* ECMAScript有两种开发模式：
* 面向过程--函数式(过程化).
* 面向对象(OOP)。面向对象的语言有一个标志，那就是类（构造函数）的概念，而通过类可以创建任意多个具有相同属性和方法的实例对象。
* 面向对象的特点（了解）
* 抽象：就是忽略一个主题中与当前目标无关的那些方面，以便更充分地注意与当前目标有关的方面。
* 封装：就是把我们抽象出的属性和对属性的操作写到类的定义中，称为封装.只能通过对象来访问。
* 继承：从已有的对象上，继承出新的对象。
* 多态：多态性是指允许不同类的对象对同一消息作出响应。
* 对象的组成
* 属性(对象下)--变量(自由的)：状态、静态的。
  + 方法(对象下)--函数(自由的)：过程、动态的。
* 使用Json创建对象（相当于字面量创建）var obj={};
* 使用new Object()（构造函数创建对象）
* 工厂模式

工厂模式是软件工程领域一种广为人知的设计模式，这种模式抽象了创建具体对象的过程。

工厂模式的实现方法非常简单，通过封装解决了创建多个相似对象的问题，但是却无从 识别对象的类型，因为全部都是Object，不像Date、Array等

* 构造函数的使用

创建自定义的构造函数意味着将来可以将他的实例标识为一种特定的类型，这就是构造 函数比工厂模式强的地方，因为它知道自己从哪里来，通过谁产生的。

* 构造函数：其实就是一个普通函数，但是内部使用了this变量。对构造函数使用new运算符，就能生成实例对象，并且this变量会绑定在实例对象上。
* 构造函数的基本特点：首字母大写 new运算符创建。
* 构造函数模式虽然好用，但也并非没有缺点。使用构造函数的主要问题，就是每个方法都要在每个实例对象上重新创建一遍。
* new的使用

以 new 操作符调用构造函数的时候，函数内部发生以下变化：  
   1、创建一个空对象，并且 this 变量引用该对象，同时还继承了该函数的原型。  
   2、属性和方法被加入到 this 引用的对象中。  
   3、新创建的对象由 this 所引用，并且最后隐式的返回 this 。

* 构造函数（类）和对象的关系（实例对象）
* 面相对象和面相过程的区别（概念）
* 应用

1. 掌握面相对象编程的好处
2. 轮播图面相对象改造

* 类的概念

在面向对象编程中，类（class）是对象（object）的模板，定义了同一组对象（又称"实例"） 共有的属性和方法。

* 类和对象
* 类就是一类事物，而对象就是一个事物，一类事物中包含若干的事物（而这其中的具体的事物就是对象）。
  + 类是对象的抽象，而对象是类的具体实例。类是抽象的，不占用内存，而对象是具体的，占用存储空间。类是用于创建对象的，它是一个定义包括在特定类型的对象中的方法和变量的软件模板。
* JSON字符串和对象直接的转换（JSON.parse）
* 应用

1. 萤火虫案例

* 综合应用

1. 烟花效果案例