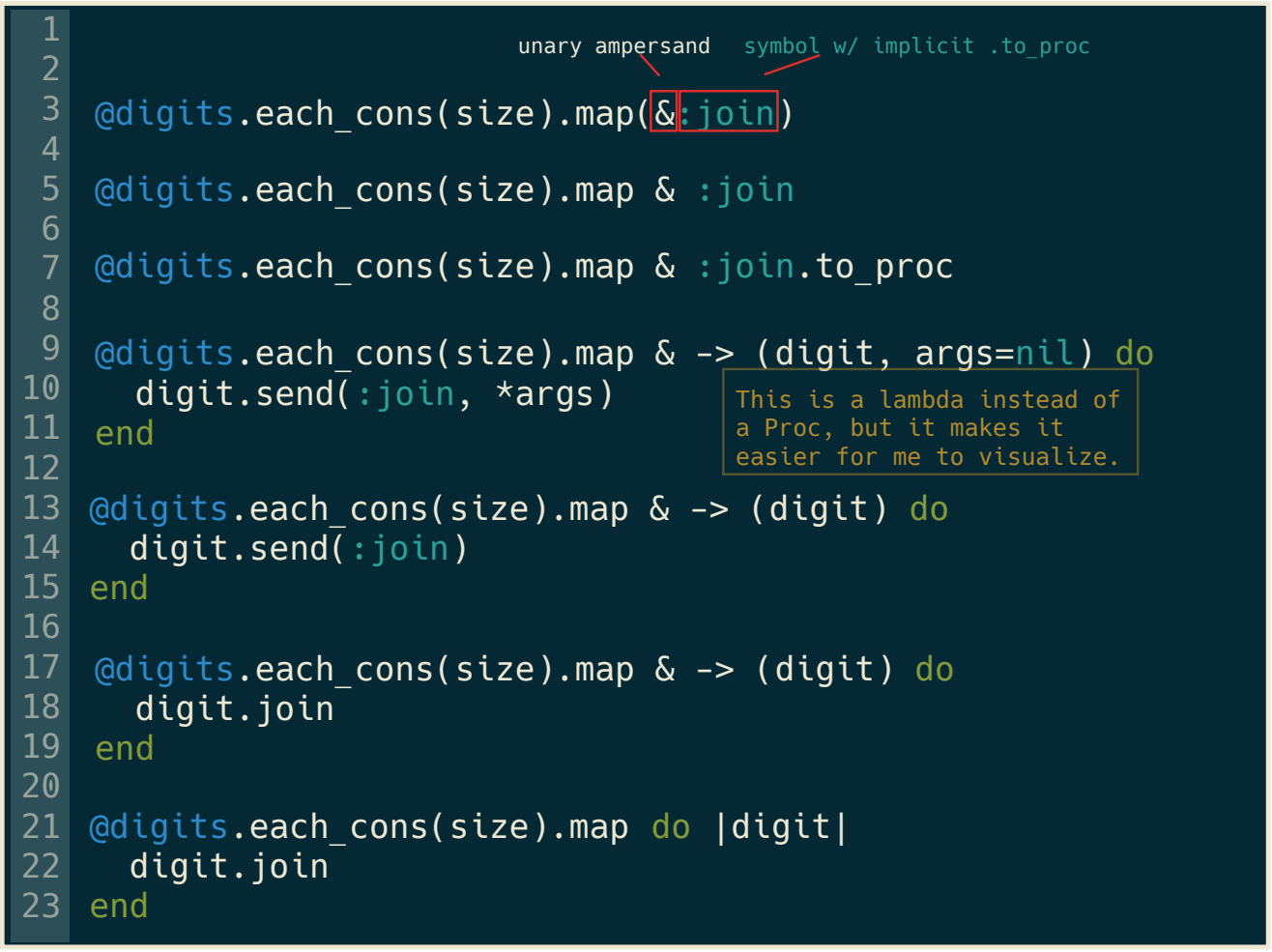
Unpacking the Ruby Pretzel Colon

How the shorthand syntax is the same as the block form, one step at a time.




```
1
2
3  @digits.each_cons(size).map(&:join)
4
5  @digits.each_cons(size).map & :join
6
7  @digits.each_cons(size).map & :join.to_proc
8
9  @digits.each_cons(size).map & -> (digit, args=nil) do
10   digit.send(:join, *args)
11 end
12
13 @digits.each_cons(size).map & -> (digit) do
14   digit.send(:join)
15 end
16
17 @digits.each_cons(size).map & -> (digit) do
18   digit.join
19 end
20
21 @digits.each_cons(size).map do |digit|
22   digit.join
23 end
```

Two separate things are going on in the pretzel colon.

The symbol (with an implicit .to_proc) is converted to a Proc, equivalent to the stabby lambda on lines 17-19.

Then the unary ampersand converts the Proc to a block, which it gives to map, resulting in the equivalent of theblock form on lines 21-23.