Review

Note: **return** statements can happen anywhere in your function; a function may have many return statements; however, the first return statement you encounter during execution exits the function!

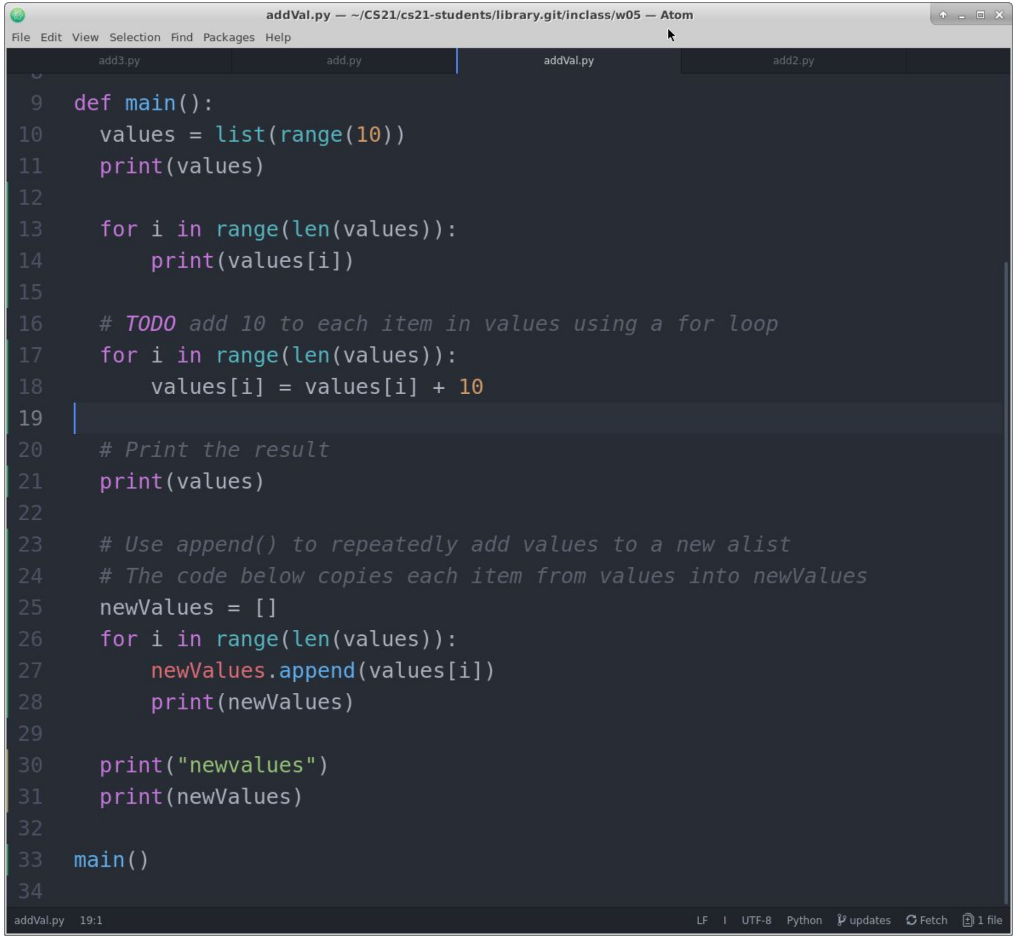
Note: lists are **mutable**, strings are **immutable**

lists have similar operations to strings: +, *, len(), []

You can add values to a list using append()

Example code

```
almond[w05]$ python3 addVal.py
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
0
1
2
3
4
5
6
7
8
9
[10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
[10]
[10, 11]
[10, 11, 12]
[10, 11, 12, 13]
[10, 11, 12, 13, 14]
[10, 11, 12, 13, 14, 15]
[10, 11, 12, 13, 14, 15, 16]
[10, 11, 12, 13, 14, 15, 16, 17]
[10, 11, 12, 13, 14, 15, 16, 17, 18]
[10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
newvalues
[10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

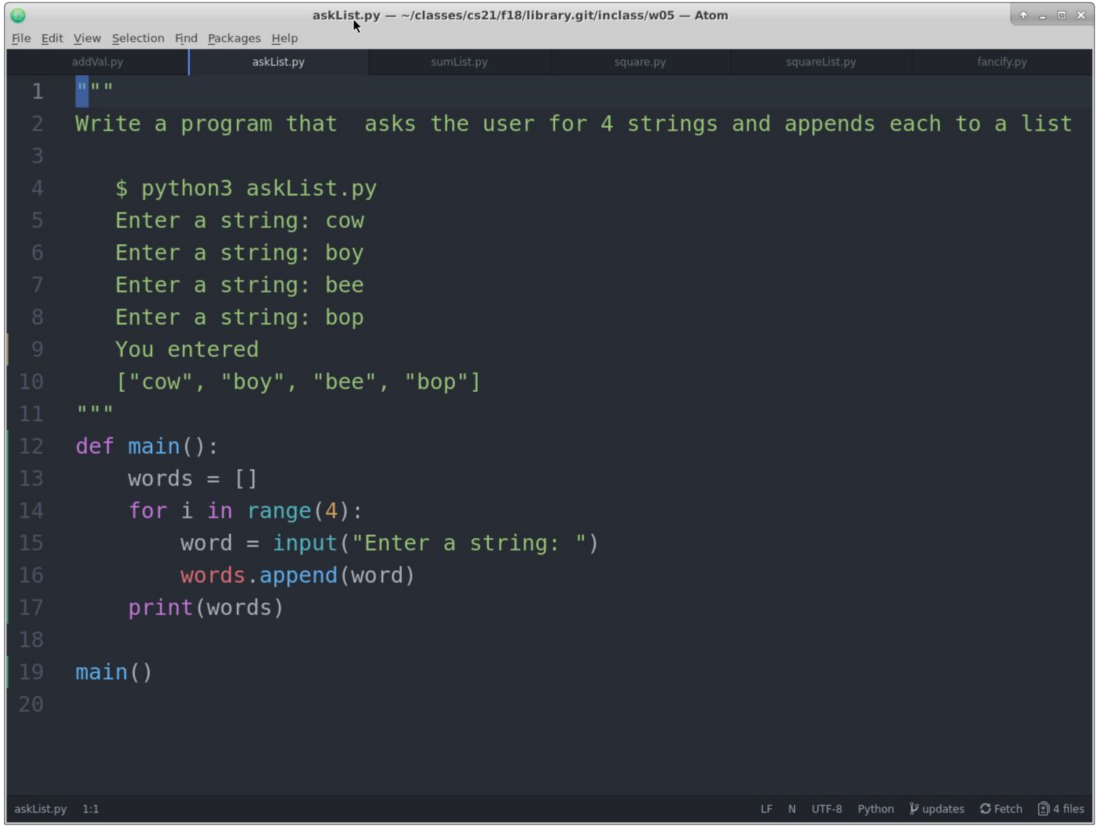

addVal.py — ~/CS21/cs21-students/library.git/inclass/w05 — Atom
File Edit View Selection Find Packages Help
add3.py
add.py
addVal.py
add2.py
addVal.py 19:1
LF
UTF-8
Python
updates
Fetch
1 file

```python
def main():
  values = list(range(10))
  print(values)

  for i in range(len(values)):
      print(values[i])

  # TODO add 10 to each item in values using a for loop
  for i in range(len(values)):
      values[i] = values[i] + 10

  # Print the result
  print(values)

  # Use append() to repeatedly add values to a new alist
  # The code below copies each item from values into newValues
  newValues = []
  for i in range(len(values)):
      newValues.append(values[i])
      print(newValues)

  print("newvalues")
  print(newValues)

main()
```

askList.py

Idea:

- Create an empty list
- Repeat 4 times
  - Ask the user for a word
  - Append the word to the list

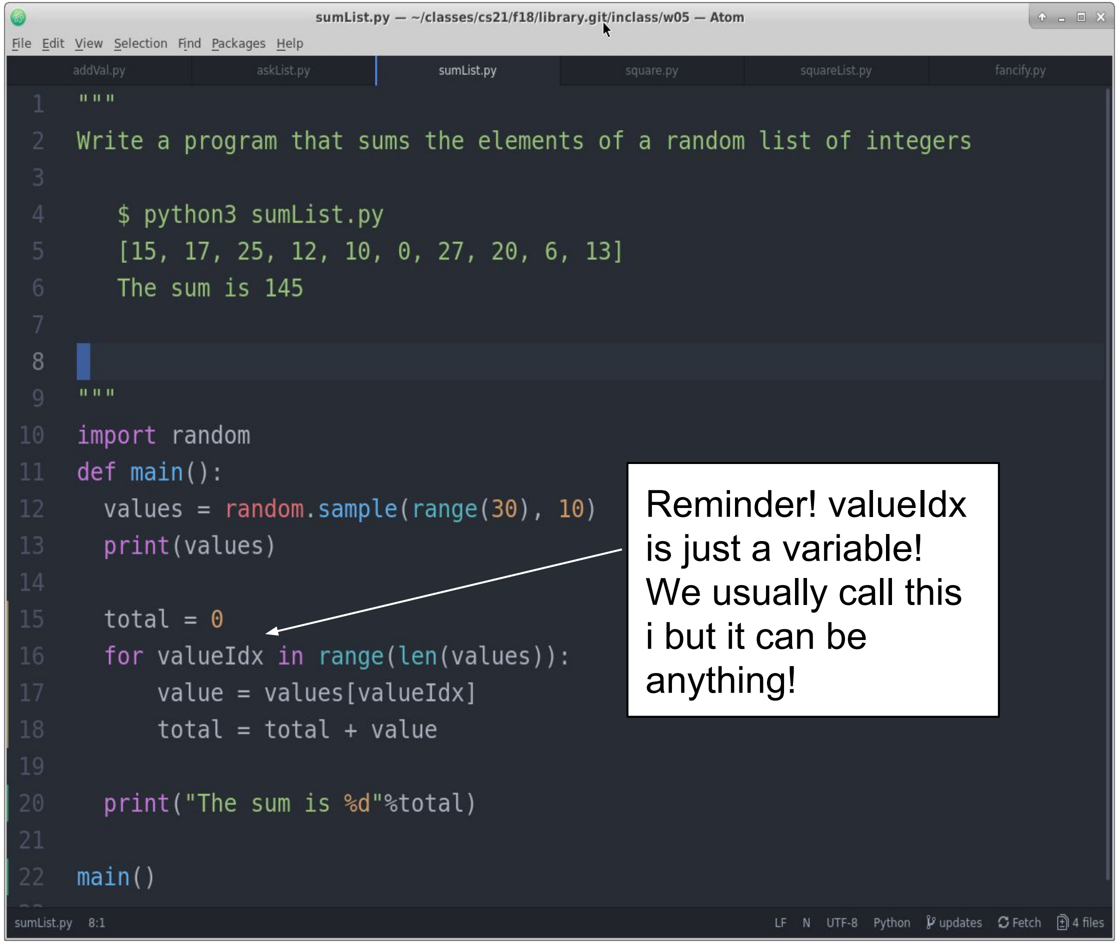

askList.py — ~/classes/cs21/f18/library.git/inclass/w05 — Atom
File Edit View Selection Find Packages Help
addVal.py
askList.py
sumList.py
square.py
squareList.py
fancify.py
askList.py 1:1
LF N UTF-8 Python updates Fetch 4 files

```
"""
Write a program that  asks the user for 4 strings and appends each to a list

   $ python3 askList.py
   Enter a string: cow
   Enter a string: boy
   Enter a string: bee
   Enter a string: bop
   You entered
   ["cow", "boy", "bee", "bop"]
"""
def main():
    words = []
    for i in range(4):
        word = input("Enter a string: ")
        words.append(word)
    print(words)

main()

```

sumList.py

Idea:

- Initialize a list to 10 random values (line 12)
- Create an accumulator (line 15)
- Go through each item in the list and add its value to total


sumList.py — ~/classes/cs21/f18/library.git/inclass/w05 — Atom
File Edit View Selection Find Packages Help
addVal.py
askList.py
sumList.py
square.py
squareList.py
fancify.py
sumList.py 8:1
LF N UTF-8 Python updates Fetch 4 files

```
1  """
2  Write a program that sums the elements of a random list of integers
3
4     $ python3 sumList.py
5     [15, 17, 25, 12, 10, 0, 27, 20, 6, 13]
6     The sum is 145
7
8
9  """
10 import random
11 def main():
12   values = random.sample(range(30), 10)
13   print(values)
14
15   total = 0
16   for valueIdx in range(len(values)):
17       value = values[valueIdx]
18       total = total + value
19
20   print("The sum is %d"%total)
21
22 main()
```

Reminder! valueIdx is just a variable! We usually call this i but it can be anything!