

λ 's and POP's

Jan Hettich
LVRUG
5/21/2014

Two kinds of Proc

	“plain old proc”	lambda
closure?	yes	yes
class	Proc	Proc
.lambda?	false	true
semantics	yield	invocation
similar to	block	method

Creation

- “Plain old procs”
 - Kernel.proc {|args| code}
 - Proc.new {|args| code}
- Lambdas
 - Kernel.lambda {|args| code}
 - literal: ->(args) {code}
 - Method.to_proc

call syntax

- direct call
 - `foo.call(args)`
 - `foo[args]`
 - `foo.(args)`
- block-style call
 - `iterator args, &foo`

call semantics

- *yield semantics* (blocks, pops)
 - similar to parallel assignment
 - return => lexically enclosing method
- *invocation semantics* (methods, lambdas)
 - number of arguments must match
 - “diminutive return”

Let's look at some code

- A Digression on Method
- Argument Passing
- *return* statement
- Using *next*
- New behavior in Ruby 2.1.0