https://zhuanlan.zhihu.com/p/91040461

The Recurse Center

https://www.recurse.com/

那里算是我经历过最好的编程社区了

which quick plug is probably the best programming community I've ever experienced

Do everything with functions

input -> output

在函数式开发中一切都要函数化(小甲鱼:重点!)

or obvious is that in functional programming you want to do everything with functions

基本思想是函数输入输出的单一映射关系：函数结果只由输入决定（不受全局变量等的影响）。指定唯一输入后就会返回唯一输出

Not pure function没有参数或没有返回值

Not pure:

```
var name = “Anjana”;
function greet() {
   console.log(“Hi, I’m ” + name);
}
```

没有作为函数的参数

we don't have name as an argument to this function

Pure function举例：

Pure:

```
function greet(name) {
   return “Hi, I’m ” + name;
}
```

这里对函数输出唯一重要的就是输入

here we have the only thing that matters to the output of this function is its input

# Use higher-order functions

## functions can be inputs/outputs

意味着可以将其他函数作为函数的输入

so this means functions that can take as inputs other functions

```
function makeAdjectifier(adjective) {
   return function (string) {
     return adjective + “ ” + string;
   };
}

var coolifier = makeAdjectifier(“cool”);

coolifier(“conference”);
  => “cool conference”
```

得到一个名为coolifier的函数

and get a function that's called coolifier

Don’t iterate

use map, reduce, filter

我们要避免的一件事就是

ok so one of these things that we're going to avoid that we're used to doing is

用map, reduce, filter这样的高阶函数而不是循环

Avoid mutability

use immutable data

变异可以理解为对象的改变

mutation in the sense I just mean changing objects in place

---

函数式编程不希望改变变量

## Mutation (bad!):

```
var rooms = [“H1”, “H2”, “H3”];
rooms[2] = “H4”;
rooms;
=> ["H1", "H2", "H4"]
```

然后rooms就发生变化了
and so then we have rooms has actually changed

## No mutation (good!):

```
var rooms = [“H1”, “H2”, “H3”];
Var newRooms = rooms.map(function (rm) {
  if (rm == “H3”) { return “H4”; }
  else { return rm; }
});
newRooms; => ["H1", "H2", "H4"]
rooms; => ["H1", "H2", "H3"]
```

阿呀,上面应该是===(小甲鱼:更严谨)
oops I have a missing equal sign there last-minute slides

# Persistent data structures for efficient immutability

Mori, Immutable.js

很赞,还有时间谈谈持久数据结构

okay good we have a little time to talk about persistent data structures yeah

看https://www.bilibili.com/video/BV1bE411B74p?t=45518：20

## FP libraries for JS

- Mori (http://swannodette.github.io/mori/)
- Immutable.js (https://facebook.github.io/immutable-js/)
- Underscore (http://underscorejs.org/)
- Lodash (https://lodash.com/)
- Ramda (http://ramdajs.com/)
- ...and more!

例如Underscore和Lodash库

so Underscore Lodash for example

# “An introduction to functional programming”

by Mary Rose Cook

https://codewords.recurse.com/issues/one/an-introduction-to-functional-programming

由MRC写的函数式编程介绍

called an introduction to functional programming by Mary Rose Cook