Ranges

* Set of values defined by starting value and ending value
* Defined using other ordered values

**Types**

* Inclusive (..)
  + 1..10 => Values from 1 to 10
* Exclusive(...)
  + 1...10 => Values from 1 to 9

Range Definitions

* Number
  + Positive
    - 1..10
  + Negative
    - -10..-1

positive\_range = 1..10

negative\_range = -10..-1

* Floating Point
  + 1.5..2.5

float\_range = 1.5..2.5

* Characters
  + ‘a’ - ‘z’

character\_range = 'a'..'z'

Exercises

puts (1..10).first(4)

puts (10..1).first

puts (10..1).first(4)

puts (1.5..2.5).to\_a

puts (1..).to\_a

**Crack the code**

puts ("ab".."az").to\_a

puts ("a")..("z").to\_a

puts ("a")..("z").to\_s

start = 'x'

ending = 'xxxx'

puts start..ending.to\_a

**Confused between the dots?**

* Returns true if the range excludes its end value.

puts (1...10).exclude\_end?

**Hash Code**

* In Ruby, the #hash method computes a hash value based on the contents of the object

range1 = 1..10

range2 = 1..10

range1.hash.eql?(range2.hash)

**Include**

puts ("ab".."az").include?("ah")

**Size**

* Returns the number of elements in the range

('a'..'z').size

**Infinite Ranges**

range = 1..

range = ..10

**Binary Search**

arr = [1,3,5,7,9,11,13,15,17]

puts (0...arr.size).bsearch {|i| arr[i] == 13}