//1.面向过程的 想到哪写到哪

//2.面向对象的 共有的属性和方法封装到一个类里 封装

//3.面向切面编程 统计一个函数执行的时间

//4.函数式编程 提纯无关于业务的纯函数 函数套函数产生神奇的效果

//5.函数式编程不是用函数来编程 函数套函数让函数更强大 OOP

//6.javascript函数称为一等公民

//7.es3 ->es5->es6

//8.对于函数式编程来讲 同样的输入一定会有同样的输出 永远不依赖于外部的状态

//9.纯函数可以记忆哦（同样的输入总有同样的输出） 不跟外界有任何关系 抽象代码方便单元测试

//10.函数式编程 新建一个独立的js 通过代码的提纯 新建共有的common.js->libs 本项目

//11.函数的柯里化函数 函数接受一堆参数->返回一个新函数 继续接受参数能够处理逻辑

//12.他可以记住参数，相当于对参数的一种缓存

//13.函数组合是为了解决柯里化函数所最后生成的洋葱样的代码

//14.组合函数相当于把一页页的洋葱贴起来

//15.一目了然和系统提供的api保持一致 还有一点不要去生成那些转瞬即逝的代码

//16.声明式的代码越来越受到欢迎 只要提供一条条的指令 程序就知道怎么做而不是一步步的告诉 var i=0

//17.惰性函数就是比较懒得函数 下一次不想再求值了

//18.函数传给函数 让函数具有更复杂的能力和功能

//19.尾部递归的性能要高于传统纯函数递归的性能

//20.函数式变成其实就是函数的种种核心技巧的拼接，但是函数式 编程会充盈着大量的闭包。闭包呢是JS中常见的核心知识

//21.常用的函数式编程的库 RxJS cycleJS lodashJS underscoreJS ramdajs

//22.抽象业务逻辑 系统里有很多可以复用 组合起来有更强大的功能额的时候 抽库 代码健壮 “单元测试！！！！！”

//23.函数式编程的唯一影响因素就是输入和输出

//24.nodejs js多线程 每一个函数式编程里面 根本不涉及到外部的那个被几个线程争执的变量

//25.范畴代表一个容器，容器内部有两种因素组成 一个value 一个是变形关系 变形关系也是函数这个变形关系只能作用于单独的该容器下面的一个元素 但是范畴和范畴之间可以相互转化 用到的也是变形函数 但是这个变形函数非常的特殊，因为它能作用于当前容器的每一个元素。他有一个特殊的名字 函子，随着计算机的发展历程，产生了很多函子。这些函子本身都是数学方法，后来被依次利用到函数式编程中。

//26.本来一个容器不能调用自身的函数 函子可以 容器只留一个接口 map 可以运行容器内的函数

//A ->map map<- B 容器包含value和变形关系，容器有个方法叫map，可以访问容器内很多方法。

//27.一个容器一旦接受了 变形关系 接受一个函子 接受一个函数 就变成了一个新的容器

//28.函子 容器 ->(值 变形关系) 普通的容器 map 函子 它的变形关系可以作用每一个值 将当前容器转换成另一个容器

//29.加一层容错机制Maybe 代表着传统编程中的if Either因子已右侧为

//30.AP因子的函数执行 普通函子的value 也可以实现链式调用

//31.Monad 函子的作用是，总是返回一个单层的函子。实现I/O 是不纯的操作

//32.函数式编程真正的火热是随着React的高阶函数真正火热升温的。

//33.测试一些性能的时候并不能单纯的在浏览器控制台去跑，需要用一些专业的性能测试工具（benchMark）

//34.