

COMP  
110

CL10 - for loops + range()

# 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

## 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**

```
a_range = range(0,100,10)
```



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

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

iterates over

list[str]	
0	"w"
1	"x"
2	"y"
3	"z"

```
for elem in my_list:
```

iterates over

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

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

iterates over

list[str]	
0	"w"
1	"x"
2	"y"
3	"z"

```
for elem in my_list:
```

iterates over

```
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

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]** and **returns** the sum.

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(0,len(xs))** loop.

More info + submission instructions on the website!