

JavaScript+jQuery前端开发基础教程

微课版

第5章 JavaScript的面向对象

本章主要内容：

- 对象
- 原型对象和继承
- 内置对象
- 类

5.1 对象

- 在JavaScript中，基本类型（number、string、boolean、null、undefined和symbol）本身并不是对象。
- JavaScript的内置对象（String、Number、Boolean、Object、Function、Array、Date、RegExp和Error等）都是是对象的一个子类型。
- 在面向对象的程序设计中，类封装了对象的共同属性和方法。属性表示对象的特征，方法表示对象的行为。具体的对象称为类的实例对象，继承了类的所有属性和方法。

- 虽然JavaScript不是纯粹面向对象，但同样支持面向对象的特性
- 。JavaScript的对象同样有属性和方法，也支持继承。一个对象可拥有多个属性和方法，并可继承原型对象的属性和方法。对象的属性可看作一个“键/值”对，键是属性名，值是属性的值。一个对象就是多个属性名到值的映射，这类似于其他程序设计语言中的“映射”“散列”“字典”等概念。
- 对象的属性和方法均通过对象访问。
 - `x = event.type` //使用事件对象的属性，获得事件类型名称
 - `event.preventDefault()` //调用事件对象的方法，阻止事件默认行为

5.1.1 创建对象

- JavaScript提供了3种创建对象的方法：
 - 字面量
 - new关键字
 - Object.create()方法

1. 使用字面量创建对象

- 在JavaScript中，花括号括起来的多个“键/值”对是一个对象常量，可将其赋给一个变量。

- `var x = {} //创建一个空对象`

- `var a = { name: 'JavaScript程序设计', price: 25 } //创建一个有name和price属性的对象`

2. 使用new关键字创建对象

- new关键字调用构造函数来创建并初始化一个对象。JavaScript的内置对象都包含内置的构造函数。
- 例如，Object()、Array()、Date()等都是构造函数。

```
var a = new Object()
```

```
//创建空对象
```

```
var b = new Object({ name: 'JavaScript程序设计', price: 25 }) //创建带有属性的对象
```

```
var c = new Array(1, 2, 3) // 创建一个数组对象
```

```
var d=new Date()
```

- 在JavaScript中，通过字面量创建的所有对象都有相同的原型对象，对象继承原型对象的属性和方法，可使用`Object.prototype`属性引用原型对象。
- 使用`new`关键字和构造函数创建对象时，实质是使用构造函数的`prototype`属性值作为原型对象。例如，`new Array()`以`Array.prototype`属性值作为原型对象，`new Date()`以`Date.prototype`属性值作为原型对象

3. 使用Object.create()方法创建对象

- Object.create()方法用指定参数创建对象，
 - 参数为null时，创建一个空对象；
 - 参数为对象常量或其他对象时，将参数作为原型对象来创建对象，新对象继承原型对象的所有属性和属性值。

```
var a = Object.create(null) //创建一个空对象
```

```
var b = Object.create({ name: 'jQuery教程', price: 30 }) //提供原型对象来创建对象
```

5.1.2 使用对象属性

- 对象属性使用“.”运算符来访问，“.”左侧为引用对象的变量名称，右侧为
- 属性名。也可用类似于数组元素的方式来访问属性。
 - `var a = {name:'C++',price:12}`
 - `document.write(a.name)`
 - `document.write(a['name'])`
- 两条语句中的[a.name](#)和[a\['name'\]](#)是等价的。如果读取一个不存在或者未赋值的属性，得到的值为undefined。

- 对象的属性是动态的。在给对象属性赋值时，如果属性存在，则覆盖原来的值，否则会为对象创建新的属性并赋值。

- ```
var a = { name: 'C++', price: 12 }
```

- ```
a.name = 'HTML' //修改属性值
```

- ```
a.nmae = 'JavaScript' //本意是为name属性赋值，输入错误，这会创建新的nmae属性
```

- 可使用delete删除对象的属性。

- ```
delete a.name
```

- 可使用for/in循环来遍历对象的属性。

5.1.3 对象的方法

- 对象的方法就是通过对象调用的函数。
- 在方法中可用this关键字来引用当前对象。
- 将函数赋给对象属性，该属性即可称为方法，通过该属性来引用函数。
- 作为方法使用的属性，可称为方法属性

5.1.4 构造函数

- 构造函数是一个特殊的方法。在构造函数中，使用 `this` 关键字访问当前对象。
- 构造函数需要和 `new` 关键字一起使用，以便创建并初始化对象。

5.1.5 with语句

- with语句的基本语法格式如下。

- with(对象) {

- 语句

- }

- 在with语句的代码块中，可直接使用对象的属性和方法，而不需要“对象名.”作为前缀。

5.2 原型对象和继承

- 对象和原型对象之间是一种“继承”的共享关系，对象继承原型对象的所有属性和方法。
- 一条基本的原则：除了原型对象外，对象的属性总是“私有的”，只属于当前对象。给不存在的属性赋值时，总是为对象创建该属性。
- 在读取一个对象属性时，如果对象没有该属性，则查看原型对象是否有该属性。如果有，则使用原型对象的属性值，否则得到undefined。

5.3 内置对象

- JavaScript常用内置对象有Array（数组）对象、Math（数学）对象、Number（数字）对象、Date（日期）对象和String（字符串）对象。
- 本节主要介绍Math对象、Date对象和String对象。

5.3.1 Math对象

- Math对象定义了常用的数学函数和常量，它没有构造函数。Math对象的主要属性和方法如下。
 - Math.E: 返回数学常量 e 。
 - Math.LN10: 返回10的自然对数。
 - Math.LN2: 返回2的自然对数。
 - Math.LOG10E: 返回以10为底 e 的对数。
 - Math.LOG2E: 返回以2为底 e 的对数。
 - Math.PI: 返回圆周率。

- `Math.SQRT1_2`: 返回2的平方根的倒数。
- `Math.SQRT2`: 返回2的平方根。
- `Math.abs(x)`: 返回x的绝对值。
- `Math.sin(x)`: 返回x的正弦值。
- `Math.cos(x)`: 返回x的余弦值。
- `Math.tan(x)`: 返回x的正切值。
- `Math.acos(x)`: 返回x的反余弦值。
- `Math.asin(x)`: 返回x的反正弦值。
- `Math.atan(x)`: 返回x的反正切值。

- `Math.ceil(x)`: 返回大于或等于x的最小整数。
- `Math.exp(x)`: 返回e的x次方。
- `Math.floor(x)`: 返回小于或等于x的最大整数。
- `Math.log(x)`: 返回x的自然对数。
- `Math.max(x, y, ……)`: 返回参数中的最大值。
- `Math.min(x, y, ……)`: 返回参数中的最小值。
- `Math.pow(x, y)`: 返回x的y次方。
- `Math.random()`: 返回一个等于或大于0、小于1的随机数。
- `Math.round(x)`: 返回x的四舍五入值，0.5向上取整。例如，2.5舍入为3，-2.5舍入为-2。

5.3.2 Date对象

- Date对象的构造函数如下。
 - `Date()`：创建表示当前日期时间的Date对象。
 - `Date(msecond)`：创建整数msecond表示的Date对象。msecond为要创建的日期距离1970年1月1日00:00:00的毫秒值。
 - `Date(datestring)`：用日期时间字符串datestring创建Date对象。
 - `Date(year, month, day, hour, minute, second, msecond)`：创建指定了年、月、日、小时、分钟、秒和毫秒的Date对象。

- Date对象的常用方法如下。
 - `getFullYear()`：返回日期的四位年份（4位整数）。
 - `getMonth()`：返回日期的月份（0-11），1月为0。
 - `getDate()`：返回日期的日数（1-31）。
 - `getDay()`：返回星期几（0-6），星期日为0。
 - `getHours()`：返回小时数（0-23）。
 - `getMinutes()`：返回分钟数（0-59）。
 - `getSeconds()`：返回秒数（0-59）。
 - `getMilliseconds()`：返回毫秒数（0-999）。
 - `getTime()`：返回当前时间，自1970年1月1日以来的毫秒数。

- `setYear()`: 设置日期的年。
- `setMonth()`: 设置日期的月。
- `setDate()`: 设置日期的日。
- `setDay()`: 设置星期几。
- `setHours()`: 设置小时数。
- `setMinutes()`: 设置分钟数。
- `setSeconds()`: 设置秒数。
- `setMilliseconds()`: 设置毫秒数。
- `setTime()`: 用距1970年1月1日的毫秒数来设置时间。

- `toString()`: 将Date对象转换为字符串。
- `toLocaleString()`: 将Date对象转换为本地字符串。
- `toDateString()`: 将Date对象转换为只含日期的字符串。
- `toLocaleDateString()`: 将Date对象转换为只含日期的本地字符串。
- `toTimeString()`: 将Date对象转换为只含时间的字符串。
- `toLocaleTimeString()`: 将Date对象转换为只含时间的本地字符串。

5.3.3 String对象

▪ 1. 构造函数

- String对象的构造函数为String(s)，new String(s)创建一个保存字符串的对象，类型为object。
- 参数s不是字符串时，JavaScript会将其转换为字符串。
- 内置函数String(s)与String对象的构造函数同名，它将参数s转换为普通字符串，类型为string。

2. String对象属性

- length属性用于返回字符串对象中保存的字符个数。
 - `var n = "abc".length` //n的值为3
- 这说明字符串“abc”是对象吗？答案是否定的。在执行该语句时，JavaScript会隐式地将字符串“abc”转换为String对象，然后通过对象返回length属性。

3. String对象方法

- `charAt(n)`: 返回字符串中的第n个字符，第1个字符位置为0。
- `charCodeAt(n)`: 返回字符串中第n个字符的Unicode。
- `concat(value1, value2, ...)`: 将参数提供的多个值按顺序添加到当前字符串末尾，返回新的字符串。
- `indexOf(s, start)`: `s`为要查找的字符串，`start`为搜索开始位置（可省略）。方法从给定位置开始在原字符串中搜索给定字符串，返回该字符串第1次出现的位置。省略搜索位置时，从第1个字符开始搜索。如果不包含给定字符串，返回值为-1。
- `lastIndexOf()`: 与`indexOf()`方法类似，返回给定字符串最后一次出现的位置。
- `replace(a, b)`: 将字符串中与`a`匹配的字符替换为`b`中的字符串。`a`可以是一个正则表达式对象，`a`具有全局属性`g`时，替换所有匹配的字符串，否则只替换第1个匹配字符串。`a`为简单字符串时，也只替换第1个匹配字符串。

- `search(a)`: 在字符串对象中查找与a匹配的子字符串。若a不是正则表达式对象，会先将其转换为正则表达式对象。如果包含匹配的字符串，返回第1个匹配的字符串位置，否则返回-1。
- `slice(start,end)`: 返回字符串中从start位置开始的，end之前（不包含end）的子字符串。参数为负数时，从字符串末尾开始计算位置。-1表示字符串最后一个字符。
- `split(dm,len)`: 使用dm指定的分隔符将字符串分解为字符串数组，数组最多len个元素。len省略时，分解整个字符串。
- `substring(m,n)`: 与`slice()`类似。区别在于，`substring()`将两个参数中的较小值作为开始位置，将另一个参数作为结束位置。
- `toLowerCase()`: 将字符串中所有字母转换为小写。
- `toUpperCase()`: 将字符串中所有字母转换为大写。

4. 使用String对象将字符串转换为HTML标记的方法

- `anchor()`: 将字符串转换为<a>标记, 参数作为标记name属性的值。
- `bold()`: 将字符串转换为标记。
- `italics()`: 将字符串转换为<i>标记。
- `strike()`: 将字符串转换为<strike>标记。
- `fixed()`: 将字符串转换为<tt>标记。
- `fontcolor()`: 将字符串转换为标记, 设置颜色。
- `fontsize()`: 将字符串转换为标记, 设置字号。
- `link`: 将字符串转换为<a>标记, 参数作为标记href属性的值。
- `sub`: 将字符串转换为<sub>标记。

5.4 类

- 在JavaScript中，对象是特定属性的集合，类则是同一类对象的共享属性和方法的集合。
- JavaScript的类使用基于原型的继承机制，继承同一个原型的所有对象是同一个类的实例（也称实例对象或类的成员）。
- JavaScript一直允许定义类，只是没有明确类的概念。ES6增加了类的相关语法，包括class关键字。

5.4.1 使用工厂函数定义类

- 早期的JavaScript支持使用工厂函数创建对象，工厂函数可看作类的定义。工厂函数通过原型对象来定义类的属性和方法。

5.4.2 使用构造函数定义类

- JavaScript可使用new关键字调用构造函数创建新对象，对新对象进行初始化。
- 构造函数的prototype属性值作为新对象的原型，这也说明使用同一个构造函数创建的所有对象均继承同一个原型，都是同一个类的实例对象。

5.4.3 使用class关键字定义类

- ES6增加了class关键字，使JavaScript具有了类似于Java和C++的类定义方式。class关键字定义类的基本语法格式如下。

- class 类名 {
- constructor(参数) { //定义类的构造函数
- ... //使用参数初始化新对象的属性
- }
- ... //定义类的方法
- }

5.4.4 为类添加和修改方法

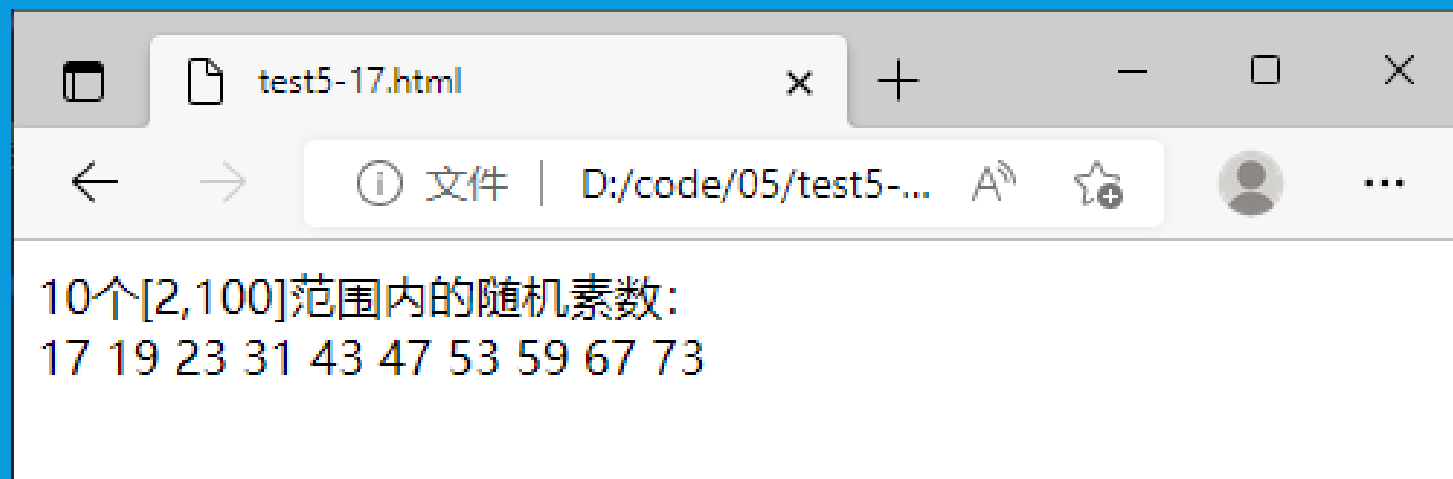
- JavaScript基于原型的继承机制，允许通过修改类的prototype属性为类的原型对象添加和修改属性，从而实现为类添加和修改方法。
- 不管是否在修改原型对象之前创建实例对象，所有实例对象均可继承添加和修改的属性。

5.4.5 子类

- 在面向对象中，通过扩展类A得到类B，则类A是父类，类B是子类。
- 子类可继承父类的所有方法。如果子类定义了与父类同名的方法，则子类的方法覆盖父类的同名方法，这称为方法的重载。
- JavaScript使用extends关键字定义子类，在子类中可使用super关键字调用父类的构造函数。
- 在子类中使用super关键字调用父类的构造函数，需要注意的是，子类应在使用this关键字之前调用父类的构造函数。如果子类的构造函数未调用父类构造函数，JavaScript会自动添加super()，以便调用父类的构造函数。

编程实践：输出随机素数

- 本节综合应用本章所学知识，使用JavaScript脚本在浏览器中输出10个100以内的随机素数，按从小到大的顺序输出，如图5-17所示。



- `Math.random()` 方法返回 $[0, 1)$ 范围内的一个随机数。

- 返回 $[a, b)$ 范围内的随机整数可使用下面的语句。

- `var x = parseInt((b - a + 1) * Math.random()) + a`

JavaScript+jQuery前端开发基础教程

微课版

第6章 浏览器对象

本章主要内容:

- Window对象
- Document对象
- Form对象

6.1 Window对象

- Window（窗口）对象是客户端JavaScript程序的顶级全局对象，它代表当前浏览器窗口，所有的其他全局对象、函数、变量自动成为Window对象的成员。

6.1.1 Window对象层次结构

- window（小写）关键字用于引用当前窗口的Window对象。
- 每个Window对象均有一个document属性，用于引用窗口中代表Web文档的Document对象。
- Document对象的forms数组包含了文档中的所有Form（表单）对象。可用下面的表达式引用第1个表单。
 - `window.document.forms[0]`
- 每个浏览器窗口中的所有对象构成了以Window对象为根节点的层次结构，通过Window对象可引用当前窗口和文档中的所有对象。

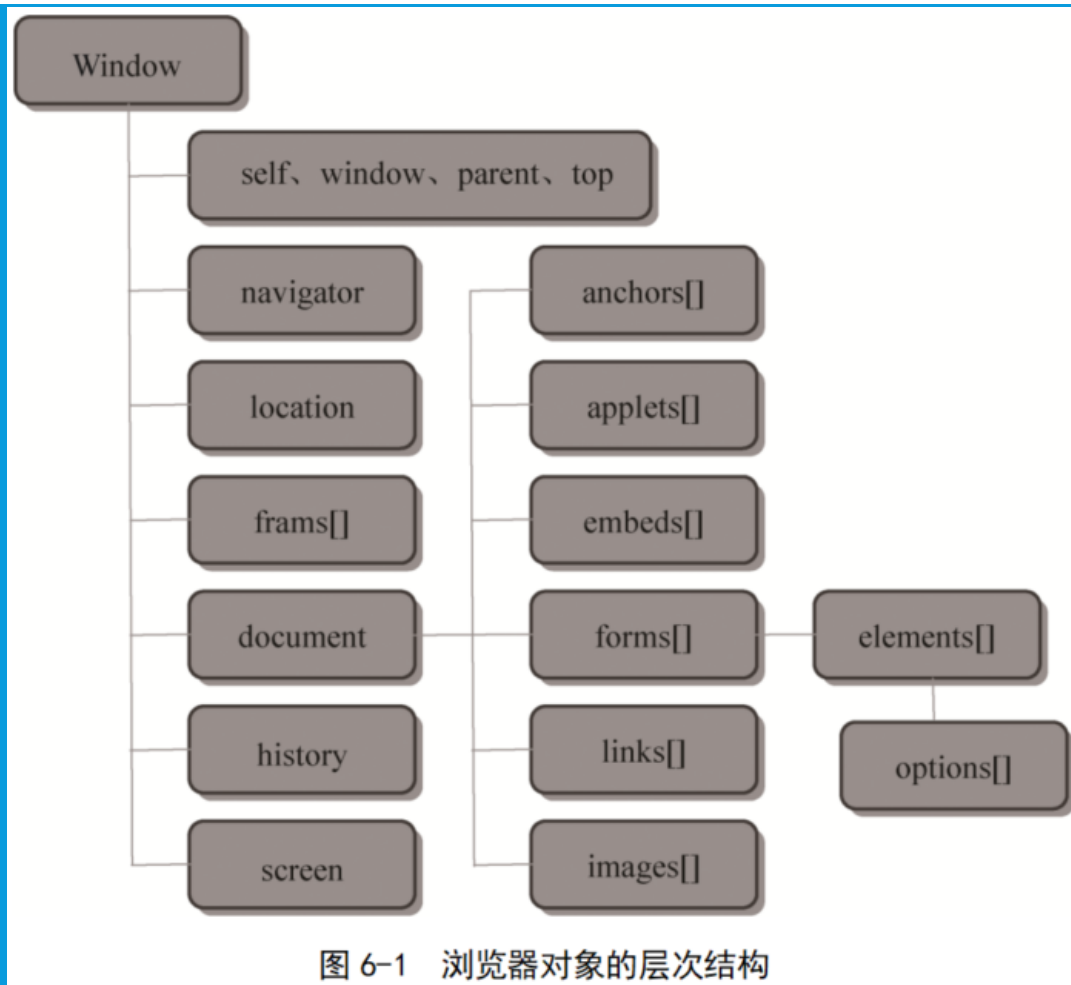


图 6-1 浏览器对象的层次结构

6.1.2 Window对象的常用属性和方法

- 1. Window对象常用属性
 - defaultStatus: 设置或返回浏览器状态栏显示的默认信息。
 - status: 设置或返回浏览器状态栏显示的即时信息。
 - document: 引用Document对象。
 - navigator: 引用包含客户端浏览器信息的Navigator对象。
 - frames: 窗口中所有框架对象的集合。
 - history: 引用表示浏览器历史的History对象。
 - location: 引用表示浏览器URL的Location对象。
 - name: 设置或返回窗口名称，窗口名称可作为<a><form>等标记的target属性值。

2. Window对象的常用方法

- `alert()`: 显示警告信息对话框。
- `confirm()`: 显示确认对话框。
- `prompt()`: 显示输入对话框。
- `blur()`: 使窗口失去焦点，即成为非活动窗口。
- `focus()`: 使窗口成为活动窗口。
- `close()`: 关闭窗口。
- `createPopup()`: 创建一个弹出式窗口。
- `setInterval()`: 设置经过指定时间间隔执行的函数或计算表达式。

- `clearInterval()`: 取消由`setInterval()`方法设置的定时时间。
- `setTimeout()`: 设置在指定的毫秒数后执行的函数或计算表达式。
- `clearTimeout()`: 取消由`setTimeout()`方法设置的指定时间。
- `moveBy()`: 相对于窗口当前坐标移动指定的像素。
- `moveTo()`: 将窗口左上角移动到指定的坐标。
- `open()`: 在一个新的浏览器窗口或已打开的命名窗口中打开URL。
- `print()`: 输出当前窗口内容。
- `resizeBy()`: 按照指定的像素调整窗口的大小。
- `resizeTo()`: 把窗口的大小调整到指定的宽度和高度。
- `scrollBy()`: 按照指定的像素来滚动内容。
- `scrollTo()`: 把内容滚动到指定的坐标。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
<https://d.book118.com/856144041151010231>