COMP 110

CL06 - for loops + sequence

Sequences

What is a Sequence?

An Abstract Data Type that is an ordered, 0-indexed set of values.

There are many specific types of sequences with their own properties. Common, built-in sequence types in Python include:

- str: a sequence of character data
- list: a dynamically sized sequence of values of a specific type
- tuple: a fixed size sequence of values of any types
- range: a sequence of integers at intervals between a start and end

Tuples

- Tuples types are made of a specific, fixed-length sequence of any mixed type(s).
- Example:

```
3d_coordinate: tuple[float, float, float] = (1.0, 1.0, 1.0)
```

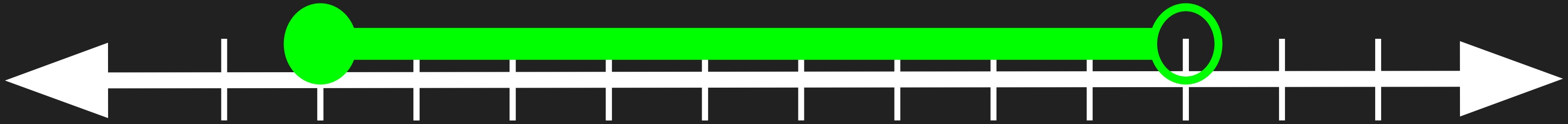
- Other example:

```
Player = tuple[str, int]

lebron : Player = ("James", 6)

mj: Player = ("Jordan", 23)
```

Range



- Includes start point, does **not** include end point, and *steps* through every point in between
- Constructor: range(start, end, [step = 1])
- Examples:
  - range(1, 5) stops at numbers 1, 2, 3, 4
  - range(1, 6, 2) stops at numbers 1, 3, 5

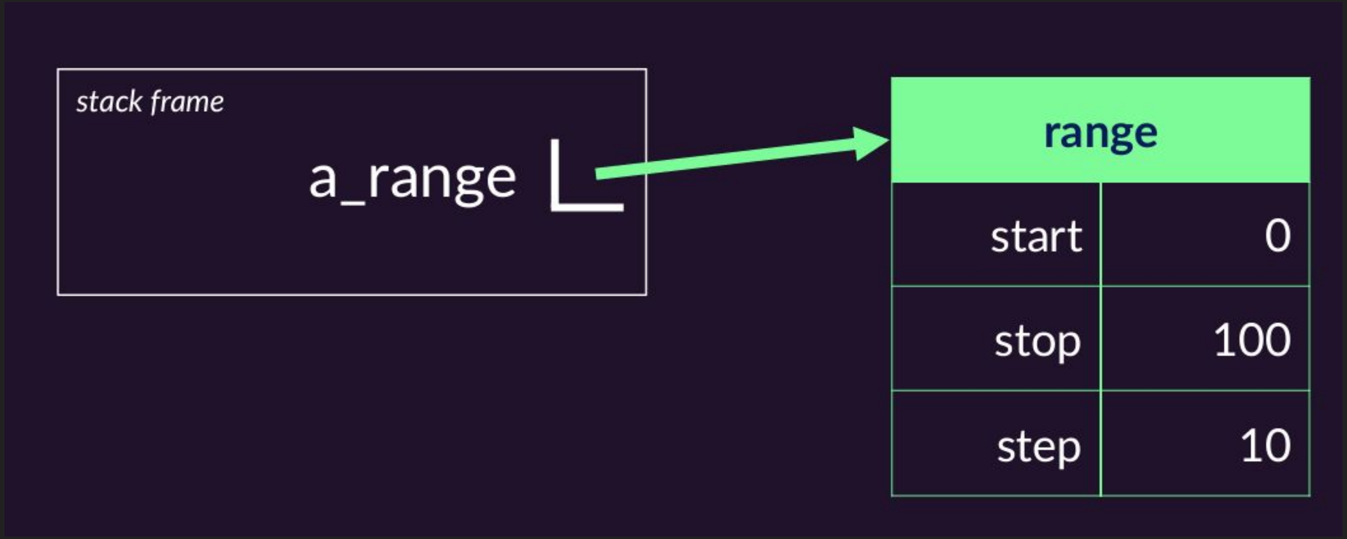
if you dont specify the step it defaults to step=1

Tuples + range() in Memory

On the heap, but don’t worry about it. :-)

Range in memory

Stored on heap. Always has three attributes: start, stop, step



for … in … loops

xs: list[int] = [1, 2, 3]

Print every element of xs

while

for … in …

Example 2

pets: list[str] = [“Louie”, “Bo”, “Bear”]

Using a for … in … loop, write code to tell each pet they’re a good boy!

Challenge: call each elem something other than “elem”

Output should be:

Good boy, Louie!

Good boy, Bo!

Good boy, Bear!

Using range() in a for … in … loop.

names: list[str] = [“Alyssa”, “Janet”, “Vrinda”]

Print every element’s index and value:

```
0: Alyssa

1: Janet

2: Vrinda
```

Challenge Question:

In your workspace, in the **lessons** folder, create the file **sum.py**

We are going to write the same function *three different ways!*

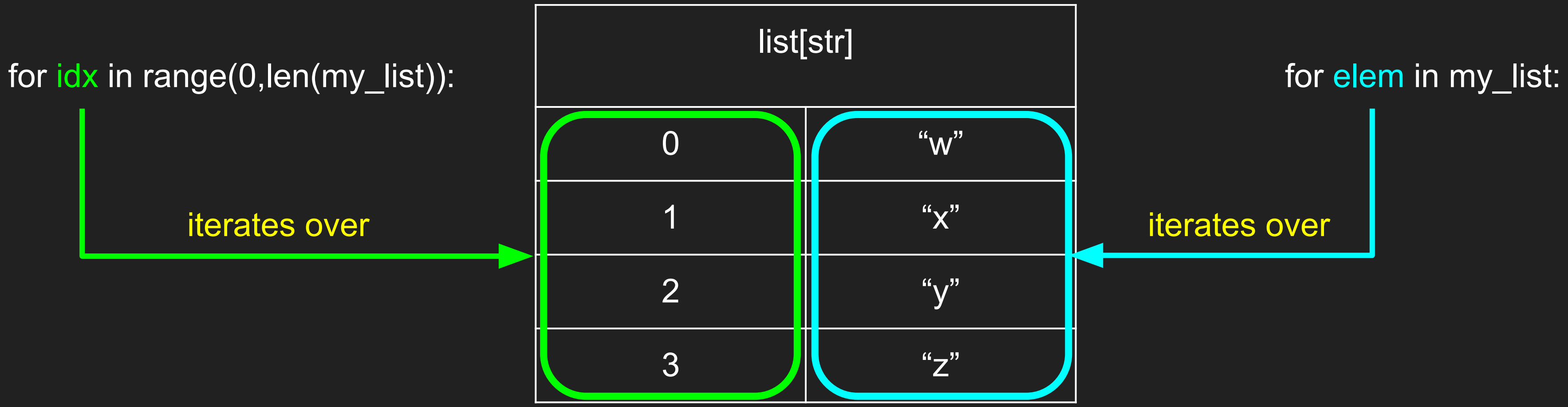
This function sums all the elements of the input vals: list[float]

For example, w_sum([1.1, 0.9, 1.0]) should compute 1.1 + 0.9 + 1.0 and return the simplified value 3.0.

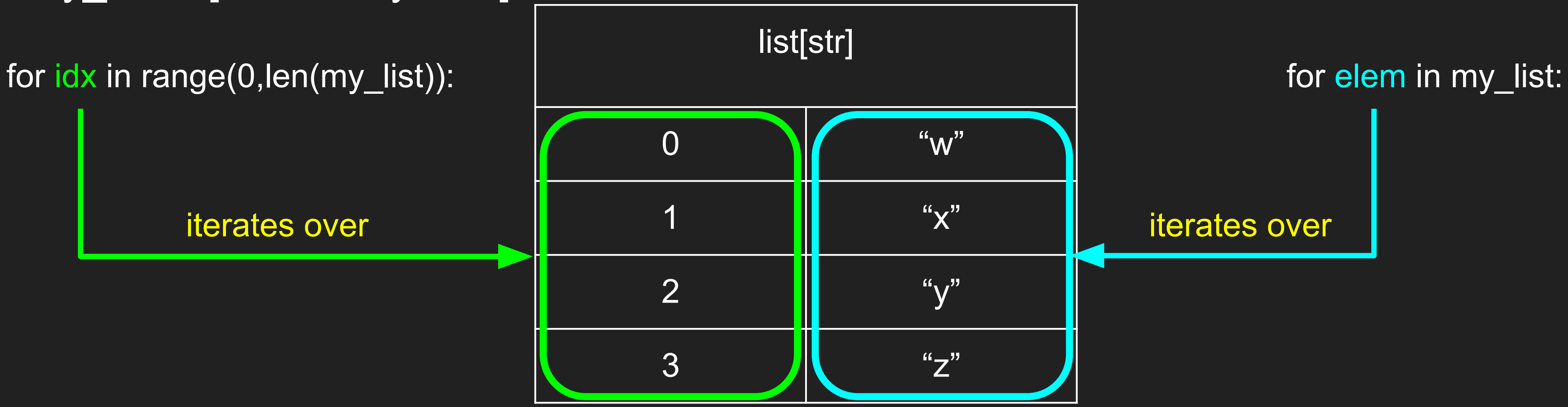
- Version A: Write a function called w_sum that uses a while loop to iterate through vals
- Version B: Write a function called f_sum that uses a for ... in ... loop.
- Version C: Write a function called f_range_sum that uses a for … in range(len(xs)) loop.

More info + submission instructions on the website!

my_list = [“w”, “x”, “y”, “z”]


for idx in range(0,len(my_list)):
iterates over
list[str]
0
1
2
3
“w”
“x”
“y”
“z”
for elem in my_list:
iterates over

```python
my_list = ["w", "x", "y", "z"]
```



```
for idx in range(0,len(my_list)):
        print(idx)

Output:
                 0
                 1
                 2
                 3
```

```
for idx in range(0,len(my_list)):
        print(my_list[idx])

Output:
                 w
                 x
                 y
                 z
```

```
for elem in my_list:
        print(elem)

Output:
                 w
                 x
                 y
                 z
```