

# 12-7

## 面向对象

面向对象编程，即OOP，是一种编程范式，满足面向对象编程的语言，一般会提供类、封装、继承等语法和概念来辅助我们进行面向对象编程。参考：三大特征分别为：封装，继承和多态

### 1：封装

我们平时所用的方法和类都是一种封装，当我们在项目开发中，遇到一段功能的代码在好多地方重复使用的时候，我们可以把他单独封装成一个功能的方法，这样在我们需要使用的地方直接调用就可以了。

### 2：继承

继承在我们的项目开发中主要使用为子类继承父类，下面是es6继承的书写方法

### 3: 多态

多态的具体表现为方法重载和方法重写：

方法重载：重载是指不同的函数使用相同的函数名，但是函数的参数个数或类型不同。调用的时候根据函数的参数来区别不同的函数

方法重写：重写（也叫覆盖）是指在派生类中重新对基类中的虚函数（注意是虚函数）重新实现。即函数名和参数都一样，只是函数的实现体不一样

三大特征的优点：

封装：封装的优势在于定义只可以在类内部进行对属性的操作，外部无法对这些属性指手画脚，要想修改，也只能通过你定义的封装方法；

继承：继承减少了代码的冗余，省略了很多重复代码，开发者可以从父类底层定义所有子类必须有的属性和方法，以达到耦合的目的；

多态：多态实现了方法的个性化，不同的子类根据具体状况可以实现不同的方法，光有父类定义的方法不够灵活，遇见特殊状况就捉襟见肘了

用我个人最简单的理解表达就是，Object的操作。另外一种理解：给你一个条件，你去找个对象帮我处理这个事情，你就不要自己动手了。

说白了，就是在操作对象，需要去创建这个对象，操作的所有的对象都是Object的实例，甚至还可以从原型去生成实例对象。

[http://www.ruanyifeng.com/blog/2010/05/object-oriented\\_javascript\\_encapsulation.html](http://www.ruanyifeng.com/blog/2010/05/object-oriented_javascript_encapsulation.html)

### 3) 继承：

对象继承分两种情况，一种是构造函数的继承，一种是原型（prototype）的继承：

1. 构造函数的继承，比较简单，只需要在子对象中添加代码：`parent.apply(this, arguments);`

关于原型的继承最优化的方法，利用空对象作为中介

## 2. 拷贝继承

可参考：

<https://segmentfault.com/a/1190000002440502>

<http://blog.csdn.net/james521314/article/details/8645815>

js继承有5种实现方式：

- 1、继承第一种方式：对象冒充
- 2、继承第二种方式：call()方法方式
- 3、继承的第三种方式：apply()方法方式
- 4、继承的第四种方式：原型链方式，即子类通过prototype将所有在父类中通过prototype追加的属性和方法都追加到Child，从而实现了继承
- 5、继承的第五种方式：混合方式 混合了call方式、原型链方式