1 定义数组

### 4.2 数组的属性

### 4.3 遍历数组

### 4.4 arguments对象

### 4.5 二维数组(了解)

### 4.6 数组的方法

#### 4.6.1 join()

#### 4.6.2 reverse()

#### 4.6.3 sort()

#### 4.6.4 concat()

#### 4.6.5 slice()

#### 4.6.6 splice()

#### 4.6.7 push()和pop()

#### 4.6.8 unshift()和shift()

#### 4.6.9 toString()

### 4.7 数组的运用

4.7.1数组的去重

### day-5 string对象

### 5.1 String对象

### 5.2 String对象的属性

### 5.3 String对象的方法

5.3.1 字符方法

5.3.2 字符串操作方法

5.3.3 字符串位置方法

5.3.4 字符串大小写转换方法

### day-6 Math与日期对象

### 6.1 Math对象

### 6.2 Math对象的属性

### 6.3 Math对象的方法

#### 6.3.1 min()和max()方法

#### 6.3.2 ceil()

#### 6.3.3 floor()

#### 6.3.4 round()

#### 6.3.5 random()

#### 6.3.6 abs()

#### 6.3.7 pow()

#### 6.3.8 sqrt()

### 6.4 Date对象的方法

#### 6.4.1 toLocaleString()

#### 6.4.2 toString()

#### 6.4.3 valueOf()

#### 6.4.4 getTime()

#### 6.4.5 getFullYear()

#### 6.4.6 getMonth()

#### 6.4.7 getDate()

#### 6.4.8 getDay()

#### 6.4.9 getHours()

#### 6.4.10 getMinutes()

#### 6.4.10 getSeconds()

#### 6.4.12 getMilliseconds()

### 6.5 获取对象

#### 6.3.1 getElementById()

#### 6.3.2 getElementsByTagName()

#### 6.3.3 getElementsByName()

## day-7 bom

### 10.1 window对象的讲解

### 7.2全局作用域

### 7.3窗口关系及框架

### 7.4窗口位置和大小

### 7.5 打开窗口

### 7.6 间歇调用和超时调用

### 7.7系统对话框

### 7.8 location对象

### 7.9 navigator对象

### 7.10 screen对象

### 7.10 history对象

## day-8 dom1

### 9.1 DOM概念:

### 8.2 节点和节点层次关系

#### 8.2.1 节点层次：DOM树形结构

#### 8.2.2 节点类型：文档中的内容，包括标签、文本、注释、标签属性等

#### 8.2.3 节点属性：

#### 8.2.8 节点关系

#### 8.2.5 常用属性

### 8.3 文档写入

#### 8.3.1 document.write():

#### 8.3.2 document.writeIn():

### 8.4 查找元素

#### 8.4.1 getElementById()

#### 8.4.2 getElementsByTagName()

#### 8.4.3 getElementsByName()

#### 8.4.4 获取元素中的属性值

#### 8.4.5 设置元素中的属性值

### 8.5标签元素的属性

#### 8.5.1获取元素节点的标签名

#### 8.5.2 获取、设置元素节点里的内容（内容可以包含html标签）

#### 8.5.3 outerHTML/innerText，非w3c的DOM标准

#### 8.5.4 innerHTML和outerHTML的区别

#### 8.5.5 innerText

#### 8.5.6 style.CSS

### 8.6节点遍历

#### 8.6.1 childNodes

#### 8.6.2 firstChild

#### 8.6.3 lastChild

#### 8.6.4 ownerDocument

#### 8.6.5 parentNode

#### 8.6.6 previousSibling

#### 8.6.7 nextSibling

Da9-9 Dom(2)

### 9.1标签节点的操作方法

#### 9.1.1 document.creatElement()（最常用）

#### 9.1.2 document.creatTextNode()

#### 9.1.3 append Child()、

#### 9.1.4 insertBefore(插入的新节点，原节点)

#### 9.1.5 body.insertBefore(header, h1);

#### 9.1.6 replaceChild(新节点，老节点)

#### 9.7.7 div.replaceChild(span,p)

#### 9.7.8 removeChild()

#### 9.7.9 div.removeChild(p)

#### 9.7.10 remove()

#### 9.7.10 div.remove()

### 9.2案例

#### 9.2.1 点击按钮创建按li

#### 9.2.2 点击按钮反向创建li

#### 9.2.3 点击按钮创建li并且可以删除节点

#### 9.2.4 利用for循操作dom

### 9.3克隆节点

### 9.4 DOM常见属性

#### 9.4.1 offsetWidth：获取元素 度(包含border)

#### 9.4.2 offsetHeight：获取元素的高度(包含border)

#### 9.4.3 clientWidth：获取元素的宽度( 不包含border)

#### 9.4.4 clientHeight：获取元素的高度 (不包含border)

#### 9.4.5 offsetTop：获取定 位元素距离父级上边的距离

#### 9.4.6 offsetLeft：获取 定位元素距离父级左边的距离

### 9.5滚动偏移动(淘宝返回顶部)

#### 9.5.1 滚动条事件

#### 9.5.2 onscroll

#### 9.5.3使用DOM操作实现图- 片懒加载

#### 9.5.4图片

#### 9.5.5图片预加载

### 9.6 作业

#### 9.6.1留言板

## day-10事件

### 10.1什么是事件

#### 10.1.1 window.onload

#### 10.1.2 window.onscroll

### 10.2事件对象

#### 10.2.1 获取对象

#### 10.2.2 事件对象的兼容写法

### 10.3鼠标事件mouse

#### 10.3.1 onclick:点击事件,按下并抬起

#### 10.3.2 onmousedown:鼠标按下

#### 10.3.3 onmouseup:鼠标抬起

#### 10.3.4 onmousemove:鼠标移动

#### 10.3.5 ondbclick:双击事件

#### 10.3.6 oncontextmenu:右键菜单

### 10.4键盘事件key

#### 10.4.1 onkeydown: 键盘按下

#### 10.4.2 onkeyup:键盘抬起

#### 10.4.3 可通过keycode属性获取唯一标识码

#### 10.4.4 实例

### 10.5表单事件

#### 10.5.1 onfocus:获取焦点

#### 10.5.2 onblur:失去焦点

#### 10.5.3 onsubmit:提交事件

#### 10.5.4 ev.preventDefault():阻止默认事件。不兼容IE6~8

#### 10.5.5 return false :阻止默认事件，没有兼容性问题

### 10.6 on的方式

#### 10.6.1 给同一个元素绑定两次事件,使用on的方式，后面会覆盖前面的

#### 10.6.2 addEventListener()

#### 10.6.2.1 不会发生覆盖的情况

#### 10.6.2.2 不兼容IE6~8

#### 10.6.2.3 removeEventListener()

#### 10.6.3 回调函数

#### 10.6.3.1 把函数当成参数传入

### 10.7事件冒泡

#### 10.7.1 一个标签元素处理完事件之后，它的父元素、父元素的父元素也都会处理事件

#### 10.7.2 取消事件冒泡

#### 10.7.3 addEventListener() 冒泡的方式

#### 10.7.4 event.stopPropagation();取消事件传递

#### 10.7.5 addEventListener(false) 冒泡的方式

#### 10.7.6 addEventListener(true) 下沉的方式

### 10.8阻止事件传递

#### 10.8.1 ev.cancelBubble = true或者ev.stopPropagation()

#### 10.8.2 实例:使用事件绑定的方式实现拖拽

### 10.9鼠标滚轮事件

#### 10.9.1 非火狐: mousewheel

#### 10.9.2 火狐:DOMMouseScroll

#### 10.9.3 案例:使用滚轮改变图片的大小

### 10.10实例

#### 10.10.1 拖拽效果

#### 10.10.2 onmousedown

#### 10.10.3 onmousemove

#### 10.10.4 onmouseup

#### 10.10.5 注意：事件加给document

## day-11 动画1

### 11.1 小球移动

### 11.2 碰壁反弹

### 11.3 无间断滚动

### 11.4 轮播图

### 11.5 模态窗口

## day-12正则表达式

### 12.1 什么是正则？

#### 12.1.1 Regular Expression，一套匹配规则，匹配想要的东西：把人类所认可的一种规则使用计算机能读懂的语言表达出来，正则正能操作字符串，是一种正常人类很难读懂的语法格式。

### 12.2 正则表达式对象的创建

#### 12.2.1 语法

#### 12.2.2 方法

#### 12.2.2.1 test():获取到匹配的字符返回true，否则返回false

#### 12.2.2.2 exec():获取到匹配的字符并返回，如果没有则返回null

### 12.3字符串的方法

#### 12.3.1 str.search(/haha/)

#### 12.3.2 str.replace(/haha/, “hehe”)

#### 12.3.3 str.match(/\d/g):找到匹配的字符并返回

### 12.4正则语法规则

#### 12.4.1 modifiers（修饰词）

#### 12.4.1.1 i 匹配时忽略大小写

#### 12.4.1.2 g 执行时全局匹配:查找所有匹配项，而非查找到一个即停止

#### 12.4.2 匹配模式

#### 12.4.2.1 小括号:(abc)

#### 12.4.2.2 中括号:[12345]

#### 12.4.2.3 [0-9]:包括所有数字

#### 12.4.2.4 [abc]:包括a、b、c其中的一个

#### 12.4.2.5 [a-z]:包括所有的小写字母

#### 12.4.2.6 [A-Z]:包括所有的大写字母

#### 12.4.2.7 [A-z]:包括所有的字母

#### 12.4.2.8 [0-9a-z]:包含所有的数字和小写字母

#### 12.4.2.9 [^ab]: 除了字符ab

#### 12.4.3 元字符

#### 12.4.3.1 点:代表任何的字符,注意:如果需要加.的时候，需要进行转义.

#### 12.4.3.2 \w：任何字母、数字、下划线

#### 12.4.3.3 \W：任何非字母、数字、下划线

#### 12.4.3.4 \d：任何数字

#### 12.4.3.5 \D：任何非数字

#### 12.4.3.6 \s：空白字符

#### 12.4.3.7 \S：非空白字符

#### 12.4.3.8 \b：查找单词边界

#### 12.4.3.9 \B：查找非单词边界

#### 12.4.3.10 了解

#### \0：查找NULL字符

#### \n：查找换行符

#### \f： 查找分页符

#### \r：查找回车符

#### \t：查找制表符

#### \v：查找垂直制表符

#### 12.4.4 量词

#### 12.4.4.1 ^n：匹配开头为n的字符串

#### 12.4.4.2 n$：匹配结尾为n的字符串

#### 12.4.4.3 n+：匹配任何包含至少一个n的字符串

#### 12.4.4.5 n{x}： 匹配任何包含x个n的字符串

#### 12.4.4.6 n{x, y}：匹配包含x到y个n的字符串

#### 12.4.4.7 了解

#### n\*：匹配任何包含0个或者多个n的字符串

#### n?：匹配任何包含0个或1个n的字符串

#### ?=n：匹配任何其后紧接指定字符串n的字符串

#### ?!n：匹配任何其后没有紧接字符串n的字符串

### 12.5实例

#### 12.5.1 判断全部数字

#### 12.5.2 判断文件后缀为PDF

#### 12.5.3 判断手机号

#### 12.5.4 判断邮箱

#### 12.5.5 判断IP地址

### 12.6转义字符

### 12.7判断密码强度

### 12.8 检测网址

### 12.9预定义模式

### 12.10中文匹配

### 12.10匹配1~255之间的值

### 12.12检测ip地址

### 12.13.工具网站：[https://regexper.com](https://regexper.com/)

## day-13面向对象1

### 13.1面向对象的思想；继承、封装、多态。

### 13.2 Function函数

#### 13.2.1基本介绍

#### 13.2.2函数的声明方式

#### 13.2.3作为值的函数

#### 13.2.4 arguments

#### 13.2.5 apply()和call()方法的应用

#### 13.2.6返回值

#### 13.2.7递归

### 13.3对象与数组

#### 13.3.1 Object介绍

#### 13.3.2 Object类型

#### 13.3.3数组与对象的差异

#### 13.3.4哈希表、关联型数组

### 13.4访问对象属性

#### 13.4.1访问对象属性

#### 13.4.2访问对象方法

#### 13.4.3修改属性与方法

#### 13.4.4删除对象属性与方法

#### 13.4.5使用this值

### 13.5构造函数

#### 13.5.1构造器函数介绍

#### 13.5.2构造器函数的好处

#### 13.5.3 instanceof操作符

#### 13.5.4返回对象的函数

### 13.6回调函数

### 13.7返回函数的函数

### 13.8能重写自己的函数

## day-14面向对象2

### 14.1变量、作用域及内存

#### 14.1.1基本类型值与引用类型值

#### 14.1.2传递参数

#### 14.1.3检测类型

#### 14.1.4执行环境与作用域

#### 14.1.5块级作用域

#### 14.1.6 var关键字

#### 14.1.7变量查询

#### 14.1.8内存生命周期

#### 14.1.9变量内存及什么是内存泄露

### 14.2基本包装类型

#### 14.2.1基本包装类型概述

#### 14.2.2 Boolean 类型

#### 14.2.3 Number 类型

#### 14.2.4 String 类型

### 14.3创建对象

#### 14.3.1工厂模式

#### 14.3.2构造函数

#### 14.3.3构造函数的规范

##### 14.3.3.1构造函数与工厂模式的不同

##### 14.3.3.2构造函数执行流程

##### 14.3.3.3构造函数this的使用

##### 14.3.3.4构造函数与普通函数的区别

### 15.4原型

#### 15.4.1原型概述

#### 15.4.2\_\_proto\_\_属性

#### 15.4.3原型模式的执行流程

#### 15.4.4 hasOwnProperty()函数、in 操作符

#### 15.4.5 constructor属性

#### 15.4.6原型链

#### 15.4.7原型声明顺序与重写原型

#### 15.4.8组合构造函数+原型模式：

## Day-15继承

#### 15.1.1原型链继承

#### 15.1.2冒充继承

#### 15.1.3组合继承

### 15.6匿名函数和闭包

#### 15.2.1匿名函数

#### 15.2.2闭包优缺点

#### 15.2.3关于this对象

#### 15.2.4模仿块级作用域

#### 15.2.5静态私有变量

#### 15.2.6模块模式

### Day16 数据存储

### 16.1 cookie

### 16.2 lostorage,

### 17.3 sessionstorage

### Day17 JS练习1

### Day18 JS练习2