

Understanding Iteration



Paolo Perrotta

FREELANCE DEVELOPER

@nusco

for Loops

```
for i in 0..5  
    puts i  
end
```

for Loops

```
for i in 0...(array.length)
  puts array[i]
end
```

Discovering Blocks

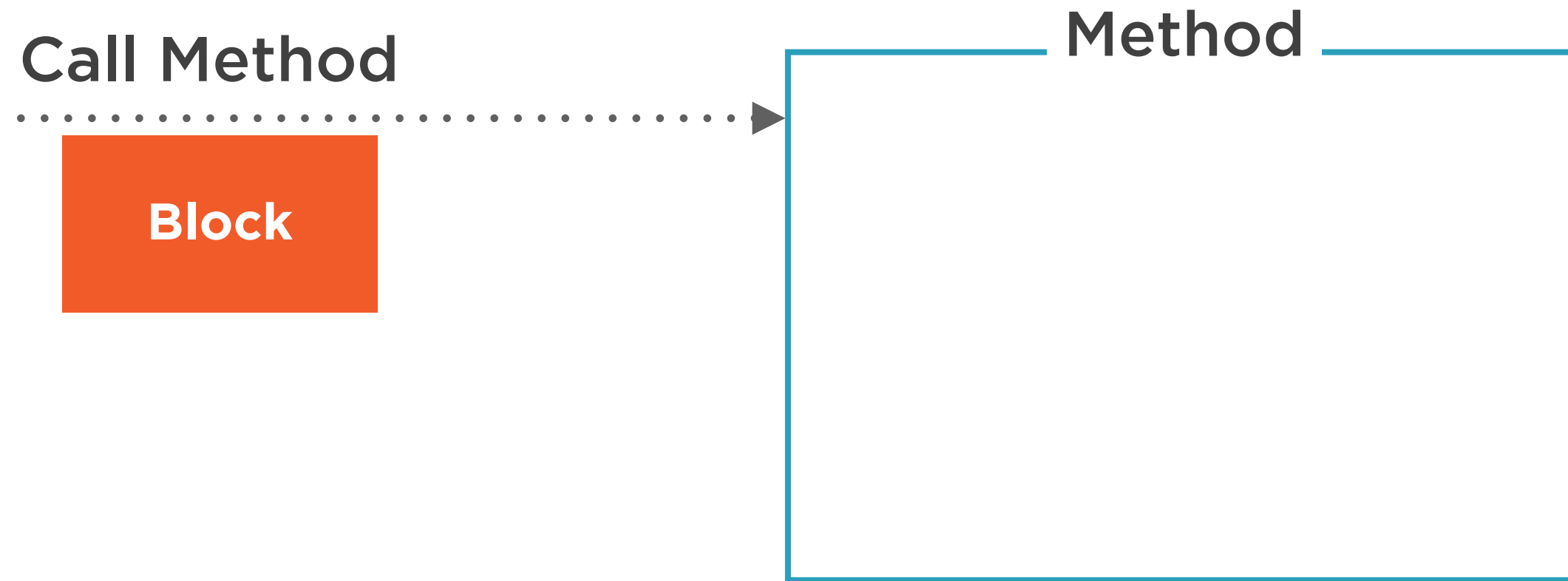
What a Block Looks Like

```
do |arguments|  
  # do something  
end
```

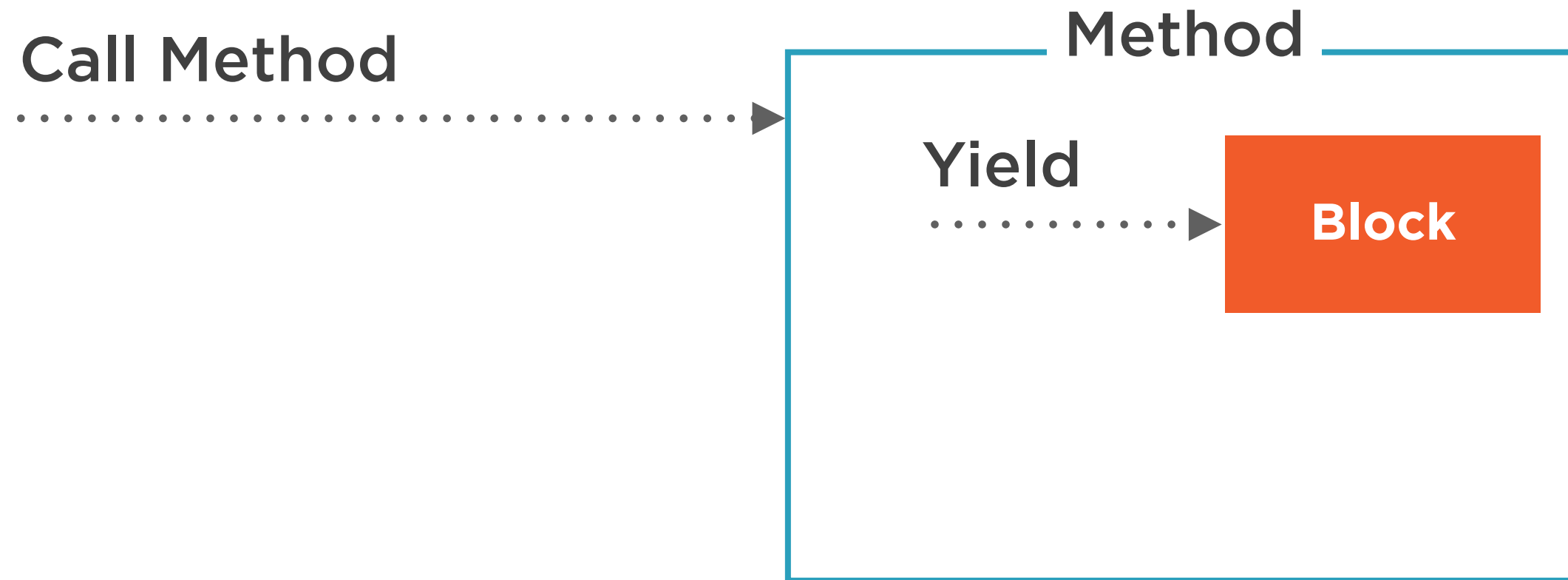
What a Block Looks Like

```
{ |arguments|  
  # do something  
}
```

Blocks as Callbacks

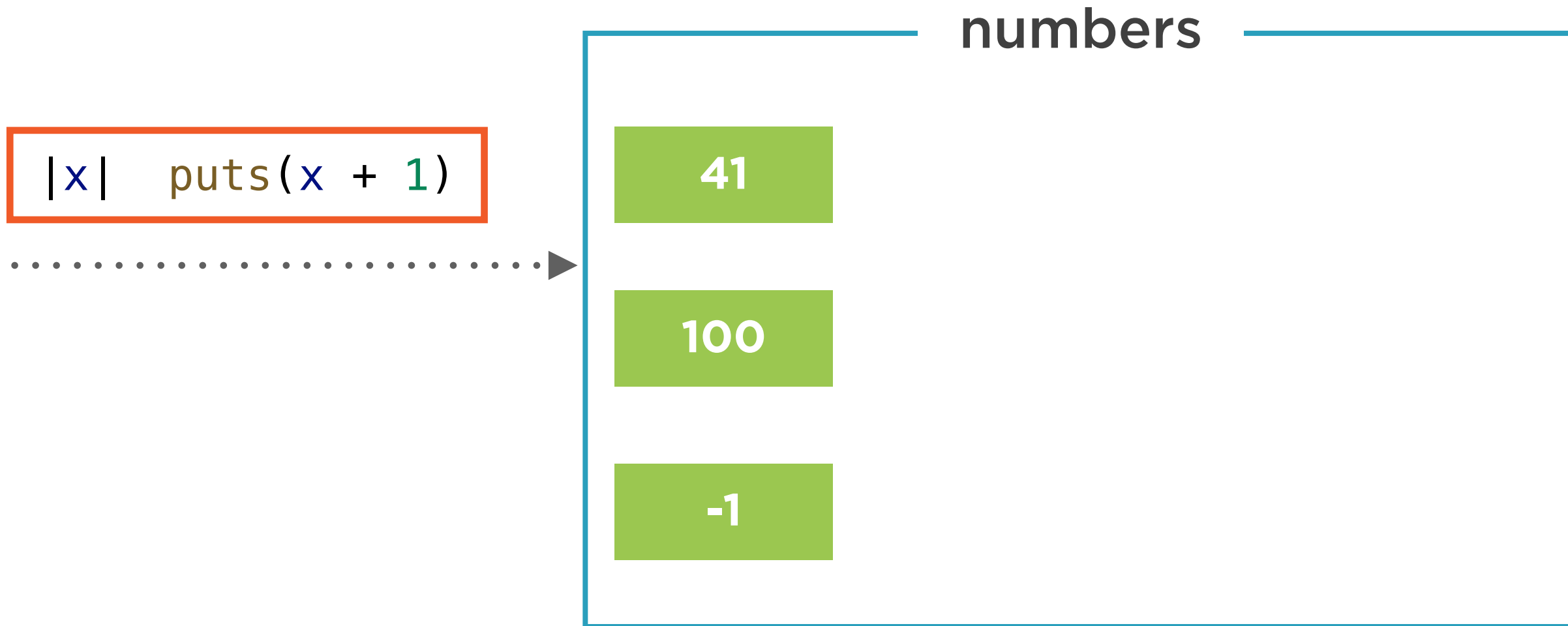


Blocks as Callbacks

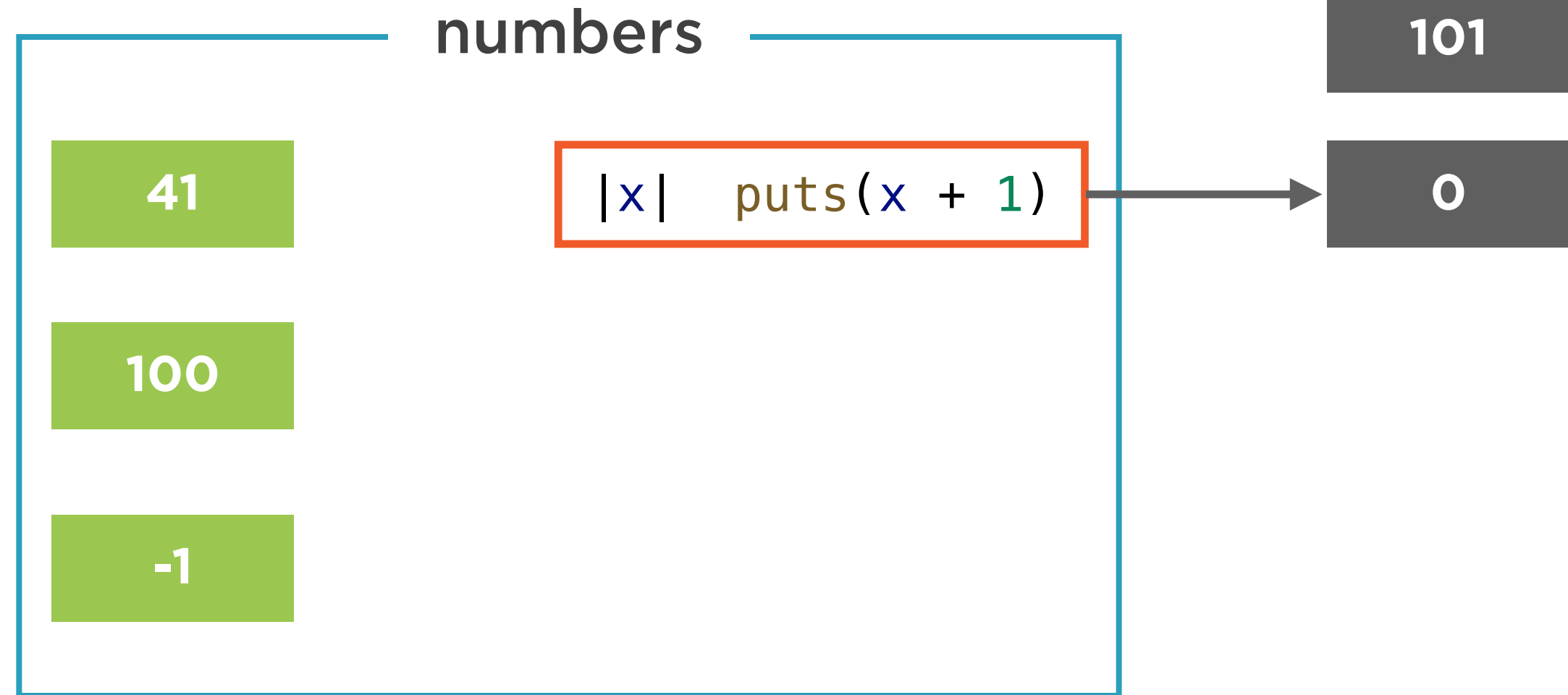


Looping without a Loop

The *each()* Method



The *each()* Method



A Little Extra: More Ways to Iterate

for Loops


```
for i in 0..5  
  puts i  
end
```



0
1
2
3
4
5

while Loops


```
i = 0  
while i <= 5  
    puts i  
    i += 1  
end
```



0
1
2
3
4
5

until Loops

```
i = 0  
  
until i > 5  
    puts i  
    i += 1  
end
```



0
1
2
3
4
5

Using *upto()*


```
0.upto(5) do |i|  
  puts i  
end
```



```
0  
1  
2  
3  
4  
5
```


Using *times()*

```
3.times do  
  puts "Yay!"  
end
```



```
Yay!  
Yay!  
Yay!
```

What We Did

```
1 TEXT_FILE = "romeo-juliet.txt"
2
3 # Load the words from a file
4 def words_from_file(text_file)
5   | File.read(text_file).downcase.gsub(/^[a-z]/, " ").split
6 end
7
8 # Load the list of words in the text
9 words = words_from_file(TEXT_FILE)
10
11 # Create a dictionary of word counts
12 word_count = {}
13 words.each do |w|
14   | # If this word is new, add it to word_count...
15   | word_count[word] = 1
16   | # ...else increment its count
17   | word_count[word] += 1
18 end
19 |
```