

A Primer on Functional Programming

Sarah Withee
@geekygirlsarah

Slides: geekygirlsarah.com/primer-fp



Who has heard of functional programming?

@geekygirlsarah #CodeMash

geekygirlsarah.com/primer-fp

Who has done some form of
functional programming?

@geekygirlsarah #CodeMash

geekygirlsarah.com/primer-fp

Who IS a functional programmer?

@geekygirlsarah #CodeMash

geekygirlsarah.com/primer-fp

Who has wanted to learn but
never had time or good
resources?

@geekygirlsarah #CodeMash

geekygirlsarah.com/primer-fp



Intro

1. Functional Programming Concepts
2. Why Use Functional Programming?
3. Thinking Functionally
4. Brief Glance at Functional Programming Languages

Hello!

I am Sarah Withee

I'm a software engineer at
Arcadia.io

I'm on all the social media as
[@geekygirlsarah](#)

(Yes, you can tweet all the things!)

[@geekygirlsarah](#) [#CodeMash](#)



geekygirlsarah.com/primer-fp



Sarah Withee @ CodeMash

@geekygirlsarah



Question of the Day: What is the last silliest conversation you've been a part of?

10:00 AM - 10 Jan 2019

2 Likes



4



2



@geekygirlsarah #CodeMash

geekygirlsarah.com/primer-fp



1. Functional Programming Concepts



Background

It's not new!

(Languages and ideas have been around
since 1950s)

@geekygirlsarah #CodeMash

geekygirlsarah.com/primer-fp





Background

Built on ideas of lambda calculus
developed in the 1930s

(I promise, we're not discussing this today)

@geekygirlsarah #CodeMash



geekygirlsarah.com/primer-fp

The Key Takeaway....




Pure Functions:

Function that, given a certain input, ***always*** produces the same output.



@geekygirlsarah #CodeMash

geekygirlsarah.com/primer-fp



Functional Programming Concepts

Pure functions don't have side effects



Functional Programming Concepts

Side effects include:

- Time
- File access
- Database access
- Network access
- Previous function calls



Functional Programming Concepts

User input is never pure because
users are unpredictable

(Duh.)



Functional Programming Concepts

Call by reference is never pure



Functional Programming Concepts

So...

It's basically impossible to write
software using 100% pure functions



FP Concepts > Examples

`sin(x)`

`abs(x)`

`sqrt(x)`

Always returns same values for x

@geekygirlsarah #CodeMash

geekygirlsarah.com/primer-fp



All math functions do this



FP Concepts > Examples

```
str.length()  
str.isEmpty()  
str.concat(str2)
```

Always returns same values for the
string str (and str2)

@geekygirlsarah #CodeMash

geekygirlsarah.com/primer-fp



Generally functions based on simple data types will be like this



FP Concepts > Examples

```
getAccountNumberFromDb(acctOwner)
```

Kidding... definitely not pure.

@geekygirlsarah #CodeMash

geekygirlsarah.com/primer-fp



Why?

- What if someone changes their name?
- What if they close an account?
- What if the number changes (like debit card stolen)?
- What if database goes down?



FP Concepts > Review

- ▶ Pure functions, given certain input, always produce same output
- ▶ No side effects
- ▶ User input is never pure
- ▶ Call by reference is never pure
- ▶ Impossible to write software in 100% pure functions

Referential Transparency

Any expression that can replace a function with its return value with no behavior changes



@geekygirlsarah #CodeMash

geekygirlsarah.com/primer-fp



Functional Programming Concepts

Mathematical example:

If $x = 3$...

$$x + 5 = 8$$

$$3 + 5 = 8$$



Functional Programming Concepts

`x = 144`

`y = sqrt(144) + 6`

`y = 12 + 6`

`# y = 18`

`# same`



Functional Programming Concepts

Pure functions *always* have
referential transparency



Functional Programming Concepts

In mathematics, all functions are transparent

In programming, this is NOT true



Functional Programming Concepts

Assignments are NOT transparent
("referentially opaque")

$$x = x + 1$$



Functional Programming Concepts

```
def addOne(int num):  
    return num + 1;
```

If x and y are the same, then
 addOne(x) and addOne(y)
give same result



Functional Programming Concepts

More languages are starting to have immutable variables by default



Functional Programming Concepts

Lambda function (or anonymous function):

A function without a name



Functional Programming Concepts

Why lambdas?

For higher level functions or to pass arguments to a function

Usually used once to few times



Functional Programming Concepts

Lambdas can't be recursive*

* otherwise they need a name or some way of maintaining state**

** which is possible but outside of this scope

@geekygirlsarah #CodeMash

geekygirlsarah.com/primer-fp



<https://stackoverflow.com/questions/2067988/recursive-lambda-functions-in-c11>

C++11:

```
std::function<int(int,int)> sum;  
sum = [term,next,&sum](int a, int b)->int {  
    if(a>b)  
        return 0;  
    else  
        return term(a) + sum(next(a),b);  
};
```

C++14:

```
void f() {  
    static int (*self)(int) = [](int i)->int { return i>0 ? self(i-1)*i : 1; };  
    std::cout<<self(10);  
}
```

Functional Programming Concepts


```
f = lambda x: x*x  
print f(5)
```

variable

lambda argument

Function definition

x




Functional Programming Concepts

Functions ARE values

Can be passed as values into other functions

@geekygirlsarah #CodeMash

geekygirlsarah.com/primer-fp



- A lot of the core of functional programming
- Because of referential transparency and pure functions, this can be like passing in a regular value

```
def divide(x, y):  
    return x/y
```

```
def divisor(d):  
    return lambda r: divide (r, d)
```

```
half = divisor(2)  
print(half(32))
```

lambda r: divide(r, 2)

divide(32, 2)




Functional Programming Concepts

Who has heard of map/filter/reduce?

@geekygirlsarah #CodeMash



geekygirlsarah.com/primer-fp



Functional Programming Concepts

Who has *used* map/filter/reduce?

@geekygirlsarah #CodeMash



geekygirlsarah.com/primer-fp



Functional Programming Concepts

Map – apply a function to all terms in a list

```
map(function, list)
```

@geekygirlsarah #CodeMash

geekygirlsarah.com/primer-fp



- Depending on language, the order of arguments may swap

```
items = [1, 2, 3, 4, 5]
```

```
squared = map(lambda x: x**2, items)
```

```
# same as
```

```
squared = []
```

```
for i in items:
```

```
    squared.append(i**2)
```




Functional Programming Concepts

Filter – creates a list for all items that match a filter (function returns true)

`filter(function, list)`

```
items = [1, 2, 3, 4, 5, 6, 7, 8]

under_5 = filter(lambda x: x<5, items)
                # same as

under_5 = []
for i in items:
    if (i < 5):
        under_5.append(i)
```



Functional Programming Concepts

Reduce – returns result of some computation on a list

```
reduce(function, list)
```

```
items = [1, 2, 3, 4, 5, 6, 7, 8]
```

```
product = reduce(lambda x, y: x * y, items)
```

```
# same as
```

```
product = 1
```

```
for i in items:
```

```
    product = product * num
```



Functional Programming Concepts

Note: Reduce works differently in different languages

`reduce_left()`

`reduce_right()`



Steven Luscher
@steveluscher

Follow

Map/filter/reduce in a tweet:

```
map([🌽, 🐮, 🍷], cook)
=> [🍿, 🍔, 🍳]
```

```
filter([🍿, 🍔, 🍳], isVegetarian)
=> [🍿, 🍳]
```

```
reduce([🍿, 🍳], eat)
=> 🤮
```

RETWEETS
8,844

LIKES
9,296



7:08 PM - 9 Jun 2016

<https://twitter.com/steveluscher/status/741089564329054208>

@geekygirlsarah #CodeMash

geekygirlsarah.com/primer-fp



2. Why Use Functional Programming?

- A lot of things thrown at you
- The beginning might seem promising
- The later stuff might just be weird
- So...why?



Why Use Functional Programming?

Pure functions are simpler and faster to write



Why Use Functional Programming?

Pure functions that work correctly
will ***always*** work correctly



Why Use Functional Programming?

Stack traces are a pain in OOP

Stack traces in FP simplify things



Why Use Functional Programming?

Unit testing IS* functional programming. No side effects make unit tests pass reliably.

* Well, should be anyway

@geekygirlsarah #CodeMash



geekygirlsarah.com/primer-fp

- Set up environment
- Run unit test
- Assert the result
- Clear out what you set up
- Should affect nothing else



Why Use Functional Programming?

Global state of program isn't affected
by pure functions



Why Use Functional Programming?

Concurrency is WAY easier

@geekygirlsarah #CodeMash

geekygirlsarah.com/primer-fp





Why Use Functional Programming?

As code grows larger, it's all more reliable

*Better small modules
-> better large modules*

Activity 1

In a moment, everyone will stand up.

1. The count starts at 0
2. Each person will take the previous number, add 1 to it
3. Say the number out loud
4. Sit down
5. Repeat 2-4 until everyone is sitting
6. Last person reports the total

Activity 2

In a moment, everyone will stand up.

1. Count yourself at first (1)
2. Find a neighbor
3. Total your two numbers together
4. One of you sits down
5. Repeat steps 2-4 for each person in the row
6. Someone will return the final result



Why Use Functional Programming?

Activity 1 was like a for/while loop

- ✦ $x = x + 1$
- ✦ Took a long time



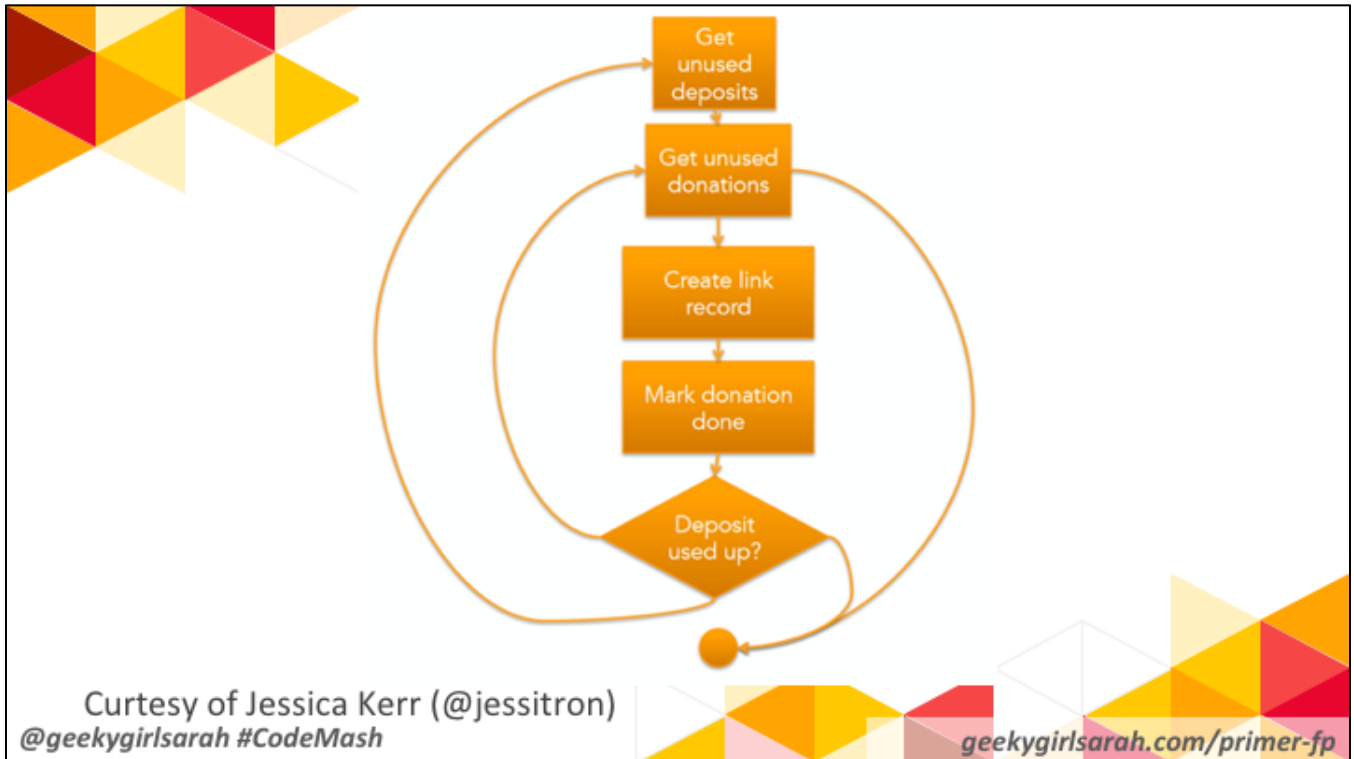
Why Use Functional Programming?

Activity 2 was (hopefully) recursive and concurrent

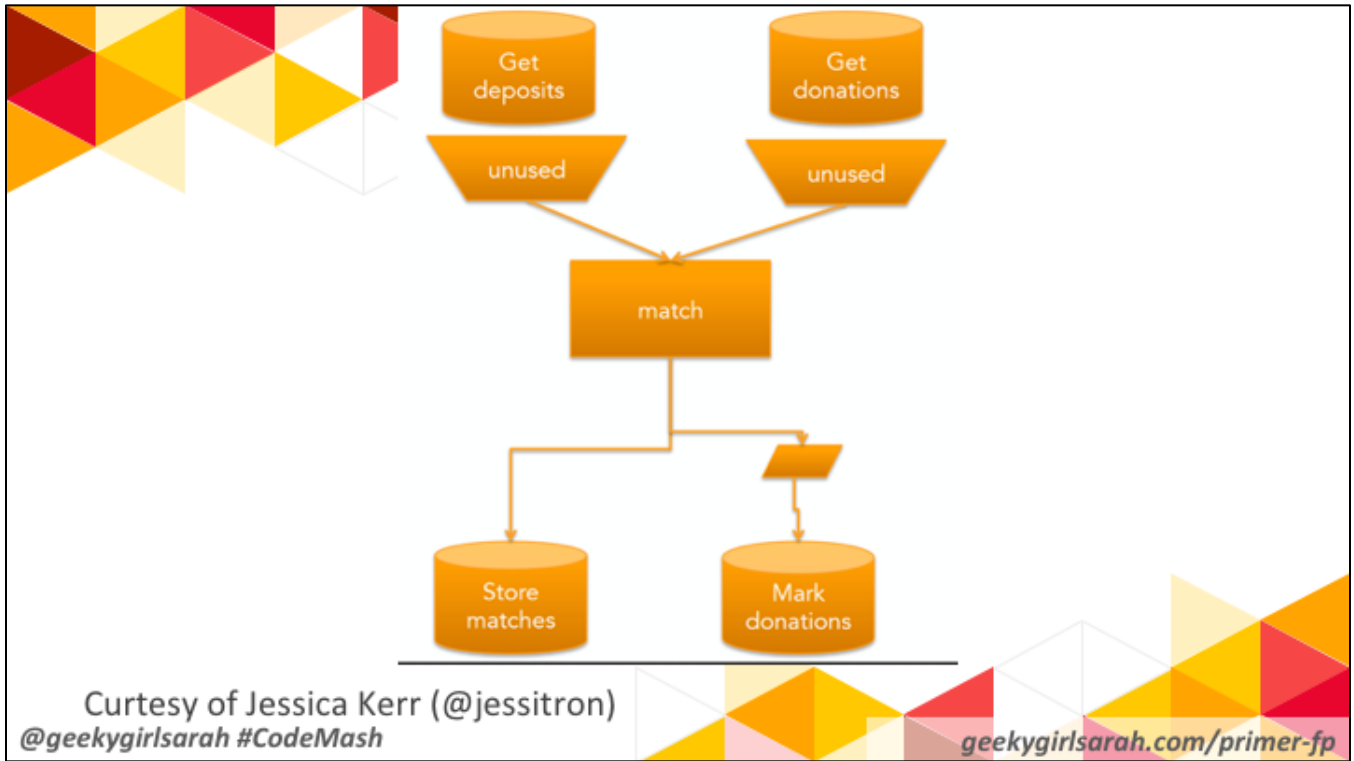
- ▶ Many were counting (no side effects)
- ▶ “Reduce” all the totals together

The background of the slide is a solid red color. It is decorated with several clusters of triangles in shades of yellow, orange, and red. Some triangles are solid, while others are outlined in white. These clusters are located in the top-left, top-right, and bottom-left corners, as well as a smaller one in the bottom-right.

3. Thinking Functionally



- A more imperative/OOP way of doing non-profit donations



- A more imperative/OOP way of doing non-profit donations



4.

Brief Glance at Functional Programming Languages

Functional Languages (Pure)

- | | | |
|-------------|-----------|--------------|
| ◀ Agda | ◀ Frege | ◀ Mercury |
| ◀ Charity | ◀ Futhark | ◀ Miranda |
| ◀ Clean | ◀ Haskell | ◀ Purescript |
| ◀ Coq | ◀ Hope | ◀ SAC |
| ◀ Cuneiform | ◀ Idris | ◀ SASL |
| ◀ Curry | ◀ Joy | ◀ SequenceL |
| ◀ Disciple | ◀ KRC | |
| ◀ Elm | ◀ Mars | |

@geekygirlsarah #CodeMash

geekygirlsarah.com/primer-fp

- 22 languages (at least)
- They don't allow side effects and guarantee referential transparency
- KRC = Kent Recursive Calculator
- SAC = Single Assignment C
- SASL = St. Andrews Static Language

Functional Languages (Impure)

- | | | | |
|---------------------|--------------------------------|---------------|---------------------|
| ▶ ActionScript | ▶ Dylan | ▶ JScript | ▶ Python |
| ▶ Alice | ▶ ECMAScript | ▶ Julia | ▶ Q (both of them) |
| ▶ APL | ▶ Emacs Lisp | ▶ Kotlin | ▶ R |
| ▶ ATS | ▶ Erlang | ▶ LFE | ▶ Red |
| ▶ CAL | ▶ Elixir | ▶ Little b | ▶ Ruby |
| ▶ C++ (since C++11) | ▶ F# | ▶ Logo | ▶ REFAL |
| ▶ C# | ▶ Groovy | ▶ Mathematica | ▶ Rust |
| ▶ Ceylon | ▶ Hop | ▶ Nemerle | ▶ Scala |
| ▶ Clojure | ▶ J | ▶ Nim | ▶ Scheme/Racket |
| ▶ Common Lisp | ▶ Java (since JDK8) | ▶ OCaml | ▶ Spreadsheets |
| ▶ Curl | ▶ JavaScript | ▶ Opal | ▶ Standard ML (SML) |
| ▶ D | ▶ JMP Scripting Language (JSL) | ▶ OPS5 | ▶ Tea |
| ▶ Dart | | ▶ PHP | ▶ Wolfram Language |

@geekygirlsarah #CodeMash

geekygirlsarah.com/primer-fp

- 50 languages (at least)
- For the most part, they implement lambdas and passing functions around (map/filter/reduce)
- Any surprises in here?

The background of the slide is a solid red color. It is decorated with several clusters of triangles in various shades of red, orange, and yellow. Some triangles are solid, while others are outlined in white. These clusters are located in the top-left, top-right, and bottom-left corners, leaving a large red rectangular area in the center for text.

5. Conclusion



Conclusion

- ▶ Functional programming is getting more popular, but been around for decades
- ▶ Functional principles make your code simpler, smaller, more reliable
- ▶ Functional concepts can work in nearly any language

Thanks!

Any questions?

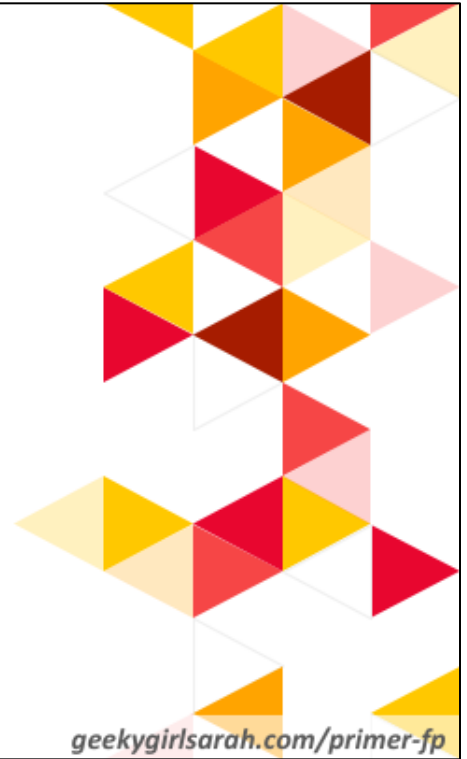
You can find me at

- ✦ @geekygirlsarah
- ✦ hello@sarahwithee.com

Slides: geekygirlsarah.com/primer-fp

@geekygirlsarah #CodeMash

geekygirlsarah.com/primer-fp





Credits

Special thanks to all the people who made and released these awesome resources for free:

- ▶ Presentation template by [SlidesCarnival](#)
- ▶ Example images by Jessica Kerr (@jessitron)



Presentation design

This presentation uses the following typographies and colors:

- Titles: Montserrat bold
- Body copy: Montserrat light

You can download the fonts on this page:

<https://www.fontsquirrel.com/fonts/montserrat>

Yellow **#ffc800** • Orange **#ffa400** • Raspberry **#f64646** • Crimson **#e8062f**

@geekygirlsarah #CodeMash

geekygirlsarah.com/primer-fp

