Web开发（二）

--- 1-2 JS基础语法

河北师范大学软件学院
Software College of Hebei Normal University

JavaScript语法概述

- 语法特点：
  - 类C语言
  - 弱类型：变量的数据类型可以任意转换
  - 动态类型：变量创建时不用指定数据类型

内容提纲

- JavaScript 基础语法
- JavaScript 变量及原始数据类型
- JavaScript 流程控制结构
- JavaScript 的基本代码规范
- 调试工具的使用

JS语句

- JS语句：对于浏览器而言，语句就是命令，它告诉浏览器要做什么

  如：document.write(" Hello! ");

- 语句通常以分号结束

  建议使用分号；

- 一系列能被浏览器执行的语句构成JS程序

JS语句

```
<script type="text/javascript">
        alert("Hello World!");
        alert("Hello JavaScript!");
        alert("Hello WebDev");
        alert("Hello ...");
</script>
```

demo1-2-1

JS语句块

- 语句块：多个语句可放在 “{”和“}”内，形成一个语句块
- JavaScript代码的执行次序与书写次序相同
- 语句块举例

```
<script type="text/javascript">
        if (true) {
                document.write('第一条语句执行</br>');
                document.write('第二条语句执行');
        }
</script>
```

demo1-2-2

JS注释

- 可以添加注释来对 JavaScript 进行解释，提高代码的可读性。JavaScript 不会执行注释。
- JS单行注释：//
- JS多行注释：/* */
- 在 HTML 插入注释： <!-- 注释内容 -->
- 在CSS中插入注释： /* */

内容提纲

- JavaScript 基础语法
- JavaScript 变量及原始数据类型
- JavaScript 流程控制结构
- JavaScript 的基本代码规范
- 调试工具的使用

字面值

- 1、2、1.0、3.1415926...
- 'hello'、"world"、 "34" ...
- true、false

变量的创建和说明

- JS为动态类型语言，声明/创建变量时，不需指明数据类型
  - var 变量名（ var 变量名=初值； ）
  - 字符串值用 ‘ ’ 或 “ ” 引起来；
- 变量名的规范
  - 变量名区分大小写;
  - 变量名以字母或 '_' 或 '$' 开头；
  - 变量名不能是关键字，保留字；

原始数据类型

- JavaScript是一种弱类型的语言
- 弱类型是指不同类型的变量之间可以相互赋值，但在某一时刻，一个变量存在某一种数据类型
  - 5 种原始数据类型：Number、String、Boolean、Undefined、Null
  - 获得变量在某一时刻的数据类型，使用typeof运算符

demo1-2-3

原始数据类型

- Number类型：1、3.1415926、1e6
- String类型：用 ‘ ’ 或 “ ” 引起一组字符
  - 如：'hello'、"world"、 "34"
- Boolean类型：true 或 false
- Undefined类型：只有一个值 undefined
- Null类型：只有一个值 null

原始数据类型

```
<script type="text/javascript">
        var a = '15';
        var i = 1;
        var pi = 3.1415;
        var name = "JavaScript";
        var id = "201201034";
        var isExists = true;
        var c = i+"";
        alert( typeof isExists);
</script>
```

demo1-2-4

原始数据类型

- 与C语言的区别：
  - Number：无整数和浮点数之分
  - String：无字符串和字符之分
  - Boolean：C 中无 Bool 型

运算符

- 算术：+、-、*、/、%、++、--
- 字符串连接：+
- 赋值：=、+=、-=、*=、/=、%=
- 比较：==、===、!=、>、<、<=、>=

  ==：值相等则为 true

  ===：类型和值都须相同则为 true
- 逻辑：与(&&)、或(||)、非(!)
- 条件：*变量名* = (*条件*)？*值1*：*值2*

使用“+”连接字符串

```
<script type="text/javascript">
        var x=3;
        var y="3";
        var z=5;
        z += y;
        var a=y+z;
        document.write(x+y+'<br/>');
        document.write(z+'<br/>');
        document.write(a+'<br/>');
        document.write(x+y+z+'<br/>');
</script>
```

demo1-2-5

比较运算符

```
<script type="text/javascript">
    var x = 3;
    var y = 3;
    var z = “3”;
    alert(x == y); //改为：x === z
</script>
```

条件运算符

```
<script type="text/javascript">
    var a = 39;
    var b = 30;
    document.write(a>=b? "a大于等于b":"a小于b");
</script>
```

demo1-2-6

运算符

- 23 + " 2 " = ? —— 232
- 15/2 = ? —— 7.5
- 23 – true = ? —— 22
- " 95 " == 95 —— true
- " 95 " === 95 —— false
- typeof 75 —— number

数据类型转换

- 变量在进行运算时，可能会发生隐式类型转换
  - 转换成 String 类型：用 + 连接

    如：var sum = ‘img’ + 3 + ’ .jpg’；

  - 转换成 Boolean 类型：变量之前加 !!

img3.jpg

- 可以对变量进行显式类型转换
  - 转换为 Number 类型：parseInt()、parseFloat()、Number()
  - 转换为 String 类型：String()
  - 转换为 Boolean 类型：Boolean()

数据类型转换

```
<script type="text/javascript">
// 转换成 Number 类型
        var a = '123.456img';
        var a1 = parseInt(a);
        var a2 = parseFloat(a);
        document.write("a1=" + a1 +'<br/>'+ "a2=" + a2+'<br/>');
// 转换成 String 类型
        var b = 3.1415926;
        var b1 = b + "";
        document.write(typeof(b1)+'<br/>');
// 转换成 Boolean 类型
        var c = 'img' + 3 + '.jpg';
        var c1 = !!c;
        document.write('c1='+c1)
        alert(typeof c1);
</script>
```

demo1-2-7

运算符左右数据类型转换规则

- +左右出现字符串时，作为字符串连接运算符使用
- -、*、/、%左右出现字符串（布尔）时，将字符串（布尔）转换为数值类型
- 比较运算符左右出现数值和字符串时，会将字符串转换为数值，出现布尔类型时，会将布尔类型转换为数值类型
- 逻辑运算符会将数据类型转换为布尔类型之后再做运算

数据类型转换规则

- 字符串转数值：
  - 从左开始截取字符串中出现数字，直到遇到非数字字符
- 数值转布尔：
  - 0为false，其他为true
- 布尔转数值：
  - false为 0 , true为 1
- 字符串转布尔：
  - 空字符串为false，其他为true

内容提纲

- JavaScript 基础语法
- JavaScript 变量及原始数据类型
- JavaScript 流程控制结构
- JavaScript 的基本代码规范
- 调试工具的使用

选择与分支语句

- If 语句
- if…else 语句
- if…else if…else 语句
- switch 语句

选择与分支语句

```
<script>
    var box = 100;
    if (box >= 100) {  //如满足条件，不执行下面任何分支
        alert('甲');
    } else if (box >= 90) {
        alert('乙');
    } else if (box >= 80) {
        alert('丙');
    } else if (box >= 70) {
        alert('丁');
    } else if (box >= 60) {
        alert('及格');
    } else {                    //如果以上都不满足，则输出不及格
        alert('不及格');
    }
</script>
```

demo1-2-8

循环语句

- for循环
- while循环
- do...while循环

循环语句

动手做：demo1-2-9

- 使用for循环，向文档中动态写入一个4行4列的表格，表格单元格内容为
- 使用while循环，向文档中动态写入一个4行4列的表格，表格单元格内容为

循环语句（终止循环）

- 如何终止循环？
  - 终止循环：break ;
  - 跳过本次循环： continue ;

demo1-2-10

内容提纲

- JavaScript 基础语法
- JavaScript 变量及原始数据类型
- JavaScript 流程控制结构
- JavaScript 的基本代码规范
- 调试工具的使用

代码规范的重要性

- 方便代码的交流和维护。
- 不影响编码的效率，不与大众习惯冲突。
- 使代码更美观、阅读更方便。
- 使代码的逻辑更清晰、更易于理解。

JavaScript代码规范

JavaScript基本规范

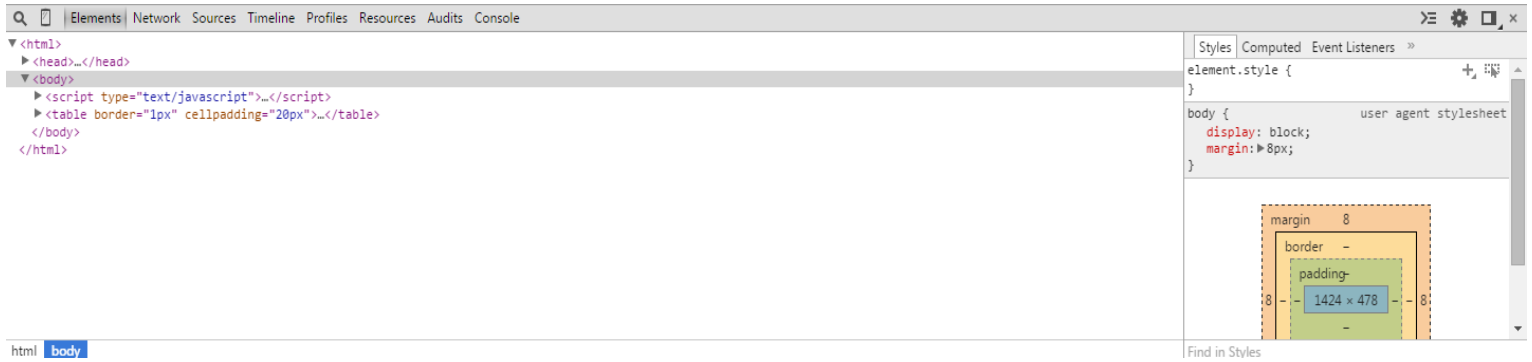
- 变量定义
  - 尽量使用 var 关键字定义（否则会被当成全局变量）
  - 尽量减少全局变量的使用
  - 变量名定义要有意义
- 单行程序，以分号结束
- 缩进和注释

内容提纲

- JavaScript 基础语法
- JavaScript 变量及原始数据类型
- JavaScript 流程控制结构
- JavaScript 的基本代码规范
- **调试工具的使用**

调试工具的使用

- 谷歌开发者工具



- firebug

调试工具的使用

- 控制台输出：console.log( )
- 跟踪程序
  - 监控错误
  - 添加断点
  - 单步、连续执行代码
  - 退出调试

小结

- JS中的语句和语句块
- 变量和原始数据类型
- JavaScript的基本代码规范
- 使用开发者工具调试JavaScript

Thank You !