# Ruby Beyond the Basics

## Introduction

John Cinnamond

@jcinnamond | panagile.com

Ruby is simple in appearance,
but is very complex inside

```
2.times do
  puts "hello".upcase
end
```

```
class Person < ActiveRecord::Base
  has_many :addresses
end
```

Object Orientation

Functional Programming

Metaprogramming

Who this course is for

How the course is structured

What you need to follow along

# Who is this course for?

# People who…

Know some ruby syntax

… and want to learn more

Ever asked “why?” or “how?”

when looking at Ruby

I'm assuming you know some basic Ruby syntax

Know how to write "hello world"

…how to write a class

…how to use if

You don’t need to
know any theory

# Make sure you can understand this…

```
firstname = “John”

lastname = “Cinnamond”


if firstname && lastname

  puts “#{lastname.upcase}, #{firstname}”

end
```

# Course Structure

# Object-Oriented Programming

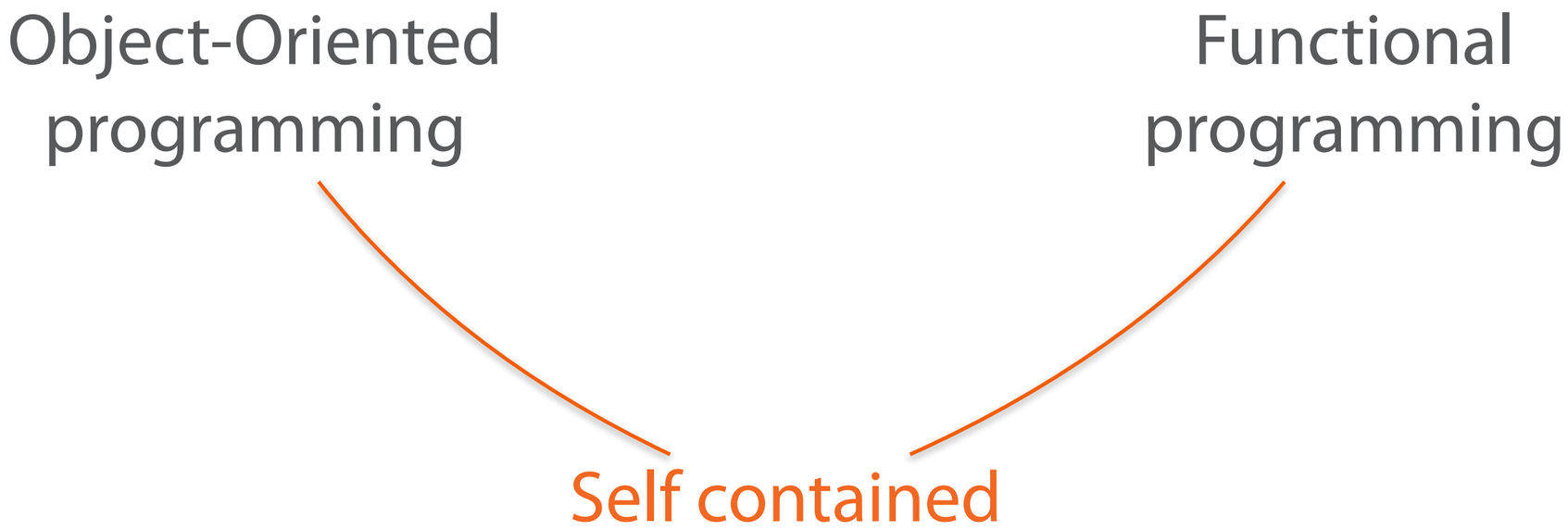
- What it means
- How Ruby supports it
- Everything is an object
- Syntax for classes, methods, and instance variables
- Duck Typing
- Sharing code
- Method dispatch
- Classes and Constants

# Functional Programming

- Procs, blocks, and lambdas
- Pure functional Ruby
- How functional thinking can help

# Metaprogramming

- Create new programming languages (almost)
- Use Ruby to write more Ruby
- Build on ideas from Object-Oriented and Functional programming


Object-Oriented
programming
Functional
programming
Self contained

# Metaprogramming

Needs knowledge about
classes and blocks

Watch the videos in order

# Setup

# Install Ruby >= 2.0

R

E

P

L

Read

E

P

L

Read
Evaluate
P
L

Read
Evaluate
Print
L

Read
Evaluate
Print
Loop

# All you need

- Ruby
- Pry or IRB
- A text editor