

Web开发（二）

--- 第五章 对象



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

内容提纲

- 对象简介
- 对象相关操作
- 构造函数
- 内置对象
- "一切皆对象"

内容提纲

- **对象简介**
- **对象相关操作**
- **构造函数**
- **内置对象**
- **"一切皆对象"**

对象简介

- 什么是对象？
 - 对象是人們要進行研究的任何事物
 - 使用对象
 - 不需要明白内部原理，会使用接口即可
 - 创建对象
 - 对象是一个整体，对外提供一些接口



对象简介

- 程序语言的分类
 - 面向过程的程序语言（举例：C语言）
 - 面向对象的程序语言（举例：C++、java）
- JavaScript 是一门基于对象的语言
 - 具有面向对象的部分特征
 - 在 JavaScript 中，一切皆对象



从现实社会谈起

- 某个具体的实物是一个对象 (Object)
 - 比如：一只狗
- 对象具有一些属性和方法



名字叫做Lili

品种：哈士奇

体重：60斤

会做跳跃运动

在程序中实现属性和行为

• 属性(properties) → 状态的, 静态的

— 通过变量来表示

— 例:

```
var name = "Lili";  
var breed = "Husky";  
var weight = 60;
```

• 方法(methods) → 过程的, 动态的

— 通过函数来实现

— 例:

```
function jump() { ... }  
function run() { ... }
```



对象的实质

- 这些描述同一个实物的数据分散存在，如何将它们统一在一起？

- JavaScript 中的对象

- 是一系列相关属性和方法的集合
 - 属性: 对象相关特征，静态的
 - 方法: 对象相关行为，动态的
- 是一种数据类型

```
var dog = {  
    name: "Lily" ,  
    breed: "Husky" ,  
    weight: 60 ,  
    jump: function(){  
        //some action.....  
    }  
}
```


内容提纲

- 对象简介
- 对象相关操作
- 构造函数
- 内置对象
- "一切皆对象"



对象相关操作

- 定义对象
- 访问
- 添加
- 修改
- 删除
- 遍历



对象相关操作

• 定义对象

- 使用一对**大括号**表示对象，属性和方法写在括号内
- 属性包括属性名和属性值，之间使用**冒号**分隔
- 属性和属性之间使用**逗号**分隔
- 方法同属性类似，只是值部分为一个函数体

```
var objName = {  
    attrName: attrValue ,  
    methodName: function(){ //some code... } ,  
    .....  
}
```

最后一个属性值后面**不加**逗号



对象相关操作

- 访问对象属性和方法

- 方式一：通过 对象名.属性名 、 对象名.方法名()
- 方式二：通过 对象名["属性名"]

- 添加对象属性

- 对象创建之后，要向对象添加属性，只需要为新属性赋值即可

对象相关操作

- 修改对象属性
 - 直接为原有属性赋新值即可
- 删除对象属性
 - 使用 `delete` 关键字

对象相关操作

- 遍历对象属性

- 使用 for in 语句

```
for (var key in objName) {  
    // key 变量，在循环中代表一个属性键  
    // 通过 objName[ key ] 访问到相应的属性值  
}
```

this 关键字

- this 关键字

- 在对象方法内部使用，指代当前对象

```
var dog = {  
  name: "Lili",  
  loves: ['eat', 'sleep', 'play'],  
  jump: function(){  
    console.log(this);  
    console.log("这条狗的名字为"+this.name);  
  }  
};  
dog.jump();
```

```
► {name: "Lili", loves: Array(3), jump: f}  
这条狗的名字为Lili
```



实例

- 天堂电影院最近播放《007》和《指环王》两部电影。使用 JavaScript 程序的对象分别来存储这两部电影的数据，包括：标题、类别、评分、播放时间（每天播放 3 次，时间自定）。
- 思考并写出这两个对象

实例

- 扩展电影对象，增加一项功能
 - 能够显示该电影下一次播放的时间
 - 即增加一个 getNextShowing 方法

实例中存在的问题

代码的可重用性低！！

- 扩展方法时，每个对象都要写一个功能类似的方法
- 如果再增加一个电影，结构相同的对象需要再定义一次



思考

- 有没有优化的方法?  函数

```
function movie(title, type, rating, showtime) {  
    var movie = {}; //原料  
    movie.title = title; //加工  
    movie.type = type;  
    movie.rating = rating;  
    movie.showtime = showtime;  
    movie.getNextShowing = function() { ... };  
    return movie; //输出  
}  
  
var movie1 = movie("007", "谍战", 8, ["8:00", "13:00", "16:00"]);  
var movie2 = movie("指环王", "战争", 9, ["9:00", "14:00", "19:00"]);
```



内容提纲

- 对象简介
- 对象相关操作
- **构造函数**
- 内置对象
- "一切皆对象"



构造函数

- 能够创建出对象的特殊函数
 - 类似一个工厂一样，能够生产对象
 - 我们需要在工厂中设定“产品（对象）”的“模板（结构）”
 - 在“生产产品（创建对象）”时指定产品的“具体参数（属性值）”

电影构造函数

- 电影构造函数定义

```
function Movie(title, type, rating, showtime) {  
  this.title = title; //加工  
  this.type = type;  
  this.rating = rating;  
  this.showtime = showtime;  
  this.getNextShowing = function() { ... };  
}
```

原料和输出?

- 电影构造函数调用

原料和输出!

```
var movie1 = new Movie("007", "谍战", 8, ["8:00", "13:00", "16:00"]);  
var movie2 = new Movie("指环王", "战争", 9, ["9:00", "14:00", "19:00"]);
```



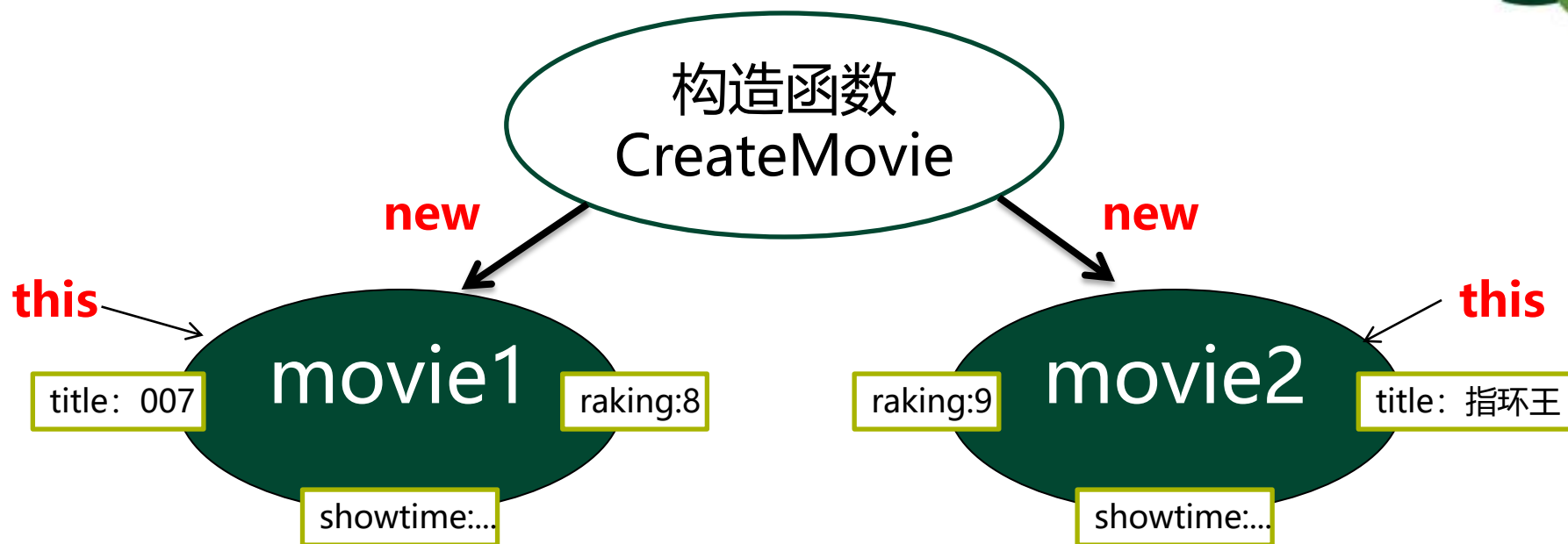
构造函数的定义

- 使用 function 关键字声明
 - 为了区别普通函数，构造函数首字母大写
- 在构造函数中，this 指代实例对象
 - 通过 this 定义和访问实例对象的属性和方法

构造函数的调用

- 通过 **new** 关键字调用构造函数
- new 关键字的作用
 - 创建原料: 隐式创建一个空对象, 并且 this 指代这个空对象
 - 产品输出: 隐式将加工过的对象 return 返回

理解构造函数



- 构造函数相当于模板，使用 new 关键字创建实例对象
- 构造函数调用时，this 指向创建的实例对象

内容提纲

- 对象简介
- 对象相关操作
- 构造函数
- 内置对象
- "一切皆对象"



内置对象

- JavaScript 提供了多种内置对象
 - String
 - Array
 - Math
 - Date
 - ...



JavaScript 中的对象

- JavaScript 对象分类

- 宿主对象

- 由 JavaScript 解析器所嵌入的宿主环境定义的 (如: window, document, event)

- 内置对象

- 由 ECMAScript 规范定义的对象或构造器对象 (如: Array、String)

- 自定义对象

- 用户自定义 JavaScript 代码创建的对象 (如: dog、movie)

String 对象与 Array 对象

- String 对象处理字符串相关操作

- 创建 string 实例对象，例：

```
var str = new String( "some string here" );
```

```
var str = "some string here" ;
```

- Array 对象处理数组相关操作

- 创建 array 实例对象，例：

```
var movies = new Array( "007" , "指环王" );
```

```
var movies = [ "007" , "指环王" ] ;
```

Math 对象

- Math 对象包含了一系列的数学运算的功能

- Math 对象不需要创建，直接使用

- 常用操作

- 四舍五入: Math.**round**()

- 向下取整: Math.**floor**() 向上取整: Math.**ceil**()

- 生成 0~1 随机数: Math.**random**()

- 取最大值: Math.**max**() 取最小值: Math.**min**()

思考 Math 同 Array、String
在内部实现时的不同?

Date 对象的使用

- Date 对象包含了一系列的日期时间处理的功能
 - 创建 Date 对象, 例: `var now = new Date();`
- 常用操作
 - 获取当前日期时间: `toLocaleString()`
 - 获取年份、月份、日期: `getFullYear()`、`getMonth()`、`getDate()`
 - 获取小时、分钟、秒钟: `getHours()`、`getMinutes()`、`getSeconds()`

参考文档

- 关于内置对象的更多使用方法，可参考：
 - http://www.w3school.com.cn/js/js_obj_date.asp
 - http://www.w3school.com.cn/js/js_reference.asp

内容提纲

- 对象简介
- 对象相关操作
- 构造函数
- 内置对象
- "一切皆对象"



数据类型

- JavaScript 数据类型

- 原始类型

- number、string、boolean、null、undefined

- 对象类型

- 对象具有属性和方法

- 除了 undefined，JavaScript的一切都是对象或者可以看成对象

内容回顾

- JavaScript 的对象就是一系列相关属性和方法的集合
 - 属性的实质是数据，在程序中用变量保存，是静态的
 - 方法的实质是行为，在程序中用函数实现，是动态的
- 创建对象
 - { }， 括号内部为属性名：属性值或方法名：方法体，属性和属性间用逗号分隔，最后一个属性值不加逗号
- 对象属性的相关操作: 增、删、改、查、遍历



内容回顾

- this 关键字
 - 在对象方法内部，使用 this 指代当前对象
- 构造函数
 - 定义：函数名称首字母大写，内部使用 this 指代 实例对象
 - 调用：new 关键字调用构造函数
 - new 关键字的作用
- 对象分类：宿主对象、内置对象、自定义对象

内容回顾

- 内置对象
 - String、Array、Boolean、Number、Date、RegExp、JSON、Math
 - 内置对象的属性和方法
- "JavaScript 中一切皆对象"
 - 对象具有属性和方法

The background of the slide is decorated with numerous overlapping circles in various shades of green and yellow, scattered across the top and right sides.

Thank You!