

LAB 4

Topics: while loops, for loops, break, continue statements

While loop - executes code indefinitely until condition is false

Rep. as follows:

While condition:

code

Example:

while True: \leftarrow Runs forever, why?
print("stuck here")

Def: Iterable - anything that can be looped over
EX: [1, 2, 3], (4, 5, 6), "hello"

range() - generates a list of numbers

range(start, stop, step)

\uparrow
start
#

\uparrow
stop

Not
inclusive

\uparrow
Diff between
each #

range(0, 3, 1) \rightarrow [0, 1, 2]

len() - returns length (# of items) in object
len("Hello") $\rightarrow 5$
len([]) $\rightarrow 0$

For loops

2 ways to write a for loop in Python

① The "Pythonic" way
for i in iterable:

code

② The C way
for i in range(len(iterable)):

code

Which should you use?

(preferred)
Can also use enumerate
for indexing / looping

Use/learn the Pythonic way unless you need the index
unless if you need to update in place

Ex: -5-4-3-2-1
S = "Hello"
0 1 2 3 4

① for char in S:
print(char)

H E L L O

② for char in range(len(S)):
print(S[char])

H E L L O

Break statement - immediately exits loop currently inside of

In Python def as: break

while True: } immediately exits
break

Continue - immediately starts next iteration of loop

In Python rep. as: continue

Lists - a Python datastructure

Similar to arrays in other langs.

Comma Separated, can be nested, passed by reference

X = [1, 2, 3]

Y = [1, "Hello", (1, 2)]

Z = [[1, 2]] → 2D List

Various method/function calls for lists

- append(...)
- extend(...)
- insert(...)

⋮