In this lecture, we will discuss…

- Ranges
- How they are useful in Ruby

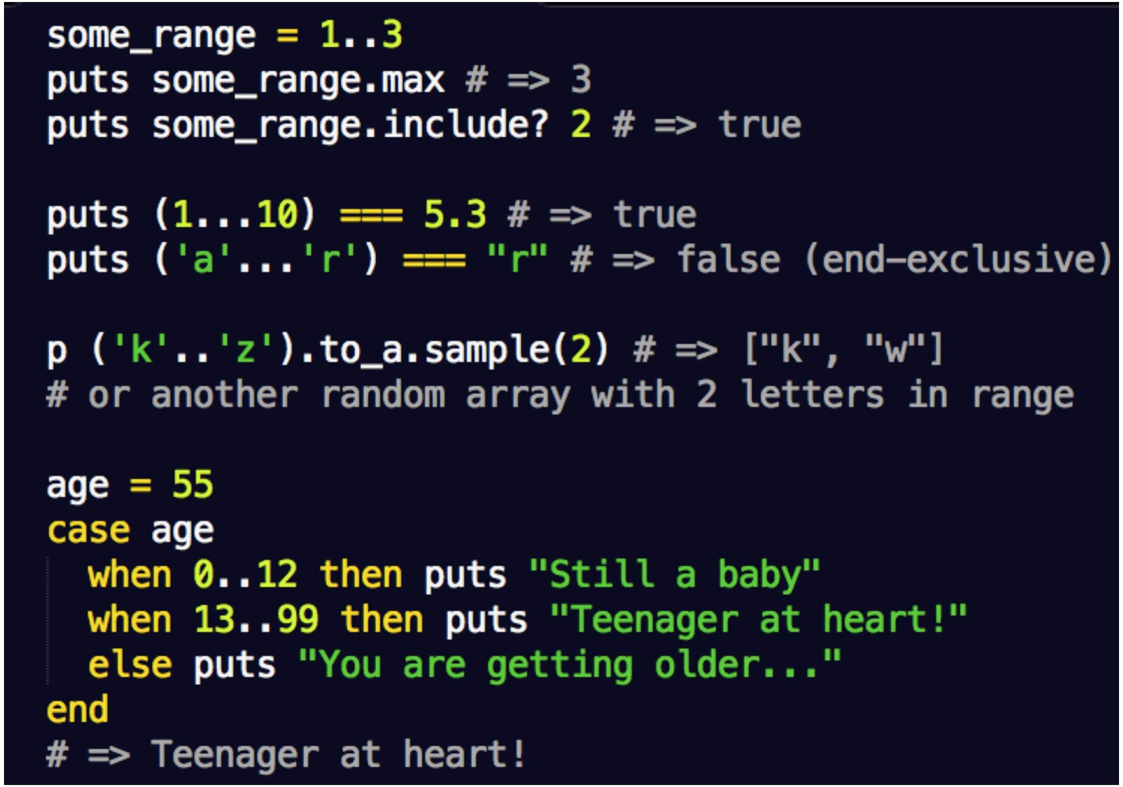
Ranges

- Used to express natural **consecutive** sequences
  - `1..20, ‘a’.. ‘z’`
- **Two** dots → **all-inclusive**
  - `1..10` (1 is **included**, 10 is **included**)
- **Three** dots → **end-exclusive**
  - `1...10` (1 is **included**, 10 is **EXCLUDED**)

Ranges

- **Efficient**
  - Only **start** and **end** stored
- Can be **converted** to an array with `to_a`
- Used for **conditions** and **intervals**

Ranges



```ruby
some_range = 1..3
puts some_range.max # => 3
puts some_range.include? 2 # => true

puts (1...10) === 5.3 # => true
puts ('a'...'r') === "r" # => false (end-exclusive)

p ('k'..'z').to_a.sample(2) # => ["k", "w"]
# or another random array with 2 letters in range

age = 55
case age
  when 0..12 then puts "Still a baby"
  when 13..99 then puts "Teenager at heart!"
  else puts "You are getting older..."
end
# => Teenager at heart!
```

Summary

- ✧ Ranges are **useful** for **consecutive sequences**
- ✧ You can **convert** a range to an array for more **functionality**

What's next?

- ✧ Hashes