JS

张利

简介

- JavaScript在1995年由Brendan Eich发明， 1997成为ECMA的标准。
- 发布于2015年的ECMAScript 6（es6）是JavaScript的最新版本。
- JavaScript 和 Java是完全不同的语言， 名字中包含Java只是为了蹭互联网的热度。

JS 引用

- 放入<head>或者<body>中。
- 放入独立的.js 文件中。
  - <script src="xxx.js"></script>

  - 优势：
    - 易于阅读和维护。
    - 可以利用缓存，来加速页面的加载。

JS 基本语法

- 每个语句以;结束, 但是并不是强制要求， 因为浏览器会在每个语句后自动补上。
  - 一条语句： var x = 1;
  - 两条语句： var x = 1; var y = 2;
  - 一条语句，多个变量: var x =1, y = 2;(可跨行，只要用，结尾)
- 语句块用{...}
- 注释：行注释：// 和 块注释：/* ... */

```
// 这是一行注释
alert('hello'); // 这也是注释

/* 从这里开始是块注释
   仍然是注释
   仍然是注释
注释结束 */
```

- 区分大小写

JS 输出

- 使用alert()或者window.alert()输入到警告框。
- 使用document.write()输入到html页面。
  - 注意：在 HTML 文档完全加载后使用 document.write() 将覆盖所有已有的 HTML， 比如条件触发。
- 使用innerHTML写入到html元素。
- 使用console.log()写入到控制台。

命名规范

- 推荐： 以小写字母开头的驼峰大小写。
- 构造变量名称（唯一标识符）的通用规则是：
  - 名称可包含字母、数字、下划线和美元符号
  - 名称必须以字母开头
  - 名称也可以 $ 和 _ 开头（但是在本教程中我们不会这么做）
  - 名称对大小写敏感（y 和 Y 是不同的变量）
  - 保留字（比如 JavaScript 的关键词）无法用作变量名称
-

JS 数据类型

- Number, javascript 不区分整形和浮点型，统一用Number来表示。
- 合法的Number类型有：
  - 123; // 整数123
  - 0.456; // 浮点数0.456
  - 1.2345e3; // 科学计数法表示1.2345x1000，等同于1234.5
  - -99; // 负数
  - NaN; // NaN表示Not a Number，当无法计算结果时用NaN表示
  - Infinity; // Infinity表示无限大，当数值超过了JavaScript的Number所能表示的最大值时，就表示为Infinity
  - 0xff00 //16进制数据
- 字符串： 用单引号或者双引号括起来的任意的文字。
  - length // 返回长度
  - indexOf('x', [n]) //指定文本首次出现的位置
  - lastIndexOf('x', [n]) //指定文本最后一次出现的位置
  - search() //和index一样，但是不能设置起始位置，但是可以用正则表达式
  - slice(start, end) //字符串截取
  - toUpperCase() , toLowerCase()
  - str.replace('A', 'B') // 默认只替换第一个， g
  - concat()
- 布尔值： true或者false

数据类型

- 日期：
  - 默认情况下，JavaScript 将使用浏览器的时区并将日期显示为全文本字符串。
  - 创建：
    - new Date()
    - new Date(year, month, day, hours, minutes, seconds, milliseconds)
    - new Date(milliseconds)
    - new Date(date string)
  - 获取：
    - var date = new Date();
    - date.getXX()

JS 数据类型

- 数组，JavaScript的`Array`可以包含任意数据类型，并通过索引来访问每个元素。
  - var arr = [1, 2, 3.14, 'Hello', null, true];
- 数组的操作：
  - 取长度：arr.length
  - 可以给length赋值，这样对改变数组的长度（截取或者增加undefined值）
  - 通过索引更改原有的值
  - indexof： 搜索一个指定元素的位置
  - slice(x,[y])：数组截取
    - 起止参数包括开始索引，不包括结束索引。
    - 如果不给slice()传递任何参数，它就会从头到尾截取所有元素。利用这一点，我们可以很容易地复制一个Array。
  - push(x, [y])：末尾添加若干元素
  - pop(): 删除最后一个元素
  - unshift(x,[y]): 向头部插入元素
  - shift(): 从头部删除元素

JS 数据类型

- 数组的操作：
  - sort()： 按照默认的顺序对数组进行排序。
  - reverse(): 对数组元素进行反转。
  - concat(): 数组拼接， 不改变原数组而是返回一个新数组。
  - jion(): 按照一定的规则把数组拼接起来，返回一个字符串。
- 多维数组： 如果数组的某个元素又是一个Array，则可以形成多维数组。
- 数组练习：
  - 在新生欢迎会上，你已经拿到了新同学的名单，请排序后显示：欢迎XXX，XXX，XXX和XXX同学！
  - var arr = ['小明', '小红', '大军', '阿黄'];
  -

JS 数据类型

- 对象：对象是一种无序的集合数据类型，它由若干键值对组成。
- JavaScript用一个{...}表示一个对象，键值对以xxx: xxx形式申明，用,隔开。
- 访问属性是通过.操作符完成的，但这要求属性名必须是一个有效的变量名。如果属性名包含特殊字符，就必须用’’括起来，并且不能用.来访问。

```
var xiaohong = {
    name: ‘小红',      //有效变量，可以用xiaohong.name 或者 xiaohong[‘name’]来访问
    'middle-school': 'No.1 Middle School’ //不是有效变量，只能用xiaohong[‘middle-school’] 来访问
};
```

- 可以包含方法

JS 数据类型

- Map：一组键值对的结构，具有极快的查找速度。

```
var m = new Map([['Michael', 95], ['Bob', 75], ['Tracy', 85]]);
m.get('Michael'); // 95
m.has(‘Michael’);//true
m.set(‘Eric’, 100)
m.delete(‘Bob’)
```

- Set: 是一组key的集合，但不存储value。由于key不能重复，所以，在Set中，没有重复的key。

```
var s = new Set([1, 2, 3, 3, ‘3']);
s // Set {1, 2, 3, “3"}
s.add(5)
s.delete(5)
```

JS 语法

- 条件判断
- 循环：
  - for
  - for … in …
  - while
  - do while

函数

- JavaScript允许传入任意个参数而不影响调用，因此传入的参数比定义的参数多也没有问题，虽然函数内部并不需要这些参数。传入的参数比定义的少也没有问题。
- arguments：它只在函数内部起作用，并且永远指向当前函数的调用者传入的所有参数

```
function foo(x) {
    console.log('x = ' + x); // 10
    for (var i=0; i<arguments.length; i++) {
    console.log('arg ' + i + ' = ' + arguments[i]); // 10, 20, 30
     }
}
foo(10, 20, 30);
```

- 匿名函数(函数表达器)

const vs let vs var

- const和let是ES6的新关键字。
- var是方法作用域， let是块作用域， 在申明它之外的{}, 不能引用，如果引用会抛错。
- const是块作用域，但是代表常量， 一旦赋值，不能改变。
- 可以对常量对象或者数组的属性或者值做更改，但是不能指定到另外一个对象或数组。

调试

- 主流浏览器都支持javascript调试。
  - step over
  - step into
  - step out

操作表单

- 用JavaScript来操作表单，可以获得用户输入的内容，或者对一个输入框设置新的内容。

```
// <input type="text" id="email">
var input = document.getElementById('email');
input.value; // ‘用户输入的值’
Input.value=‘diaolanshan@aliyun.com'
```

- 对于单选款(radio) 或者多选框，使用checked而不是value来获取值。

操作DOM

- 文档定位。
  - 方式一：
    - document.getElementById()
    - document.getElementsByTagName()
    - document.getElementsByClassName()
    - element.children //element下所有的子元素（直属）
    - element.firstElementChild/lastElementChild //第一个或者最后一个元素
  - 方式二：
    - querySelector()
    - querySelectorAll()
    -

操作DOM

- 更新文档
  - innerHTML, 修改DOM节点内文本内容，或者直接通过HTML片段修改DOM节点内部的子树
  - innerText， 修改DOM节点的文本，相对于innerHTML, 更加安全。
  - 通过元素属性更新节点
    - p.style.color = ‘#ff0000';

- 插入DOM
  - appendChild和innerHTML的区别?
  - document.createElement();
  - insertBefore()， 语法：parentElement.insertBefore(newElement, referenceElement);

- 删除元素
  - 首先得到要删除的节点，获取其父节点（element.parentElement;）再调用删除方法（parent.removeChild(element)）

JSON

- JSON 指的是 JavaScript 对象标记法（JavaScript Object Notation）
- JSON数据转换为JS 对象： JSON.parse()

```
var myJSON = '{ "name":"Bill Gates", "age":62, "city":"Seattle" }';
var myObj =  JSON.parse(myJSON);
```

- JavaScript 对象进行字符串化

```
var obj = { name:"Bill Gates", age:62, city:"Seattle"};
var myJSON = JSON.stringify(obj);
```

-