

**NPTEL MOOC**

# **PROGRAMMING, DATA STRUCTURES AND ALGORITHMS IN PYTHON**

**Week 3, Lecture 1**

**Madhavan Mukund, Chennai Mathematical Institute**

**<http://www.cmi.ac.in/~madhavan>**

# More about `range()`

- \* `range(i, j)` produces the sequence  $i, i+1, \dots, j-1$
- \* `range(j)` automatically starts from 0;  $0, 1, \dots, j-1$
- \* `range(i, j, k)` increments by  $k$ ;  $i, i+k, \dots, i+nk$ 
  - \* Stops with  $n$  such that  $i+nk < j \leq i+(n+1)k$
- \* Count down? Make  $k$  negative!
  - \* `range(i, j, -1)`,  $i > j$ , produces  $i, i-1, \dots, j+1$

# More about `range()`

- \* General rule for `range(i, j, k)`
  - \* Sequence starts from `i` and gets as close to `j` as possible without crossing `j`
- \* If `k` is positive and `i >= j`, empty sequence
  - \* Similarly if `k` is negative and `i <= j`
- \* If `k` is negative, stop “before” `j`
  - \* `range(12, 1, -3)` produces 12, 9, 6, 3

# More about `range()`

- \* Why does `range(i, j)` stop at `j-1`?
  - \* Mainly to make it easier to process lists
  - \* List of length `n` has positions `0, 1, ..., n-1`
  - \* `range(0, len(l))` produces correct range of valid indices
    - \* Easier than writing `range(0, len(l)-1)`

# range() and lists

- \* Compare the following
  - \* `for i in [0,1,2,3,4,5,6,7,8,9]:`
  - \* `for i in range(0,10):`
- \* Is `range(0,10) == [0,1,2,3,4,5,6,7,8,9]`?
  - \* In Python2, yes
  - \* In Python3, no!

# range() and lists

- \* Can convert `range()` to a list using `list()`
  - \* `list(range(0,5)) == [0,1,2,3,4]`
- \* Other type conversion functions using type names
  - \* `str(78) = "78"`
  - \* `int("321") = 321`
    - \* But `int("32x")` yields error

# Summary

- \* `range(n)` has is implicitly from `0` to `n-1`
- \* `range(i, j, k)` produces sequence in steps of `k`
  - \* Negative `k` counts down
- \* Sequence produced by `range()` is not a list
  - \* Use `list(range(...))` to get a list