# 21.3.31学习记录

## 一、对象

1、使用工厂方法创建对象，通过该方法可以大批量创建对象

```
function  creatPerson（）{
    //创建一个新的对象
    var  obj  =  new  Oject（）；
    //向对象中添加属性

    //将新的对象返回
    return obj;
}
```

**使用工厂方法创建的对象，无法区分对象的类型**

2、创建一个构造函数，专门用来创建Person对象的；

注意：构造函数就是一个普通的函数，创建方式和普通函数没有区别；不同的是构造函数习惯上首字母大写。还有构造函数需要使用new关键字使用

执行流程：

①立刻创建一个新的对象

**②将新建对象设置为函数中的this！！！！！所以在构造函数中可以用this来引用新建的对象！！！**

③逐行执行函数中的代码

④将新建的对象作为返回值返回

注意：使用同一个构造函数创建的对象，我们称为一类对象，也将一个构造函数称为一个类；用构造函数构造的对象称为该类的**实例；**

**可以使用instanceof可以检查一个对象是否是一个类的实例：**

**对象 instanceof 构造函数**

**而所有的对象都是Object的后代，所以instanceof检查均返回true；**

**构造函数内部写一个方法会在每次执行时都会创建一个独立唯一的方法，这样有些浪费内存，因为方法的功能完全一样；**

**可以在全局作用域中定义共享的方法，但是这样污染了全局作用域的命名空间而且定义在全局作用域中也很不安全；**

## 二、原型 prototype

前言：我们创建的每一个函数，解析器都会向函数中添加一个属性prototype；

这个属性对应着一个对象，这个对象就是原型对象

当函数以构造函数的形式调用时，它所创建的对象中都会有一个隐含的属性，**指向该构造函数的原型对象**，我们可以通过_proto_来访问该属性

原型对象就相当于一个公共的区域，所有同一个类的实例都可以访问到这个原型对象，**我们可以将对象中共有的内容，统一设置到原型对象中！**

**当我们访问对象的一个属性或方法时，会先在对象自身中寻找，如果有则直接使用；如果没有就去原型里找！**

**总结：创建构造函数时，将这些对象共有的属性和方法，统一添加到构造函数的原型对象中！！不用分别添加也不会污染全局作用域！**

**补充：使用in检查对象中含有的属性时，对象内没有但原型中有，也返回true**

**使用对象的hasOwnProperty（）来检查对象自身是否含有属性**

**使用该方法只有当对象自身中含有属性时才会返回true**

**原型对象也是对象，所以它也有原型，当我们使用一个对象的属性或方法时，查找流程是自身——原型对象——原型的原型；直到找到**

**Object对象的原型，因为Object对象的原型没有原型，此时若还没有找到那就返回undefined**

## 三、toString

1、当我们直接在页面中打印一个对象时，实际上输出的对象的toString（）方法的返回值

2、输出对象时不输出【object Object】，可以为对象添加一个toString（）方法

## 四、垃圾回收（GC）

前言：程序运行过程中也会产生垃圾，会导致程序运行的速度过慢，所以需要垃圾回收机制来处理垃圾

垃圾概念：

当一个对象没有任何的变量或属性对它进行引用，此时我们将永远无法操作该对象，占用过多堆内存所以必须清理；

--JS自动垃圾回收机制，会自动将这些垃圾对象从内存中销毁，我们不需要也不能进行垃圾回收的操作

--我们需要做的**只是要将不再使用的对象设置为null**，这样就像我们扔垃圾到垃圾箱里，清洁人员便会自动回收；

## 五、数组

前言：内建对象（ES标准）

数组（Array）

--数组也是一个对象

--普通对象　　属性名　　属性值

--数组　　索引　　值

数组的存储性能比普通对象要好，开发中常用

创建数组对象{

第一种：new Array（）；这种只传一个数的时候表示创建该长度的数组

typeof 数组，会返回Object

①添加元素：数组【索引】 = 值

②读取元素：数组【索引】　　（读取不存在的索引不报错而是返回undefined）

③数组中可以使任意的类型

第二种：var arr = 【】；（所谓的字面量）

这种只传一个数的时候表示创建只含有一个该数的数组

}

属性：①length：设置或返回数组长度

注意，修改length小于原数组长度，会删除多余的元素

还可以向数组的最后一位添加元素arr【arr.length】

方法：

1、push：向数组的末尾添加一个或更多元素，并返回新的长度

--原数组**改变**

2、pop：删除数组的最后一位元素，返回删除的元素

--原数组**改变**

3、unshift：向数组的开头添加一个或更多元素，并返回新的长度

--原数组**改变**，而且其他元素的索引也因此改变

4、shift：删除数组第一个元素，返回删除元素

--原数组**改变**

5、slice：从某个已有的数组中返回选定的元素

--参数1为开始索引，若为一个负值则从后往前取

--参数2为结束索引（不包括），省略表示到末尾

--原数组不变！！！！

6、splice：①删除数组中的指定元素，返回删除的元素

--原数组**改变**

--参数1：表示开始位置的索引

--参数2：表示删除的数量

②替换、添加新元素

--参数3及以后：可以传递新的一些元素，会插入到开始索引位置

7、conca：连接两个或多个数组。返回新的数组

--原数组不变！！！！，生成新数组

--参数可以传数组，可以传元素

8、join: 将数组转换为一个字符串，返回生成的字符串

--参数表示指定一个字符串称为数组元素间的连接符，默认逗号连接，传空串则无连接符

--原数组不变！！！！！

9、reverse： 反转数组

--原数组**改变**

10、sort： 对数组元素进行排序，按照unicode编码进行排序；对纯数字排序可能会有错误

--原数组**改变**

**制定排序规则：**

**回调函数中需要定义两个形参**

**浏览器会分别使用数组中的元素作为实参取调用回调函数；**

**使用哪个元素调用不确定，但是肯定参数a一定在参数b前面；**

**浏览器会根据回调函数的返回值来决定元素的顺序，**

**如果返回一个大于0的值，则元素会交换位置**

**如果返回一个小于0的值，则元素位置不变**

**如果返回一个0免责则认为两个元素相等，也不交换位置**

遍历数组：

for循环

forEach（）--只支持ie8以上浏览器

--需要一个函数作为参数{

像这种函数，由我们穿件但是不由我们调用的，称为回调函数

数组中有几个元素函数就会执行几次，每次执行浏览器会将遍历到的元素以实参的形式传递进来

--浏览器会在回调函数中传递三个参数{

第一个参数就是当前正在遍历的元素value

第二个参数就是当前正在遍历元素的索引index

第三个参数就是正在遍历的数组obj

}

}

# 六、函数

方法：

1、call和aplly

--这两个方法都是函数对象的方法，需通过函数对象来调用

--当对函数调用call和apply都会调用函数执行

--在调用call和apply可以将一个对象指定为第一个参数，**此时这个对象将会成为函数执行时的this；**

--call方法可以将实参在对象之后依次传递

--apply方法需要将实参封装到一个数组中统一传递

**2、arguments:**

在调用函数时，浏览器每次都会传递进两个隐含的参数：

--函数的上下文对象this

--封装实参的对象arguments（一个类数组对象，也可以通过索引操作）

--在调用函数时，我们所传递的实参都会在argument中保存

--即使不定义形参，也可以通过arguments来使用实参

--它的属性callee，对应一个函数对象，就是当前正在指向的函数的对象

## 3、Date对象

--如果直接使用构造函数创建一个Date对象，则会封装为当前代码执行时间

--创建一个指定的时间对象

--需要在构造函数中传递一个表示时间的字符串作为参数

--月/ 日/年 时:分:秒

方法：

①getDate： 获取当前日期对象是几日

②getDay： 获取当前日期对象是周几，返回0-6的值，0表示周日

③getMonth：获取当前日期对象的月份，返回0-11的值，0表示1月

④getFullYear：获取当前日期对象的年份

⑤小时，分钟，秒。。。

**⑥getTime：获取当前日期对象的时间戳，指从1970年1月1日0时0分0秒到当前时间的毫秒数；计算机底层保存时间时使用时间戳**

获取当前的时间戳：Date.now（）；

## 4、Math

--Math和其他对象不同，他不是一个构造函数

--属于工具类不用创建对象，它里面封装了数学运算相关的属性和方法

方法：

①abs： 返回数的绝对值

②ceil： 对数进行向上取整

③floor： 对数进行向下取整

④round：对数进行四舍五入

**⑤random：用来生成一个0-1之间的随机数**

**--生成0-x之间的随机数：Math。round（Math。random（）*x）；**

**--生成x-y之间的随机数：Math.round(Math.random()*(y-x)+x);**

⑥max： 获取多个数中的最大值

⑦min：最小值

⑧pow（x，y）：返回x的y次幂

⑨sqrt： 开方

## 5、包装类

前言：在JS中为我们提供了三个包装类，通过这三个包装类可以将基本的数据类型的数据装换为对象。

--String（）

--可以将基本数据类型字符串转换为String对象

--Number（）

--可以将基本数据类型的数字转换为Number对象

--Boolean（）

--可以将基本的数据类型的布尔值转换为Boolean对象

**但是注意：我们在实际应用中不会使用基本数据类型的对象，false的布尔对象在转换回布尔值时就成了true**

**当我们对一些基本数据类型的值取调用属性和方法时，浏览器会临时使用包装类将其转换为对象，然后在调用对象的属性和方法，调用完了以后，再将其转换为基本数据类型**

### 5.1 String方法

**this总结：**

**--1.以函数形式调用时，this永远都是window**

**--2.以方法的形式调用时，this是调用方法的对象**

**--3.以构造函数的形式调用时，this是新创建的那个对象**

**--4.使用call和apply调用时，this是指定的那个对象**

# 复习：

# 子元素选择器： >

伪类选择器：

未访问的链接 a:link

访问过的链接 a:visited

鼠标经过的链接　　a:hover

选择鼠标正在按下还没有弹起鼠标的链接　a:active

顺序：LVHA的顺序声明；链接要单独指定样式

## 文本修饰：

text-aign 文本对齐

text-decoration 装饰文本

none：默认 underline:下划线 overline line-through

text-index段落首行缩进　text_index: 20px　text_index: 2em

line-height行间距

列表：

有序列表：ol...li

无序列表：ul...li

自定义列表：dl...dt/dd

元素分类：

html元素一般分为块元素和行内元素

块元素：div h1-h6 p ul ol

自己独占一行\行高、外边距、内边距都可以控制\是一个容器及盒子\宽度默认是容器（父级宽度）的100%

注意：

文字类的元素内不能使用块级元素

<p>里面不能放块级元素，特别是不能放div

行内元素：a strong b em i del s ins u span

一行内可以显示多个\高和宽直接设置是无效的\默认宽度就是他本身内容的宽度\行内元素只能容纳文本和其他行内元素

注意：

链接里不能再放链接

<a>可以放块级元素

行内块元素: img input td

一行可以显示多个\默认宽度就是他本身内容的宽度\高度、行高、外边距、内边距都可以控制

元素模式转换：

转换为块级元素：display:block

转换为行内元素: display:inline

转换为行内块元素：display:inline-block