# 第一个月

## day-1 HTML基础及常用标签

### 1.1 Photoshop的基本操作

#### 1.1.1 常用快捷键

#### 1.1.2 创建选区的基本方法

#### 1.1.3 图层的创建及基本功能

#### 1.1.4 抠图

#### 1.1.5 切片工具的使用

#### 1.1.6 标尺和参考线

#### 1.1.7 吸管工具

#### 1.1.8 Photoshop中常用的图像存储格式

### 1.2 HTML简介、HTML标

### 1.3 标签的属性和属性值

### 1.4 常用标签

#### 1.4.1 标题标记以及属性

#### 1.4.2 段落标记以及对齐方式属性

#### 1.4.3 图像标记以及属性

#### 1.4.4 水平线标记以及属性

#### 1.4.5 HTML里面常用的特殊符号（空格等）

### 1.5 路径(绝对相对、相对路径)

## day-2 列表，超链接及表格

### 2.1 HTML元素的分类

#### 2.1.1 行内元素（内联元素）

#### 2.1.2 块状元素

### 2.2 无序列表

#### 2.2.1 无序列表标签

#### 2.2.2 无序列表type属性

### 2.3 有序列表ol

#### 2.3.1 有序列表标签

#### 2.3.2 有序列表type属性和reversed(倒序)属性

### 2.4 自定义列表 dl

#### 2.4.1 自定义列表标签（定义项、描述项）

### 2.5 超链接的语法及实现

#### 2.5.1 超链接的标签

#### 2.5.2 超链接href属性

#### 2.5.3 超链接target属性

#### 2.5.4 超链接对象（文本、图像、空连接、锚点链接）

### 2.6 表格的基本结构

#### 2.6.1 table标记

#### 2.6.2 tr标记

#### 2.6.3 td标记

#### 2.6.4 th标记

#### 2.6.5 thead\tbody\tfoot

### 2.7表格标记table的属性

#### 2.7.1 border/width/height属性

#### 2.7.2 cellspacing/cellpadding属性

#### 2.7.3 background/bgcolor属性

#### 2.7.4 algin属性和valgin属性

### 2.8单元格的合并与拆分

#### 2.8.1 水平合并单元格colspan

#### 2.8.2 垂直合并单元格rowspan

### 2.9使用表格布局网页

## day-3：表单

### 3.1 表单标记及其属性

#### 3.1.1 action 属性（了解）

#### 3.1.2 method属性（了解）

#### 3.1.3 target属性

### 3.2 输入标记

#### 3.2.1 文本框

#### 3.2.2 密码框

#### 3.2.3 单选框

#### 3.2.4 复选框

#### 3.2.5 文件域

#### 3.2.6 提交按钮

#### 3.2.7 普通按钮

#### 3.2.8 重置按钮

#### 3.2.9 图像域

### 3.3 下拉菜单

#### 3.3.1 下拉列表框的结构

#### 3.3.2 Selected属性

### 3.4 文本域标记及其属性

#### 3.4.1 Cols属性

#### 3.4.2 rows属性

### 3.5 案例

## day-4 css选择器，样式表

### 4.1 CSS的基本概念及语法

### 4.2 Css选择器

#### 4.2.1 全局选择器

#### 4.2.2 标签选择器

#### 4.2.3 类选择器

#### 4.2.4 ID选择器

#### 4.2.5 关系选择器（包含选择器、子选择器、相邻选择器）

#### 4.2.6 属性选择器

#### 4.2.7 选择器的优先级

### 4.3 伪类选择器

#### 4.3.1 a:link

#### 4.3.2 a:hover

### 4.4 在html中使用css的方法

#### 4.4.1 行内样式

#### 4.4.2 内嵌样式

#### 4.4.3 链接样式

#### 4.4.4 导入样式

#### 4.4.5 样式表的优先级

## day-5 文字，段落属性，背景属性，列表属性

### 5.1 文字属性

#### 5.1.1 color字体颜色

#### 5.1.2 font-size字号

#### 5.1.3 font-family字体

#### 5.1.4 font-weight字体加粗

### 5.2 段落属性

#### 5.2.1 文字修饰（text-decoration）

#### 5.2.2 水平对齐方式（text-align）

#### 5.2.3 文本缩进（text-indent）

#### 5.2.4 文本行高 (line-height)

### 5.3 背景属性

#### 5.3.1 background-color背景颜色

#### 5.3.2 background-image背景图像

#### 5.3.3 background-repeat背景重复

#### 5.3.4 background-position背景图像位置

#### 5.3.5 background-attachment背景附件

#### 5.3.6 background背景复合属性

### 5.4 列表属性

#### 5.4.1 list-style-type

#### 5.4.2 list-style-image

#### 5.4.3 list-style-position

### 5.5 案例

## day-6 css盒模型及css3新增属性

### 6.1 设置尺寸

#### 6.1.1 设置宽度width

#### 6.1.2 设置高度height

### 6.2 边框属性

#### 6.2.1 边框复合属性border

#### 6.2.2 上边框border-top

#### 6.2.3 右边框border-right

#### 6.2.4 下边框border-bottom

#### 6.2.5 左边框border-left

#### 6.2.6 常用边框样式：solid 实线、dashed 虚线、dotted点线

### 6.3 内填充padding

#### 6.3.1 上内填充padding-top

#### 6.3.2 右内填充padding-right

#### 6.3.3 下内填充padding-bottom

#### 6.3.4 左内填充padding-left

#### 6.3.5 掌握WEB标准中padding的使用规范

#### 6.3.6 padding对元素宽高的影响

### 6.4 外边距margin

#### 6.4.1 上外边距margin-top

#### 6.4.2 右外边距丁margin-right

#### 6.4.3 下外边距margin-bottom

#### 6.4.4 左外边距margin-left

#### 6.4.5 掌握WEB标准中margin的使用规范

### 6.5 box-shadow属性

### 6.6 text-shadow属性

### 6.7 border-radius属性

### 6.8 box-sizing属性

### 6.9 text-overflow属性

### 6.10 gradient渐变

### 6.11 浏览器调试

## day-7 布局属性

### 7.1 什么是网页布局

### 7.2 浮动float属性

#### 7.2.1 left左浮动

#### 7.2.2 right右浮动

### 7.3 清除浮动clear属性

### 7.4 浮动属性的应用

#### 7.4.1 Div的浮动（实现布局）

#### 7.4.2 列表的浮动（实现导航）

### 7.5 使用盒模型布局搭建布局色块

### 7.6 案例

## day-8 display与overflow属性的应用及定位

### 8.1 dislay:inline的应用

### 8.2 display:block的应用

### 8.3 display:none的应用

### 8.4 display:inline-block的应用

### 8.5 CSS样式中overflow的运用规范

### 8.6 CSS样式中visibility的运用规范

### 8.7 visibility:hidden与display:none的区别

### 8.8 掌握定位属性的四个值：

#### 8.8.1 默认值：static

#### 8.8.2 绝对定位：absolute

#### 8.8.3 相对定位：relative

#### 8.8.4 固定定位：fixed

### 8.9掌握定位属性方向值运用规范

#### 8.9.1 上部：top的运用规范

#### 8.9.2 下部：bottom的运用规范

#### 8.9.3 左部：left的运用规范

#### 8.9.4 右部：right的运用规范

### 8.10 掌握z-index空间位置的运用规范

### 8.11 案例

## day-9项目一

## day-10 项目(静态页面巩固布局)

## day-11 HTML5(新标签)

### 11.1 HTML5 概述

### 11.2基本格式

### 11.3 HTML5新标签

#### 11.3.1 新增的结构化标签

#### 11.3.2新增的非结构化标签

### 11.4表单新属性

#### 11.4.1 新增的表单元素

#### 11.4.2 新增的表单元素的属性

### 11.5案例：

## day-12 CSS3

### 12.1 CSS3选择器

#### 12.1.1相邻兄弟选择器

#### 12.1.2通用兄弟选择器

#### 12.1.3属性选择器

#### 12.1.4目标伪类

#### 12.1.5元素状态伪类

#### 12.1.6结构伪类

#### 12.1.7否定伪类

#### 12.1.8伪元素:first-letter

#### 12.1.9伪元素::selection

### 12.2内容生成

#### 12.2.1伪元素:after和:before

#### 12.2.2伪元素:first-line

#### 12.2.3内容content

### 12.3 CSS3变形效果

#### 12.3.1 transform

#### 12.3.2 transform-origin

#### 12.3.3浏览器版本

#### 12.3.4 3D 变形简介 （了解）

#### 12.3.5 transform-style

#### 12.3.6 perspective

#### 12.3.7 3D 变形属性

### 12.4过渡效果

#### 12.4.1过渡简介

#### 12.4.2 transition-property

#### 12.4.3 transition-duration

#### 12.4.4 transition-timing-function

#### 12.4.5 transition-delay

#### 12.4.6简写和版本

### 12.5案例：

#### 12.5.1 CSS3时钟

## day-13 CSS动画效果

### 13.1CSS3 前缀和 rem

#### 13.1.1CSS3 前缀

#### 13.1.2长度单位 rem

### 13.2动画

#### 13.2.1动画简介

#### 13.2.2 animation 属性详解

#### 13.2.3简写和版本

### 13.3第三方动画库：animate css（了解）

### 13.4案例:

## day-14 rem布局，flex布局，vw布局，移动端(viewport)

### 14.1媒体查询

#### 14.1.1什么是媒体查询

#### 14.1.2如何使用媒体查询满足不同的分辨率

### 14.2 flex布局

#### 14.2.1什么是flex布局

#### 14.2.2如何实现flex布局

### 14.3 rem布局

#### 14.3.1什么是rem布局

#### 14.3.2如何实现rem布局

### 14.4分栏布局

#### 14.4.1分栏布局页面的编写

### 14.5案例：

## day-15 移动端项目(静态页面)

## day-16 响应式布局(@media)

### 16.1响应式布局

#### 16.1.1什么是响应式布局

#### 16.1.2如何实现响应式布局

### 16.2案例：

#### 16.2.1 gulp官网

## day-17 bootstrap1(静态样式的使用)

### 17.1 Bootstrap 介绍

#### 17.1.1 Bootstrap 概述

#### 17.1.2 Bootstrap 特点

#### 17.1.3 Bootstrap 结构

### 17.2排版样式

#### 17.2.1常用的排版样式

### 17.3表格与按钮

#### 17.3.1常用的表格

#### 17.3.2常用的按钮

### 17.4表单与图片

### 17.5栅格系统

### 17.6辅助类和响应式工具

## day-18 bootstrap2(组件的使用，只需要教会看文档，复制粘贴代码)

### 18.1组件应用(导航组件）

### 18.2插件应用（轮播插件）

## day-19 bootstrap 项目 1

## day-20 bootstrap 项目 2

第二个月

## day-1 js的概述盒基础语法(减了一天)

### 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

### 1.11 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 案例

## day-3 函数

### 3.1 while语句

### 3.2 do-while语句

### 3.3 函数定义

### 3.4 函数调用

### 3.5 函数参数

#### 3.5.1 形参

#### 3.5.2 实参

### 3.6 函数的返回值

### 3.7 局部变量和全局变量（了解）

## day-4 数组的使用(减了一天)

### 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

## day-5 对象的概述，math和string对象

### 5.1 Math对象

### 5.2 Math对象的属性

### 5.3 Math对象的方法

#### 5.3.1 min()和max()方法

#### 5.3.2 ceil()

#### 5.3.3 floor()

#### 5.3.4 round()

#### 5.3.5 random()

#### 5.3.6 abs()

#### 5.3.7 pow()

#### 5.3.8 sqrt()

### 5.4 String对象

### 5.5 String对象的属性

### 5.6 String对象的方法

#### 5.6.1 字符方法

#### 5.6.2 字符串操作方法

#### 5.6.3 字符串位置方法

#### 5.6.4 字符串大小写转换方法

## day-6 时间与日期对象

### 6.1 Date对象

### 6.2 Date对象的方法

#### 6.2.1 toLocaleString()

#### 6.2.2 toString()

#### 6.2.3 valueOf()

#### 6.2.4 getTime()

#### 6.2.5 getFullYear()

#### 6.2.6 getMonth()

#### 6.2.7 getDate()

#### 6.2.8 getDay()

#### 6.2.9 getHours()

#### 6.2.10 getMinutes()

#### 6.2.11 getSeconds()

#### 6.2.12 getMilliseconds()

### 6.3 获取对象

#### 6.3.1 getElementById()

#### 6.3.2 getElementsByTagName()

#### 6.3.3 getElementsByName()

### 6.4 定时器

#### 6.4.1 setInterval()

#### 6.4.2 setTimeout()

#### 6.4.3 clearInterval()

## day-7 动画1

### 7.1 小球移动

### 7.2 碰壁反弹

### 7.3 无间断滚动

### 7.4 大图滚动

## day-8 动画2(减了一天，tween.js可以不要)

### 8.1 天猫轮首页播图（tween.js版）

### 8.2 当当网轮播图

### 8.3 模态窗口

## day-9 dom1

### 9.1 DOM概念:

### 9.2 节点和节点层次关系

#### 9.2.1 节点层次：DOM树形结构

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

#### 9.2.3 节点属性：

#### 9.2.9 节点关系

#### 9.2.5 常用属性

### 9.3 文档写入

#### 9.3.1 document.write():

#### 9.3.2 document.writeIn():

### 9.4 查找元素

#### 9.4.1 getElementById()

#### 9.4.2 getElementsByTagName()

#### 9.4.3 getElementsByName()

#### 9.4.4 获取元素中的属性值

#### 9.4.5 设置元素中的属性值

### 9.5标签元素的属性

#### 9.5.1获取元素节点的标签名

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

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

#### 9.5.4 innerHTML和outerHTML的区别

#### 9.5.5 innerText

#### 9.5.6 style.CSS

### 9.6节点遍历

#### 9.6.1 childNodes

#### 9.6.2 firstChild

#### 9.6.3 lastChild

#### 9.6.4 ownerDocument

#### 9.6.5 parentNode

#### 9.6.6 previousSibling

#### 9.6.7 nextSibling

### 9.7标签节点的操作方法

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

#### 9.7.2 document.creatTextNode()

#### 9.7.3 append Child()、

#### 9.7.4 insertBefore(插入的新节点，原节点)

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

#### 9.7.6 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()

#### 9.7.11 div.remove()

### 9.8案例

#### 9.8.1 点击按钮创建按li

#### 9.8.2 点击按钮反向创建li

#### 9.8.3 点击按钮创建li并且可以删除节点

#### 9.8.4 利用for循操作dom

### 9.9克隆节点

### 9.10 DOM常见属性

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

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

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

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

#### 9.10.5 offsetTop：获取定 位元素距离父级上边的距离

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

### 9.11滚动偏移动(淘宝返回顶部)

#### 9.11.1 滚动条事件

#### 9.11.2 onscroll

#### 9.11.3使用DOM操作实现图- 片懒加载

#### 9.11.4图片

#### 9.11.5图片预加载

### 9.12 作业

#### 9.12.1留言板

## day-10 bom webstorage

### 10.1 window对象的讲解

### 10.2全局作用域

### 10.3窗口关系及框架

### 10.4窗口位置和大小

### 10.5 打开窗口

### 10.6 间歇调用和超时调用

### 10.7系统对话框

### 10.8 location对象

### 10.9 navigator对象

### 10.10 screen对象

### 10.11 history对象

## day-11事件1(主要事件绑定，事件委托，事件流，事件冒泡，减了2天，砍掉飞扬的小鸟)

### 11.1什么是事件

#### 11.1.1 window.onload

#### 11.1.2 window.onscroll

### 11.2事件对象

#### 11.2.1 获取对象

#### 11.2.2 事件对象的兼容写法

### 11.3鼠标事件mouse

#### 11.3.1 onclick:点击事件,按下并抬起

#### 11.3.2 onmousedown:鼠标按下

#### 11.3.3 onmouseup:鼠标抬起

#### 11.3.4 onmousemove:鼠标移动

#### 11.3.5 ondbclick:双击事件

#### 11.3.6 oncontextmenu:右键菜单

### 11.4键盘事件key

#### 11.4.1 onkeydown: 键盘按下

#### 11.4.2 onkeyup:键盘抬起

#### 11.4.3 可通过keycode属性获取唯一标识码

#### 11.4.4 实例

### 11.5表单事件

#### 11.5.1 onfocus:获取焦点

#### 11.5.2 onblur:失去焦点

#### 11.5.3 onsubmit:提交事件

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

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

### 11.6 on的方式

#### 11.6.1 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

#### 11.6.2 addEventListener()

#### 11.6.2.1 不会发生覆盖的情况

#### 11.6.2.2 不兼容IE6~8

#### 11.6.2.3 removeEventListener()

#### 11.6.3 回调函数

#### 11.6.3.1 把函数当成参数传入

### 11.7事件冒泡

#### 11.7.1 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

#### 11.7.2 取消事件冒泡

#### 11.7.3 addEventListener() 冒泡的方式

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

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

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

### 11.8阻止事件传递

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

#### 11.8.2 实例:使用事件绑定的方式实现拖拽

### 11.9鼠标滚轮事件

#### 11.9.1 非火狐: mousewheel

#### 11.9.2 火狐:DOMMouseScroll

#### 11.9.3 案例:使用滚轮改变图片的大小

### 11.10实例

#### 11.10.1 拖拽效果

#### 11.10.2 onmousedown

#### 11.10.3 onmousemove

#### 11.10.4 onmouseup

#### 11.10.5 注意：事件加给document

## 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

## 12.11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

### 14.4原型

#### 14.4.1原型概述

#### 14.4.2\_\_proto\_\_属性

#### 14.4.3原型模式的执行流程

#### 14.4.4 hasOwnProperty()函数、in 操作符

#### 14.4.5 constructor属性

#### 14.4.6原型链

#### 14.4.7原型声明顺序与重写原型

#### 14.4.8组合构造函数+原型模式：

#### 14.4.9动态原型模式

#### 14.4.10寄生构造函数

#### 14.4.11稳妥构造函数

## day-15面向对象3

### 15.1继承

#### 15.1.1原型链继承

#### 15.1.2冒充继承

#### 15.1.3组合继承

### 15.2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

### 15.3案例

#### 15.3.1暴躁的小球面向过程与面向对象版

#### 15.3.2拖拽面向过程与面向对象版

#### 15.3.3选项卡面向过程与面向对象版

## day-16 JQ1

### 16.1 jQuery入门

#### 16.1.1什么是 jQuery

#### 16.1.2学习 jQuery 的条件

#### 16.1.3 jQuery 的版本

#### 16.1.4 jQuery 的功能和优势

#### 16.1.5下载及运行 jQuery

### 16.2基础核心

#### 16.2.1代码风格

#### 16.2.2加载模式

#### 16.2.3对象互换

#### 16.2.4多个库之间的冲突

### 16.3常规选择器

#### 16.3.1简单选择器

#### 16.3.2进阶选择器

#### 16.3.3高级选择器

### 16.4过滤选择器

#### 16.4.1基本过滤器

#### 16.4.2内容过滤器

#### 16.4.3可见性过滤器

#### 16.4.4子元素过滤器

### 16.5表单选择器

#### 16.5.1常规选择器

#### 16.5.2表单选择器

#### 16.5.3表单过滤器

## day-17 JQ2

### 17.1基础 DOM 和 CSS 操作

#### 17.1.1 DOM 简介

#### 17.1.2设置元素及内容

#### 17.1.3元属性操作

#### 17.1.4元素样式操作

#### CSS 方法

### 17.2 DOM 节点操作

#### 17.2.1创建节点

#### 17.2.2插入节点

#### 17.2.3包裹节点

#### 17.2.4节点操作

### 17.3基础事件

#### 17.3.1绑定事件

#### 17.3.2简写事件

#### 17.3.3复合事件

### 17.4事件对象

#### 17.4.1事件对象

#### 17.4.2冒泡和默认行为

### 17.5高级事件

#### 17.5.1模拟操作

#### 17.5.2命名空间

#### 17.5.3事件委托

#### 17.5.4 on、off 和 one

### 17.6案例：

#### 17.6.1数据操作

## day-18 JQ3

### 18.1动画效果

#### 18.1.1显示、隐藏

#### 18.1.2滑动、卷动

#### 18.1.3淡入、淡出

#### 18.1.4自定义动画

#### 18.1.5列队动画方法

#### 18.1.6动画相关方法

#### 18.1.7动画全局属性

### 18.2插件开发

#### 18.2.1插件概述

#### 18.2.2自定义插件

## day-19项目(至象微尔)1

## day-20项目(至象微尔)2

# 第三个月

## day-1 nodejs1

### 1.1 Node.js的简介

#### 1.1.1 什么是Node

#### 1.1.2 Node的用途

#### 1.1.3 Node的特点

#### 1.1.4 Node的历史

#### 1.1.5 Node的核心模块

### 1.2 Node的安装

#### 1.2.1官网Http://nodejs.org

#### 1.2.2 Node的安装

#### 1.2.3 Node的基本命令

### 1.3 Node的fs模块

#### 1.3.1读取文件(同步读取|异步读取)

#### 1.3.2 Node向文件写内容(同步写入|异步写入)

#### 1.3.3读写操作的综合运用：文件的拷贝

#### 1.3.4大文件如何读写(流的概念和使用)

## day-2 nodejs1

### 2.1 Node的http模块

#### 2.1.1 Node的http模块概述

#### 2.1.2 创建一个请求

#### 2.1.3创建一个http的服务

## day-3 express

### 3.1 Express的基本应用

#### 3.1.1 什么是Express

#### 3.1.2 官网http://www.expressjs.com.cn

#### 3.1.3 安装express

#### 3.1.4 express的Helloworld

#### 3.1.5 express的脚手架

#### 3.1.6 express的基本路由

#### 3.1.7静态文件

### 3.2 Express的复杂应用

#### 3.2.1路由

#### 3.2.2中间件

#### 3.2.3模板引擎

#### 3.2.4错误处理

### 3.3案例：

#### 3.3.1定义接口服务

#### 3.3.2创建上传文件的服务

## day-4 ajax1

### 4.1安装服务器端环境

#### 4.1.1 Deployd是什么

#### 4.1.2 Deployd如何安装

#### 4.1.3介绍接口

### 4.2 Ajax 概述

#### 4.2.1什么是ajax

#### 4.2.2 ajax的优点

### 4.3 XMLHttpRequest

#### 4.3.1创建xhr对象

#### 4.3.2 xhr对象创建的兼容性问题

### 4.4 GET 与 POST

#### 4.4.1 get方式请求Deployd的接口

#### 4.4.2 post方式请求Deployd的接口

#### 4.4.3 get方式与post方式的区别

### 4.5封装 Ajax

## day-5 ajax2(JQ,CORS,jsonp)

### 5.1 Json

#### 5.1.1 JSON 语法

#### 5.1.2 解析和序列化

### 5.2 JQ中的AJAX

#### 5.2.1 $.load()方法

#### 5.2.2 $.get()和$.post()

#### 5.2.3 $.getScript()和$.getJSON()

#### 5.2.4 $.ajax()方法

#### 5.2.5 表单序列化

### 5.3跨域访问

#### 5.3.1 同源策略

#### 5.3.2 jsonp跨域

## day-6 php1

## 6.1 php基础语法

#### 6.1.1 php简介

#### 6.1.2 php变量

#### 6.1.3 php常量

#### 6.1.4 php分支语句

#### 6.1.5 php输出

## 6.2预定义变量

#### 6.2.1 php$\_GET

#### 6.2.2 php$\_POST

#### 6.2.3 php$\_REQUEST

## day-7 php2

## 7.1 MySQL

#### 7.1.1 MySQL语句入门

#### 7.1.2 MySQL创建数据库/表

#### 7.1.3 增删改查

## day-8 php3

## 8.1 后台管理项目(淘宝卖家)

#### 8.1.1使用php搭建后台环境

#### 8.1.2使用mysql创建用户表格/商品表格

#### 8.1.3 增删改查实现商品的上架与下架

8.1.4 配合前端页面,实现数据的动态变化

## day-9 微信公众号开发

## 9.1 环境配置

## 9.2 获取token值

## 9.3 获取accesstoken

## 9.4 自定义菜单

## 9.5 事件

## day-10 微信jssdk

## 10.1 获取ticket

## 10.2 验证签名

## 10.3 jssdk解析

## 10.4 微信分享

## 10.5 链接数据库

## 10.6 saeSql

## day-11 微信项目练习

## 11.1 内嵌网页开发

## day-12 requirejs 加入项目

### 12.1 require.js简介

### 12.2 基本API

#### 12.2.1 define,require,requirejs 三个变量

#### 12.2.2加载配置文件

#### 12.2.3 define的使用

#### 12.2.4 html引入并使用

### 12.3 加载非规范模块

### 12.4 AMD/CMD

## day-13 less，sass

### 13.1什么是Less/Sass

### 13.2如何安装

### 13.3如何编译

### 13.4 Less语法

#### 13.4.1导入

#### 13.4.2普通变量

#### 13.4.3默认变量

#### 13.4.4特殊变量

#### 13.4.5多值变量

#### 13.4.6嵌套(选择器嵌套)

#### 13.4.7嵌套(属性嵌套)

#### 13.4.8混合(无参数mixin)、混合(有参数mixin)、混合(有默认参数mixin)、混合(有多组值参数mixin)

#### 13.4.9混合(@content)

#### 13.4.10继承

#### 13.4.11运算

#### 13.4.12函数

### 13.13 Sass语法

#### 基本同Less差不多

## day-14 gulp

### 14.1 gulp概述

### 14.2 gulp的安装

### 14.3 gulp的使用

#### 14.3.1基本的配置

#### 14.3.2高级的配置

#### 14.3.3 常用的插件

### 14.4 gulp搭建项目实战

## day-15 同城帮项目1

### 15.1 首页静态页面搭建

#### 15.1.1 首页header创建

#### 15.1.2 首页banner创建

#### 15.1.3 热门手机 热门回收 创建

#### 15.1.4 店铺列表创建

#### 15.1.5 好评店铺列表创建

### 15.2 店铺页面搭建

#### 15.2.1 header创建

#### 15.2.2 店铺介绍创建

#### 15.2.3 商品列表创建

#### 15.2.4 用户列表创建

#### 15.2.5 店铺详情创建

#### 15.2.6 服务保障创建

#### 15.2.7 footer创建

## day-16 同城帮项目2

### 16.1物车页面创建

#### 16.1.1 Header部分创建

#### 16.1.2 Logo部分创建

#### 16.1.3 确认订单创建

#### 16.1.4 Footer创建

### 16.2同城帮数据截取

#### 16.2.1使用浏览器调试中的netwrok

#### 16.2.2筛选出xhr请求，截取json文件

#### 16.2.3在店铺页面，筛选出js请求，截取jsonp中的json文件

### 16.3 node.js服务器搭建

#### 16.3.1配置express框架；

#### 16.3.2书写server.js文件

#### 16.3.3启动服务器

### 16.4 切换城市模块

#### 16.4.1使用ajax请求json文件；

#### 16.4.2把json数据解析出来显示在城市列表的模块上；

#### 16.4.3点击 ABCD这样子的标签，切换对应的城市；

### 16.5 banner图上的树形菜单模块

#### 16.5.1 使用ajax请求json文件；

#### 16.5.2 把json文件中的数据解析出来现在二级菜单上

## day-17 同城帮项目3

### 17.1登陆注册模块

#### 17.1.1 在server.js中创建注册路由

#### 17.1.2 在客户端点击注册按钮，发送数据到服务器进行注册

#### 17.1.3 在server.js中创建登陆路由

#### 17.1.4 在客户端点击登陆按钮，发送数据到服务器进行登陆验证

### 17.2 店铺列表

#### 17.2.1 使用jquery的插件，分页插件

#### 17.2.2 在server.js中创建店铺列表对应的路由，要求传参pageNum;

#### 17.2.3 点击分页按钮，获取对应页的json数据；

#### 17.2.4 使用baiduTemplate或者artTemplate，店铺信息显示在页面上；

### 17.3 店铺详情页面

#### 17.3.1 请求商品列表json数据

#### 17.3.2 请求用户评价详情数据

#### 17.3.3 进行商品列表 用户评价的tab切换

### 17.4 购物车效果

#### 17.4.1 在店铺页面点击加入购物车按钮实现数据存储；

#### 17.4.2 在购物车页面把数据读取出来显示

#### 17.4.3 详细讲解localstorage或者sessionStorage进行页面传值

## day-18 git的基本使用

### 18.1 git的简介

### 18.2 git的使用

### 18.3创建git仓库

### 18.4创建github远程中央仓库

### 18.5克隆项目

### 18.6提交项目

### 18.7创建分支

### 18.8合并分支

### 18.9创建组织团队开发

### 18.10忽略文件

### 18.11解决冲突

### 18.12回滚版本

## day-19 高德地图(并且将其接入到项目中)

### 19.1 高德地图

#### 19.1.1 登陆高德地图的api官网，注册账号；

#### 19.1.2 学习高德地图的api

#### 19.1.3 创建地图

#### 19.1.4 创建地图控件（工具条 比例尺）

#### 19.1.5 创建地图标记market

#### 19.1.6 把地理位置转换成经纬度，创建对象的market显示在地图上

### 19.2 在同城帮项目应用地图

#### 19.2.1 点击“地图模式”，弹出地图；

#### 19.2.2 实现地图布局；

#### 19.2.3 给地图模块添加分页插件；

#### 19.2.4 点击分页按钮，移动地图，标记对应页数的店铺地址

#### 19.2.5 给地图上的店铺添加信息窗体

#### 19.2.6 实现”地点联想”搜索

## day-20 amd，cmd(规范)