动力节点-课程体系-V9.1.0

课程体系说明：标号 1 表示必须掌握；标号 2 表示能够理解；标号 3 表示简单了解；标号 4 表示扩展内容，根据学生学习情况自行决定是否讲解。

1. WEB 前端

1.1. HTML

1.1.1. HTML 概述

1.1.1.1. HTML 是什么

超文本标记语言（Hyper Text Markup Language），由标签组成，除支持普通文本之外，还支持流媒体（超文本）。

1.1.1.2. HTML 与 W3C

（1）W3C 是什么
（2）W3C 制定了 HTML 标准
（3）HTML 的版本（4.0 和 5.0）

1.1.1.3. HTML 怎么开发

（1）html 文件的扩展名是 html/htm
（2）使用普通的文本编辑器即可开发
（3）可以使用专业的开发工具
网页制作三剑客之一 DreamWeaver
HBuilder
其他...

1.1.1.4. HTML 怎么运行

直接采用浏览器打开执行

1.1.1.5. 世界五大主流浏览器介绍

3

IE
FireFox
Chrome
Opera
Safari

1.1.1.6. 安装 FireFox 和 Chrome 浏览器

2

1.1.1.7. 安装 HBuilder 开发工具

2

1.1.2. 第一个 HTML

1

（1）主要强调 HTML 的标签都是成对儿的，有开始标签，一般都会有对应的结束标签。
（2）强调代码的合理缩进
（3）强调 HTML 语法松散不严格
（4）强调 HTML 不区分大小写
（5）强调标签的属性，每个属性都包括属性名和属性值，属性值可以使用单引号括起来，也可以使用双引号括起来。
（6）强调 HTML 中的注释怎么写
（7）<meta charset="UTF-8"/>解决乱码问题
乱码产生是因为文件采用 UTF-8 方式，但浏览器打开该文件时采用 GBK 方式打开的，所以乱码
该行代码的作用就是告诉浏览器采用哪一种字符编码打开该文件。

1.1.3. 基本标签

1.1.3.1. 段落标记

1

1.1.3.2. 标题字

h1~h6

1.1.3.3. 换行

独目标记的概念。

1.1.3.4. 水平线

<hr/>

1.1.3.5. 预留格式

pre

1.1.3.6. 粗体字

1.1.3.7. 斜体字

1.1.3.8. 插入字

<ins></ins>

1.1.3.9. 删除字

<del></del>

1.1.3.10. 右上角加字

1.1.3.11. 右下角加字

1.1.3.12. font 标签

This is some text!

1.1.4. 实体符号

1.1.4.1. 空格

1.1.4.2. 大于号

>

1.1.4.3. 小于号

<

1.1.5. 表格

1.1.5.1. 基本表格

```
<table border="像素" width="像素或百分比" height="像素或百分比" align="center">
    <tr align="center">
        <td>test</td>
        <td>test</td>
        <td>test</td>
    </tr>
    <tr>
        <td align="center">test</td>
        <td>test</td>
        <td>test</td>
    </tr>
    <tr>
        <td>test</td>
        <td>test</td>
        <td>test</td>
    </tr>
</table>
```

1.1.5.2. 单元格合并

行合并：rowspan

列合并：colspan

1.1.5.3. th 标签

1.1.5.4. thead、tbody、tfoot 标签

1.1.6. 背景颜色和背景图片

bgcolor

background

1.1.7. 图片

（1）

（2）只设置图片的宽度，高度等比例缩放，不建议设置高度，设置高度容易失真。

1.1.8. 超链接

（1）超链接的作用

向服务器发送请求，链接到某个资源

（2）链接到网络中的某个资源

</a

（3）链接到本地的某个资源

（4）图片做超链接

<img/>

（5）超链接的 target 属性

_blank

_self

_parent

_top

（6）用户点击超链接和在浏览器地址栏上直接输入 URL 是完全相同的效果，只不过超链接更傻瓜式。

1.1.9. 列表

1.1.9.1. 有序列表

```
<ol type="1/A/a/I">
   <li>中国
      <ol>
         <li>北京</li>
         <li>天津</li>
         <li>上海</li>
      </ol>
   </li>
   <li>美国</li>
   <li>日本</li>
</ol>
```

1.1.9.2. 无序列表

```
<ul type="disc/circle/square">
   <li>中国
      <ul>
         <li>北京</li>
         <li>天津</li>
         <li>上海</li>
      </ul>
   </li>
   <li>美国</li>
   <li>日本</li>
</ul>
```

1.1.10. 表单

1.1.10.1. 表单起什么作用

用户填写表单，提交数据给服务器，所以表单是专门用来收集用户数据的。

1.1.10.2. 第一个表单

```
<form action="http://192.168.101.2:8080/crm/login"
用户名<input type="text" name="uname" />
密码<input type="password" name="pwd"/>
<input type="submit" value="登录"/>
<!--普通按钮不具备提交表单的能力-->
<input type="button" value="登录"/>
</form>
<!--submit 放到 form 外部无法提交表单-->
<input type="submit" value="登录"/>
```

（1）action 属性等同于超链接的 href 属性，填写请求的 url

（2）input 标签属于输入域标签，input 标签的 type 属性是 text，表示文本框，是 password，表示密码框

（3）input 标签的 type 是 submit 表示提交按钮，该按钮可以提交表单，所谓表单的提交是发送请求 url，并携带数据给服务器。

（4）所有按钮的 value 属性都是用来设置按钮上显示的文本内容

（5）发送请求并提交数据时，数据格式遵循 HTTP 协议，所有浏览器都会采用这种格式：url?name=value&name=value&name=value...，其中 name 是 input 标签的 name 属性，value 是 input 标签的 value 属性

（6）文本框和密码框的 value 不需要开发人员指定，用户填写的数据就是 value。

（7）submit 按钮放到 form 标签外面无法提交表单

（8）普通按钮不具备提交表单的能力。

1.1.10.3. 用户注册表单

表单项包括

（1）用户名

<input type="text" name="username" />，value 属性不需要写，用户填写的数据就是 value

（2）密码

<input type="password" name="pwd" />，value 属性不需要写，用户填写的数据就是 value

（3）性别

男<input type="radio" name="gender" value="m"/>，性别是单选按钮，用户只能选，所以该标签需要添加 value 属性

女<input type="radio" name="gender" value="f" checked/>，checked 默认选中

同一组的单选按钮，name 必须相同

提交给服务器的数据是：gender=m 或者 gender=f

（4）兴趣

运动<input type="checkbox" name="aihao" value="sport"/>

音乐<input type="checkbox" name="aihao" value="music" />

跳舞<input type="checkbox" name="aihao" value="dance" checked/>，checked 表示默认选中

同一组的复选框，name 相同

以上三项都被选中的话，提交的数据是：aihao=sport&aihao=music&aihao=dance

（5）学历

学历

```
<select name="xueli">
   <option value="gz">高中</option>
   <option value="zk">专科</option>
   <option value="bk" selected>本科</option>
</select>
```

selected 默认选中

当选中本科时，提交的数据为：xueli=bk

（6）简介

<textarea cols="列数" rows="行数" name="jianjie"></textarea>，文本域没有 value 属性，用户填写的内容就是 value

（7）注册按钮

<input type="submit" value="注册"/>：该标签放在 form 标签内部才起作用

（8）重置按钮

<input type="reset" value="重置" />

（9）再次强调表单提交时的数据格式

action?name=value&name=value&name=value&name=value...

这是 HTTP 协议中规定的，这些有规律的数据提交给服务器之后，以后服务器端的 java 程序要解析这段数据的。

form 表单的 method 属性

（1）method 不写或写上 get 都属于 get 请求，method 写 post 才是 post 请求

（2）get 请求在 HTTP 协议的请求行上提交数据，最终提交的数据会显示在浏览器地址栏上

（3）post 请求在 HTTP 协议的请求体中提交数据，最终提交的数据不会显示到浏览器地址栏上

（4）只有当使用 form 表单，并且 method 属性设置为 post 才是 post 请求，其他请求均为 get，超链接也是 get 请求。

1.1.10.4. 下拉列表怎么显示多个条目，怎么支持多选

（1）支持多选：multiple="multiple"
（2）显示多个条目：size="3"

1.1.10.5. file 控件

（1）文件上传时使用
（2）<input type="file"/>

1.1.10.6. hidden 控件

隐藏域控件，页面上看不到，但提交表单时会提交数据

1.1.10.7. readonly 与 disabled

（1）readonly：只读，不能修改，提交表单时数据会提交
（2）disabled：只读，不能修改，提交表单时数据不会提交

1.1.10.8. input 控件的 maxlength

maxlength 属性规定输入字段的最大长度，以字符个数计。

1.1.11. HTML 中元素的 id 属性

（1）讲述 HTML 文档是一棵树（DOM 树），树上有很多节点，每个节点一般都会有 id 属性，id 属性具有唯一性，在同一个文档中不能重复，id 是该节点的唯一标识。后期所学的 javascript 语言可以对 DOM 树上的节点进行增删改，达到动态效果。javascript 主要通过节点的 id 来获取该元素。

（2），超链接有 id；<form id=""></form>，form 表单有 id；<input type="text" id="username"/>，input 标签有 id。

1.1.12. div 和 span

（1）理解 div 是一种图层，div 主要使用在网页布局方面，通过后期所学的 CSS 可以设置 div 的宽度、高度、位置等样式。div 比 table 的布局更加灵活。

（2）div 图层可以嵌套使用

（3）默认情况下 div 独占一行，span 不会独占行。

1.2. CSS

1.2.1. CSS 的作用

层叠样式表语言，修饰 HTML，让 HTML 更好看，是 HTML 的化妆品

1.2.2. HTML 中嵌入 CSS 样式的三种方式

1.2.2.1. 内联定义方式

```
<div style="font-size:12px; text-align:center;">HTML 中引用 CSS 的行内式方法</div>
```

1.2.2.2. 样式块

（1）<style type="text/css"></style>

id 选择器

标签选择器

class 选择器

（2）CSS 的注释怎么写

1.2.2.3. 引入外部独立的 CSS 样式文件

```
<link href="css 文件路径" rel="stylesheet" type="text/css" />
```

1.2.3. 边框

```
（1）
div{
    border : 1px solid red;
}
（2）
div{
    border-width : 1px;
    border-style : solid;
    border-color : red;
}
```

1.2.4. 隐藏

```
div{
    display : none;
}
```

1.2.5. 字体

```
div{
    font-size : 12px;
    color : red;
}
```

1.2.6. 文本装饰

```
a{
    text-decoration : none;
}
```

```
a{
    text-decoration : underline;
}
```

1.2.7. 列表

```
ul{
    list-style-type : none;
}
```

1.2.8. 设置鼠标悬停效果

```
:hover
```

1.2.9. 定位

```
div{
    position : absolute;
    left : 100px;
    top : 100px;
}
```

1.2.10. 鼠标小手

```
div{
    cursor : pointer;
}
```

1.3. JavaScript

1.3.1. JavaScript 概述

（1）简称 JS

（2）一种脚本语言，脚本语言的特点

（3）JavaScript 和 JScript 的关系

（4）JavaScript 主要用来操作 HTML 中的节点，产生动态效果

（5）JavaScript 和 Java 的区别

1.3.2. JavaScript 包括三块：ECMAScript、DOM、BOM

（1）ECMAScript 是 ECMA 制定的 262 标准，JavaScript 和 JScript 都遵守这个标准，ECMAScript 是 JavaScript 核心语法

（2）DOM 编程是通过 JavaScript 对 HTML 中的 dom 节点进行操作，DOM 是有规范的，DOM 规范是 W3C 制定的。

（3）BOM 编程是对浏览器本身操作，例如：前进、后退、地址栏、关闭窗口、弹窗等。由于浏览器有不同的厂家制造，所以 BOM 缺少规范，一般只是有一个默认的行业规范。

1.3.3. 嵌入 JS 三种方式以及 JS 的注释

1.3.3.1. 行间事件

（1）<input type="button" value="hello" onclick="window.alert('hello js')" />

（2）JS 是一种基于事件驱动型的编程语言，当触发某个事件之后，执行一段代码

（3）JS 中的任何一个事件都对应一个事件句柄，例如鼠标单击事件 click，对应的事件句柄就是 onclick，事件句柄都是以标签的属性方式存在。在事件句柄后面可以编写 JS 代码，当触发这个事件之后，这段 JS 代码则执行了。

（4）JS 中的字符串可以使用单引号括起来，也可以使用双引号括起来

（5）window 是 JS 中的内置 BOM 顶级对象，代表当前浏览器窗口，window 对象有一个 alert()函数，该函数可以在浏览器上弹出消息框。

（6）JS 中的一条语句结束后可以使用“;”结尾，也可以不写。

（7）window.alert()中的 window.可以省略。

1.3.3.2. 页面 script 标签嵌入

（1）<script type="text/javascript">JS 代码</script>
（2）window.alert()的执行会阻塞当前页面的加载
（3）一个页面中可以写多个脚本块
（4）脚本块的位置没有限制
（5）暴露在脚本块中的 JS 代码在页面打开的时候遵循自上而下的顺序依次逐行执行

1.3.3.3. 外部引入

（1）<script type="text/javascript" src="js 文件路径"></script>
（2）<script type="text/javascript" src="js 文件路径">这里不能写 JS 代码</script>
（3）这种写法错误：<script type="text/javascript" src="js 文件路径"/>

1.3.4. 标识符和关键字

（1）标识符命名规则和规范按照 java 执行
（2）关键字不需要刻意记

1.3.5. 变量

1.3.5.1. 变量的声明与赋值

（1）变量未赋值，系统默认赋值 undefined
（2）JS 是一种弱类型编程语言，一个变量可以接收任何类型的数据
（3）一行上也可以声明多个变量

1.3.5.2. 函数的定义与调用

（1）函数类似于 java 语言中的方法，是一段可以完成某个功能的可以被重复利用的代码片段

（2）定义函数的两种语法

第一种：普通函数定义，这种方式较多

```
function 函数名(形式参数列表){
   函数体;
}
```

例如：

```
function sum(a, b){
   return a + b;
}
```

注意：

a 和 b 是形式参数列表，也是两个局部变量。

JS 中的函数不需要指定返回值类型，因为 JS 是弱类型编程语言，变量可以接收任何类型的数据，也就是说 JS 中的函数可以返回任何类型的数据，当然也可以不返回任何数据。返回数据使用 return 语句。

JS 中的函数在调用的时候，实参可以随意，例如调用以上的 sum 函数，可以这样调用：sum()，没有传任何实参的时候 a 和 b 变量没有赋值，则 a 和 b 都是 undefined。也可以这样调用 sum(10)，这样就表示 a 变量赋值 10，b 变量仍然是 undefined。还可以这样调用：sum(1,2)，这样则表示 a 是 1，b 是 2。

第二种：如果是把函数的声明当做类进行定义这种方式较多

```
函数名 = function(形式参数列表){
   函数体;
}
```

例如：

```
sum = function(a, b){
   return a + b;
}
```

（3）JS 中的函数定义在脚本块中，页面在打开的时候，函数并不会自动执行，函数是需要手动调用才能执行的。

（4）由于 JS 是一种弱类型编程语言，所以函数不能同名，没有重载机制

（5）这样的代码顺序是可以的，页面打开的时候会先进行所有函数的声明，函数声明优先级较高。

```
<script type="text/javascript">

   sayHello();
```

```
    function sayHello(){
        alert("Hello JS");
    }

</script>
```

（6）用户点击按钮，调用函数

```
<script type="text/javascript">
    function sayHello(){
        alert("hello js");
    }
</script>

<input type="button" value="hello" onclick="sayHello();"/>
```

1.3.5.3. 局部变量和全局变量

（1）局部变量：函数的形参是局部变量，另外使用 var 关键字在函数体中声明的变量是局部变量，函数执行结束之后，局部变量的内存就释放了。

（2）全局变量：在函数体外声明的变量属于全局变量，另外不使用 var 关键字声明的变量无论位置在哪，它都是全局变量，全局变量在浏览器关闭时销毁。

1.3.6. JS 数据类型

1.3.6.1. typeof 运算符

JS 中为什么会有 typeof 运算符

typeof 运算符怎么用，代码怎么写

语法格式是：

```
    function sum(a, b){
        if("number" === typeof a && "number"=== typeof b){
            return a + b;
        }
        alert("数据格式不合法");
```

```
    return 0;
}
```

typeof 运算符的运算结果都是全部小写的字符串

"undefined"
"number"
"string"
"boolean"
"object"
"function"

1.3.6.2. ES6 版本之前的数据类型有 6 种

Undefined

只有一个值 undefined，变量声明没赋值，系统默认赋值 undefined

Number

（1）Number 类型包括哪些值：0,1，-1,3.14,12,300，NaN，Infinity
（2）parseInt()函数
（3）parseFloat()函数
（4）Math.ceil()函数：向上取整
（5）isNaN()函数

String

（1）可以使用单引号，也可以用双引号
（2）JS 中的字符串包括小 String，也包括大 String，小 String 属于原始类型，大 String 是 JS 的内置对象，大 String 属于 Object 类型。
（3）无论大 String 还是小 String，它们的属性和方法都是通用的。
（4）字符串中常用方法讲一些，主要讲解字符串的 substr()和 substring()的区别。

Null

（1）该类型只有一个值：null

（2）typeof 运算符的执行结果是"object"

Boolean

（1）只有两个值：true 和 false

（2）Boolean()函数

（3）JS 中的 if 语句自动调用 Boolean()函数。

Object

（1）JS 中如何定义一个类。

（2）JS 中如何创建一个对象。

（3）JS 中如何访问对象属性，调用对象的方法。

（4）JS 中的一个函数，既是函数声明，又是类的定义，同时函数名也可以看做构造方法名。直接调用函数表示普通函数调用，如果使用 new 运算符来调用该函数则会创建对象。

（5）使用 prototype 属性动态的给对象扩展属性以及方法。

1.3.6.3. ES6 版本及之后包括的数据类型有 7 种

除了以上 6 种类型之外，还有一种类型叫做：Symbol

1.3.7. null NaN undefined 区别

（1）=、==、===三者的区别

（2）null NaN undefined 三者类型不同，null 和 undefined 的值可以等同

1.3.8. JS 中的事件

1.3.8.1. 常用事件

（1）blur 失去焦点
（2）change 下拉列表选中项改变，或文本框内容改变
（3）click 鼠标单击
（4）dblclick 鼠标双击
（5）focus 获得焦点
（6）keydown 键盘按下
（7）keyup 键盘弹起
（8）load 页面加载完毕
（9）mousedown 鼠标按下
（10）mouseover 鼠标经过
（11）mousemove 鼠标移动
（12）mouseout 鼠标离开
（13）mouseup 鼠标弹起
（14）reset 表单重置
（15）select 文本被选定
（16）submit 表单提交

1.3.8.2. 注册事件的两种方式

（1）在标签中使用事件句柄的方式注册事件

```
<body onload="sayHello()"></body>
```

（2）在页面加载完毕后使用 JS 代码给元素绑定事件

```
<script>
    window.onload = sayHello;
</script>


<script>
    window.onload = function(){


    }
</script>
```

重点：通过事件注册，理解回调函数的概念

1.3.8.3. 代码的执行顺序

这是一种错误的写法：

```
<body>
    <script type="text/javascript">
        var elt = document.getElementById("btn");
    </script>
    <input type="button" id="btn" value="mybtn"/>
</body>
```

这样写：

```
<body>
    <input type="button" id="btn" value="mybtn"/>
    <script type="text/javascript">
        var elt = document.getElementById("btn");
    </script>
</body>
```

或者这样写：

```
<body>
    <script type="text/javascript">
        window.onload = function(){
            var elt = document.getElementById("btn");
        }
    </script>
    <input type="button" id="btn" value="mybtn"/>
</body>
```

1.3.8.4. 通过 keydown 事件演示回车键 13，ESC 键 27

1.3.9. JS 运算符之 void

运算符就讲这一个，告诉学生其它运算符和 java 一样用。void 主要讲：javascript:void(0)的用法。

1.3.10. JS 之控制语句

告诉学生控制语句和 Java 一样用，课堂上不再讲解。只讲一下 for..in 语句的使用，使用 for..in 语句遍历数组，以及遍历一个对象的属性。

1.3.11. JS 内置对象

1.3.11.1. Array

（1）创建数组
（2）JS 中的数组特点
（3）JS 中数组对象常用方法：push，pop，join，reverse 等。
（4）数组遍历

1.3.11.2. Date

（1）new Date() 获取当前系统时间
（2）new Date().getTime()获取时间戳
（3）new Date().getFullYear()、getMonth()等方法。

1.3.12. BOM 和 DOM 的区别与联系

1.3.13. DOM 编程案例

1.3.13.1. innerHTML innerText 操作 div 和 span

1.3.13.2. JS 的正则表达式(Regular Expression)

正则表达式概述

（1）正则表达式是一门独立的学科，不止用在 JS 中

（2）正则表达式专门用来做字符串格式匹配的

常用的正则表达式符号

参考 30 分钟入门正则表达式：

^　字符串开始

$　字符串结束

\s　空白

*　0~N 次

+　1~N 次

?　0 或 1 次

{3}　3 次

{3,}　3~N 次

{3,5}　3~5 次

(a|b)　a 或 b

[a-z]　a 到 z

[^abc]　不是 abc

会写简单的正则表达式

（1）qq 号正则

（2）必须由数字和字母组成，不能含有其它符号的正则

（3）给学生一些常用的正则表达式

会创建 JS 中的正则表达式对象

（1）var regExp = new RegExp("^[1-9][0-9]{4,}$");

（2）var regExp = /^[1-9][0-9]{4,}$/;

会调用 JS 中正则表达式对象的 test()函数

写一个校验用户名只能由数字和字母组成的案例

1.3.13.3. 表单验证

（1）用户名不能为空
（2）用户名必须在 6-14 位之间
（3）用户名只能有数字和字母组成，不能含有其它符号（正则表达式）
（4）密码和确认密码一致，邮箱地址合法。
（5）统一失去焦点验证
（6）错误提示信息统一在 span 标签中提示，并且要求字体 12 号，红色。
（7）文本框再次获得焦点后，清空错误提示信息，如果文本框中数据不合法要求清空文本框的 value
（8）最终表单中所有项均合法方可提交

1.3.13.4. 复选框全选和取消全选

document.getElementById()
document.getElementsByName()
document.getElementsByTagName()
以上三个函数告知学生很重要

1.3.13.5. 获取下拉列表选中项的 value

change 事件

1.3.13.6. 显示网页时钟

window.setInterval()
window.clearInterval()
主要两个函数
捎带着提一下 window.setTimeout()

1.3.13.7. 拼接 html 的方式，设置 table 的 tbody

```
<table>
    <thead></thead>
    <tbody id="userListTbody"></tbody>
</table>
<script>
    var html = "";
    html += "<tr>";
    html += "<td>";
    html += "zhangsan";
    html += "</td>";
    html += "<td>";
    html += "2000-10-11";
    html += "</td>";
    html += "</tr>";


    html += "<tr>";
    html += "<td>";
    html += "lisi";
    html += "</td>";
    html += "<td>";
    html += "2001-10-11";
    html += "</td>";
    html += "</tr>";


    var userListTbody = document.getElementById("userListTbody");
    userListTbody.innerHTML = html;
<script>
```

1.3.13.8. 要求学生在此之后会使用浏览器的 F12

会使用 F12 调试面板：第一会调错；第二会定位并查看 HTML 页面元素。

1.3.14. BOM 编程案例

1.3.14.1. window.open()和 window.close()

1.3.14.2. window.alert()和 window.confirm()

1.3.14.3. 如果当前窗口不是顶级窗口，将当前窗口设置为顶级窗口

```
if(window.top != window.self){
    window.top.location = window.self.location;
}

// ??????????????????
if(window.top != window.self){
    window.top = window.self;
}
```

1.3.14.4. 历史记录

```
window.history.back(); window.history.go(-1); window.history.go(1);
```

1.3.14.5. window.location.href

提示一下 document.location.href 也可以完成同样功能

1.3.15. JSON 对象

1.3.15.1. eval 函数

1.3.15.2. 怎么创建 json 对象以及访问 json 对象的属性

1.3.15.3. json 在开发中有什么用

数据交换作用

1.3.15.4. 写一个这样的 JSON

描述一个班级的总人数，另外包括描述班级中每个学生的信息，这样的 JSON 怎么写

1.3.15.5. JS 中[]和{}的区别

1.3.16. 总结一下浏览器向服务器发送请求的常见方式

（1）直接在浏览器地址栏上写 URL，get 请求。

（2）点击页面超链接，get 请求。

（3）提交 form 表单，可以是 get，也可以是 post。

（4）window.open(url);

（5）window.location.href=url;

（6）document.location.href=url;