函数式编程 by Anto Aravinth

2018年1月第一版

## 序言

快速掌握一个知识体系的秘诀是抓住概念并理清概念之间的关系。

## 常识

JavaScript

1. **万物皆变量-var**：包括函数也是 var 类型，或者说不区分变量类型，运行时靠分析值来判断
2. 与函数式编程对立的是 **命令式编程** 。
3. **缓存机制**：如果函数是恒等式，则多次对同一函数相同参数的调用结果将会被缓存重用。

目录

[序言 1](#_Toc519330711)

[常识 1](#_Toc519330712)

[一、函数式编程简介 1](#_Toc519330713)

[1、函数式编程的概念： 1](#_Toc519330714)

[2、JS中函数与方法的区别 2](#_Toc519330715)

## 一、函数式编程简介

1、函数式编程的概念：

函数式编程是一种范式，我们能够一次创建仅仅依赖输入（参数）就可以完成自身逻辑的函数。

**优势**

**引用透明性**：

* 函数不访问（或者说依赖）外部变量。
* 所有需要的信息都通过参数传入。
* 恒等式特性：保证了函数被多次调用时仍然返回同样的结果。

**透明性的好处**

* 代码**可缓存**
* JavaScript一大特性：函数值缓存，如果函数是恒等式，则多次对同一函数相同参数的调用结果将会被缓存重用。
* 易**测试**
  + - 如果需要依赖全局变量，则难以单独测试。
* 可**并发**执行
* 多线程运行，不需要加锁

2、JS中函数与方法的区别

函数是Root下的函数体

var simple = (a) => {return a}

simple(a)

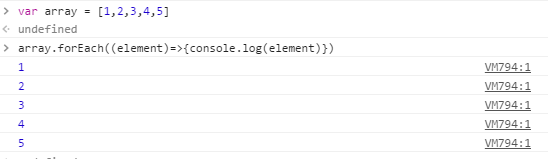
方法是 对象 下的函数体

var obj = { simple : (a) => {return a} }

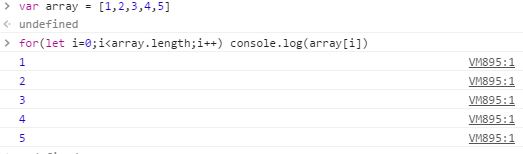
obj.simple(5)

3、抽象机制：用声明式 而 不用命令式

声明式



定义式



最大区别就是一个写具体代码，一个调用函数（这里是数组内置函数forEach）。