

ReactiveX for Python

George Adams IV

Father of Two

Full-Stack / DevOps Engineer

Dunning-Kruger

Agenda

- Why Use Rx
- The Pattern
- Hot & Cold
- Running Code Examples

Why Use Rx

- Consistency across languages (.NET, Java, JS, etc.)
- Highly composable
- Simplifies programs with time as a variable

The Pattern

The Pattern

1. Define data source
2. Observe changes
3. Subscribe

Observable

```
Observable.timer(0, 1000)  
Observable.create(lambda o: o.on_next(42))
```

Observer

```
class MyObserver(Observer):
    def on_next(value):
        pass

# on_error, on_completed
```

Observing with Lambdas

```
source.select(  
    lambda x: x * 2  
)
```

Scheduler

Defines where and when to do work.

- asyncio

Subjects

"Never" use them.

Hot & Cold

Hot & Cold

A *cold* Observable is *not* producing values.

A *hot* Observable *is* producing values.

If a tree falls in the woods...

Observables do not produce values until there's a subscription.

Code