

برنامه‌نویسی Functional

یه ماشینی هست به اسم ماشین تورینگ، این ماشین یه سری کارهای مهمی رو انجام میده
ماشین تورینگ: به زبان‌های برنامه‌نویسی که می‌تونه کارهای ماشین تورینگ رو انجام بده، تورینگ کاملیت
می‌گن.

پس بنابراین اگه زبانی، تورینگ کاملیت باشه، یعنی تمام کارهای لازم برای یه برنامه‌نویس رو انجام میده

با ریاضی میشه اثبات کرد که یه زبان تورینگ کاملیت هست یا نه.

زبان‌های برنامه‌نویسی فانکشنال تورینگ کاملیت هستن همشون چون باید قابلیت‌های لازم رو داشته باشن.

باید همه کارهایی رو که با کلاس میشه کرد با اون فانکشنال‌ها بشه کرد.

وقتی داخل یه زبان برنامه‌نویسی فانکشنال داریم یه تابع رو صدا می‌زنیم، دیتاها کجا ذخیره میشن؟

مثلا متراژ خونه رو باید داخل فانکشن ذخیره کرد.

کلی تکنیک هست که بعضی‌هاشو میشه تو زبان‌های غیر فانکشنال هم اجرا کرد.

فرض کنیم که نمی‌خوایم از این استفاده کنیم:

```
def sum(a,b):  
    return a+b
```

پس یعنی شبیه این:

```
def sum10(a):  
    return a+10  
  
def sum15(a):  
    return a+15
```

یه تکنیک هست که تو پایتون هم میشه استفاده کرد.

```
def sum_maker(b):  
    def f(a):  
        return a+b  
    return f
```

مثلا برای مثال املاکی اینجوری استفاده می‌شه:

```
def amlak(x,y,z,...):  
    def get_metrage():  
        return x  
    return get_metrage
```

بنابراین در یک برنامه functional خالص و pure مثل purely تنها همین خط اجرا می‌شه:

```
a(b(c(d(f(e))))))
```

این روش برا برنامه‌های concurrent و برنامه‌های حساس خیلی خوبه چرا؟ چون دیتا هیچ وقت تغییر نمی‌کنه! مثلاً اگه بخوایم مترآژ خونه رو عوض کنیم، باید یه کپی از خونه با اون مترآژی که می‌خوایم بگیریم.

یه سوال!

یه کد مشخص رو چند نفر اجرا میکنن و به ترتیب هم نیست ولی دیتاها شیر میشه و یه مقدار ثابتی رو هر متغیر و هر نفر می‌بینه

```
a = 0  
Lock = False  
if Lock == False:  
    a = 1  
    Lock = True  
else:  
    return
```

چه کنیم که وقتی هر دو نفر همزمان این کد رو ران می‌کنن یه نفر تغییر بده یه نفر تغییر نده؟
برنامه‌نویسی موازی!