

Chapter 7.1 OOP general theory

<https://www.cnblogs.com/TomXu/archive/2012/02/03/2330295.html>

<https://zhuanlan.zhihu.com/p/148720234>

我们将考虑**对象创建算法**，了解**对象之间的关系**（包括基础的关系 — 继承）是如何建立的

概论、范式与思想

ECMAScript是基于**原型**实现的**面向对象编程语言**。

基于原型的OOP模型与基于静态类的方式有许多不同。让我们一起来看看他们直接详细的差异。

基于类模型与基于原型模型的特征

基于静态类

在基于类的模型中，有个关于类和实例的概念。

类与对象

类代表了一个**实例**（也就是**对象**）的**抽象**。

实例的特点是：**属性**（对象描述）和**方法**（对象活动）。

对象**存储了状态**（即在一个类中描述的所有属性的具体值），类为他们的实例定义了**严格不变的结构**（属性）和**严格不变的行为**（方法）。

层次继承

基于原型

“原型是一个**对象**，不仅可被用于其他对象的原型副本，或者是作为辅助对象，如果其他对象没有必要的特征，其他对象可以将其委托给这些对象。”