FP- 学员预习

函数式的概念

> 函数式编程通过使用函数来将值转换成抽象单元，接着用于构建软件系统。

函数式编程和命令式编程的区别

命令式编程，命令式编程往往是建立在直接操作和检查程序状态之上。

函数式编程更倾向于的是，函数式编程的思路是将程序拆分并抽象成多个函数再组装回去。

函数式编程和面向对象编程的区别

面向对象强调的是，一切皆是对象，函数式强调的是，一切皆是函数。

面向对象强调的是主语，函数式强调的是谓词。

函数式的特点

函数是第一等公民

围绕着函数，取代面向过程式的代码，往往能够有以下收益：

- **表达力更加清晰**，因为“一切都是函数”，通过函数的合理命名，函数原子的拆分，我们能够一眼看出来程序在做什么，以及做的过程；
- **利于复用**，因为“一切都是函数”，函数本身具有天然的复用能力；
- **利于维护**，纯函数和幂等性保证同样的输入就有同样的输出，在维护或者调试代码时，能够更加专注，减少因为共享带来的潜在问题。

纯函数

> 一个函数如果输入参数确定，输出结果是唯一确定的，那么它就是纯函数。

多人协作的开发过程中，如何去考虑副作用?

函数的编写

在我们编写一个函数时，比如，一个典型的接口请求，我们怎么写?

组件的封装

在我们进行一个组件的封装时，我们应该考虑哪些问题和维度?

把一个组件，拷贝过来，会有什么问题?

架构的设计

反过来，我们思考一下：模块化、微前端，究竟是解决什么问题？副作用究竟是什么?

熵增。

在理解了副作用的基础上，大家可以往这个角度进行延伸思考。

高阶函数

- 函数可以作为参数传递；
- 函数可以作为返回值；

函数的科里化

> 在计算机科学中，柯里化（currying），又译为卡瑞化或加里化，是把接受多个参数的函数变换成接受一个单一参数（最初函数的第一个参数）的函数，并且返回接受余下的参数而且返回结果的新函数的技术。这个技术由克里斯托弗·斯特雷奇以逻辑学家哈斯凯尔·加里命名的。

函数式的应用与实战

函数的缓存与闭包

函数的编排

应用场景 – 设计模式

装饰器

策略