# 尚马教育 JAVA WEB课程

# Javascript1

文档编号：B03

创建日期： 2017-06-13

最后修改日期：2019-09-23

版 本 号：V3.0

电子版文件名：尚马教育-第二阶段-3.javascript.docx

**文档修改记录：**

|  |  |  |  |
| --- | --- | --- | --- |
| 更新日期 | 更新作者 | 更新说明 | 版本号 |
| 2017-07-30 | 张元林 | 初始版本 | V1.0 |
| 2019-09-13 | 郭雪岩 | 添加验证及正则 | V2.0 |

**主讲人：**

**郭雪岩**

目录

[1. Javascript介绍](#_Toc24594_WPSOffice_Level1) [3](#_Toc24594_WPSOffice_Level1)

[1.1. 语言特性](#_Toc9603_WPSOffice_Level2) [3](#_Toc9603_WPSOffice_Level2)

[1.2. 发展历史](#_Toc13091_WPSOffice_Level2) [3](#_Toc13091_WPSOffice_Level2)

[1.3. Tips](#_Toc25018_WPSOffice_Level2) [3](#_Toc25018_WPSOffice_Level2)

[1.4. Javascript组成](#_Toc7486_WPSOffice_Level2) [4](#_Toc7486_WPSOffice_Level2)

[2. Javascript语法](#_Toc9603_WPSOffice_Level1) [5](#_Toc9603_WPSOffice_Level1)

[2.1. Javascript引入方式](#_Toc27591_WPSOffice_Level2) [5](#_Toc27591_WPSOffice_Level2)

[2.2. Javascript基本语法](#_Toc14367_WPSOffice_Level2) [5](#_Toc14367_WPSOffice_Level2)

[2.2.1. 注释](#_Toc9603_WPSOffice_Level3) [5](#_Toc9603_WPSOffice_Level3)

[2.2.2. 变量](#_Toc13091_WPSOffice_Level3) [5](#_Toc13091_WPSOffice_Level3)

[2.3. 运算符](#_Toc14771_WPSOffice_Level2) [7](#_Toc14771_WPSOffice_Level2)

[2.4. 条件语句](#_Toc5582_WPSOffice_Level2) [7](#_Toc5582_WPSOffice_Level2)

[2.5. 循环语句](#_Toc24808_WPSOffice_Level2) [7](#_Toc24808_WPSOffice_Level2)

[2.6. 函数](#_Toc7055_WPSOffice_Level2) [8](#_Toc7055_WPSOffice_Level2)

[2.7. Js事件](#_Toc10535_WPSOffice_Level2) [9](#_Toc10535_WPSOffice_Level2)

[2.8. 正则表达式](#_Toc21347_WPSOffice_Level2) [9](#_Toc21347_WPSOffice_Level2)

## Javascript介绍

### 语言特性

* JavaScript（简称js）是一种描述语言，基于对象和事件驱动的脚本语言
* JavaScript特点
* 脚本语言（一种轻量级的编程语言）
* 一种解释性语言（无需预编译）
* 被设计为向HTML页面添加交互行为
* 运行于客户端（浏览器）

### 发展历史

* 95年LiveScript（Netscape公司）
* Netscape将其更名为JavaScript
* 微软IE3.0并附带JScript、CEnvi的ScriptEase以及JavaScript三足鼎立
* 97年JavaScript1.1版本被ECMA（欧洲计算机制造商）标准化为ECMAScript的脚本语言

### Tips

* JavaScript与ECMAScript的关系

ECMAScript是JavaScript的标准和基础

* JavaScript与JAVA的关系

雷锋和雷峰塔的关系

JavaScript和Java都与sun公司有合作，是借势Java

### Javascript组成

|  |
| --- |
|  |

* 什么是ECMAScript

ECMAScript定义了脚本语言的所有属性、方法和对象

包括语法、类型、关键字、保留字、运算符、对象等

除了JavaScript外，同时也是Nombas的ScriptEase和Flash脚本ActionScript的基础

* 什么是DOM

Js中将整个文档对象（html文件）描述成树状模型结构，有元素节点、属性节点、文档节点等等。节点之间有父子关系 js可以通过描述出的节点及关系，动态的操作节点和节点属性。

* 什么是BOM

Js可对浏览器窗口进行访问和操作

例如：弹出新的浏览器窗口，移动、关闭浏览器窗口及调节浏览器窗口大小，WEB浏览器详细的定位对象

## Javascript语法

### Javascript引入方式

* Script标签中编写
* 通过script标签引入（当前标签无法再编辑js代码）
* 在页面中直接编写javascript

例：onclick="javascript:alert(1)" 尽量不用

### Javascript基本语法

#### 注释

JavaScript的注释方法，与Java一致

//单行

/\*

多行

\*/

#### 变量

变量声明规则与java基本一致

|  |
| --- |
| 1、由于 JavaScript 对大小写敏感，变量名也对大小写敏感 （y 和 Y 是两个不同的变量）  2、变量必须以字母或下划线或美元符$开始 ，不能使用关键字作为变量名  //JS的变量声明 ，是通过 var 语句来声明  //弱类型，声明时无需指定数据类型，赋值时根据值决定类型  var num = 10 ; |

Js中变量类型

|  |
| --- |
| * Number 数字类型 * String 字符串类型 * Boolean 布尔类型 * Undefined 未定义或未初始化类型 * Null 表示空对象 一般作为对象类型的初始值 * Object 复合类型 包含js中的各种对象 |

Typeof关键字 可以返回当前变量的类型

|  |
| --- |
| typeof运算符返回值如下：  undefined：变量被声明后，但未被赋值  string：用单引号或双引号来声明的字符串  boolean：true或false  number：整数或浮点数  function：函数方法  object：对象 或 数组 或 Null 类型 |

类型转换函数

|  |
| --- |
| parseInt():字符串转换为整数  parseFloat():字符串转换为浮点数  toString()：返回字符串  注意parseInt parseFloat对字符串的转换 |

### 运算符

|  |
| --- |
|  |

Js中运算符与java中基本一致

+可以做数字加 也可以做字符串拼接

注意：小数相加时会丢失精度 如0.1+0.2 需要取位数 使用toFixed（位数）

==与===的区别

### 条件语句

条件语句语法与java基本一致

If...else

Switch...case

### 循环语句

循环语句语法与java基本一致

For

for...in 可循环数组成员

While

Do...while

### 函数

* 一组可以通过调用运行的语句

编写为函数后，页面加载时函数中的代码不会立即运行，调用时才运行

* 分类
* 系统函数(parseInt();)
* 自定义函数(mytest();)
* 系统函数

|  |
| --- |
| parseInt  parseFloat  IsNaN  eval |

* 自定义函数

|  |
| --- |
| function 函数名(参数1,参数2,参数3){  //代码  }  等同于  var 函数名 = function(参数1,参数2,参数3){  //代码  }  Js中的函数不需要定义返回值类型 如果需要返回值 在函数中加上return关键字即可  function 函数名(参数1,参数2,参数3){  //代码  return xxx;  } |

注意1：特殊写法 立即执行函数

(function(){

})();

注意2：js中作用域，有函数作用域与全局

### Js事件

Js会自动对行为进行监听，当触发某些行为时，可以调用自定义函数来完成用户交互。

事件通常与函数配合使用，当事件发生时去调用函数

事件举例：

|  |
| --- |
| * 鼠标点击 onclick * 鼠标悬浮于页面的某个热点之上 onmouseover * 在表单中选取输入框 onfocus * 内容改变 onchange * 页面或图像载入 onload * 键盘按键 onkeyup * 表单提交 onsubmit |

### 正则表达式

在js中 经常用正则表达式配合进行前端校验，给用户相应提示

语法：/ reg /

两个特殊的符号'^'和'$'。他们的作用是分别指出一个字符串的开始和结束。

例：/^xxxxxx$/

* 字符匹配符

|  |
| --- |
| （1）[…]：匹配方括号中包含的字符集中的任意一个字符。例如，正则表达式“[abc]”，字符“a”、“b”、“c”都可以与之匹配。  （2）[^…]：匹配方括号中未包含的任何字符。例如，正则表达式“[^ab]”，除了字符“a”、“b”之外,其他的任何字符都可以与之匹配。  （3）[a-z]，[1-9]：匹配指定范围内的任何字符。  （4）[^a-z]，[^1-9]：匹配不在指定的范围内的任何字符。  （5）\d：匹配任意一个数字字符，等效于[0-9]。  （6）\D：匹配任意一个非数字字符，等效于[^0-9]。  （7）\s：匹配任何空白字符，包括空格、制表符等。  （8）\S：匹配任何非空白字符。  （9）\w：匹配任何英文字母、数字和下划线，等效于[A-Za-z0-9\_]。  （10）\W：匹配任何非英文字母和数字类字符，但不包括下划线，等效于[^A-Za-z0-9\_]。  （11）.：匹配除换行符\n之外的任何单字符。 |

* 次数限定符

|  |
| --- |
| （1）{n}:出现n次  （2）{n,}：规定前面的元素或组合项至少出现n次。例如“o{2,}”，字符串“bod”不能与之匹配，但“foood”可以与之匹配。  （3）{n,m}：规定前面的元素或组合项至少连续出现n次，至多连续出现m次，其中n≤m，逗号和数字之间不能有空格。例如“o{1,3}”，字符串“food”和字符串“foood”都能与之匹配。  （4）+：规定前面的元素或组合项必须出现一次或连续出现多次，等效于{1,}。  （5）\*：规定前面的元素或组合项可以出现零次或连续多次，等效于{0,}。例如“zo\*”，字符串“z”和“zoo”都能与之匹配。  （6）?：规定前面的元素或组合项出现零次或一次，等效于{0,1}。例如“zo?”,字符串“z”和“zo”都能与之匹配。 |

* 常见正则表达式

|  |
| --- |
| 1．匹配身份证号码：^\d{15}$|^\d{18}$  2．邮箱：^\w+@\w+.[a-zA-Z]{2,3}(.[a-zA-Z]{2,3})?$;  3．中国的邮政编码：^\d{6}$  4．匹配字符串中的中文字符：[\u4e00-\u9fa5]  5．验证5到12位的qq号：^\d{5,12}$  6.验证手机号 ^1(3|4|5|7|8)\d{9}$ |

配合正则表达式，编写验证功能。

|  |
| --- |
| 有邮箱和手机号两个输入框  验证通过时显示绿色对勾 不通过显示红色提示  两个框单独验证  验证全部正确时 表单可以提交 否则表单不允许提交 |