

面向对象

课堂主题

面向对象(一) - 面向对象基础

课堂目标

1. 明确面向对象编程思想
2. 了解面向对象和面向过程的对比优势
3. 掌握 **new** 构造函数
4. 掌握 ES6 **class** 的基本使用
5. 掌握原型和原型链之间的关系

正课内容

什么是对象

对象就是一个封装了数据(属性)和方法的集合体。

什么是面向对象程序设计(Object Oriented Programming)

面向对象，英文名字叫 **Object Oriented**，是一种编程方法。是和面向过程相对应的。

面向对象和面向过程

我们以吃饭这件事来看看什么是面向对象，什么是面向过程。

对于吃饭这件事来说我们通常有两个解决方案：点外卖和自己做。这俩方案中，点外卖就是面向对象，而自己做的就是面向过程。具体对比如下：

在面向过程的吃饭中，我们想要填饱肚子，需要自己亲自下厨把这顿饭做出来，那么，我们就需要先想好吃什么，然后去买菜、洗菜、洗米、蒸饭、炒菜等等一系列的事情（方法）。在做饭的过程中，我们还会面临各种各样的问题，比如，媳妇忽然想换个口味，不想吃今天做的菜了，或者我做的是个咸的西红柿炒蛋，但是媳妇今天要吃甜的，怎么办我们就需要重新做。

那如果是面向对象会怎么样呢？我们今天想要吃什么了，直接打开外卖软件找到自己想要吃的东西，下单就可以了。

在这个场景中，外卖软件就是「对象」。在面向对象编程中，抛弃了函数，想要实现一个功能不再是通过函数的叠加调用实现的了。而是通过对象。

使用 **dragEl** 来实现元素拖拽

如何创建一个对象

工厂模式

最常用的设计模式之一。这种类型的设计模式属于创建型模式，它提供了一种创建对象的最佳方式。

new 运算符

- new 运算符的使用注意事项
 - 参数问题
 - this 问题
- 构造函数
- 对象的实例化

面向对象特征

面向对象具有抽象，封装，继承，多态四大基本特性。（也有说法三大特征封装，继承，多态，这里我们不做辩论）

class 的使用

在 ES6 中，为了给我们区分普通函数和构造函数（类），专门新增了一个糖衣语法 class，为了让我们专门来写面向对象的内容，这个也贴近 Java 等传统语言。接下来我们一边讲解面向对象，一边来看 class 的用法。

抽象性

什么是抽象？简单理解就是抽取功能相似的代码片段，把它归为一类。在这里我们引申出了一个新的概念类，虽然前面我们已经用到了类，但是没有把类专门拿出来讲。在这里我们重新说明一个类和对象。对象：一个封装了方法和数据（属性）的集合体，这个我们前边已经多次使用了，那类是什么呢？

类：专业点说法就是一组具有相同属性和行为的对象的抽象，只看这个貌似不太好理解，怎么理解这个类呢？以我们前边的代码为例，function Drag 这个就是一个类，我们把拖拽所需要的常用功能都封装在这个类中，每次需要需要使用到拖拽这个功能的时候，就实例化这个类，它会给我生成一个 drag 对象，通过该对象我们就实现了拖拽这个功能。总结一下：生成拖拽对象的函数就是我们类。

原型和原型链

每个对象拥有一个原型对象，对象以其原型为模板、从原型继承方法和属性。原型对象也可能拥有原型，并从中继承方法和属性，一层一层、以此类推。这种关系常被称为原型链 (prototype chain)

总结

- 什么是类
- 什么是类的实例化
- 原型和原型链之间的关系

下节课内容

- static 静态方法
- 继承
- 基于观察者模式的自定义事件体系