

# Changing values of parameters

If a variable is **immutable** and you change its value inside a function, the change **does not last** past the lifetime of the function

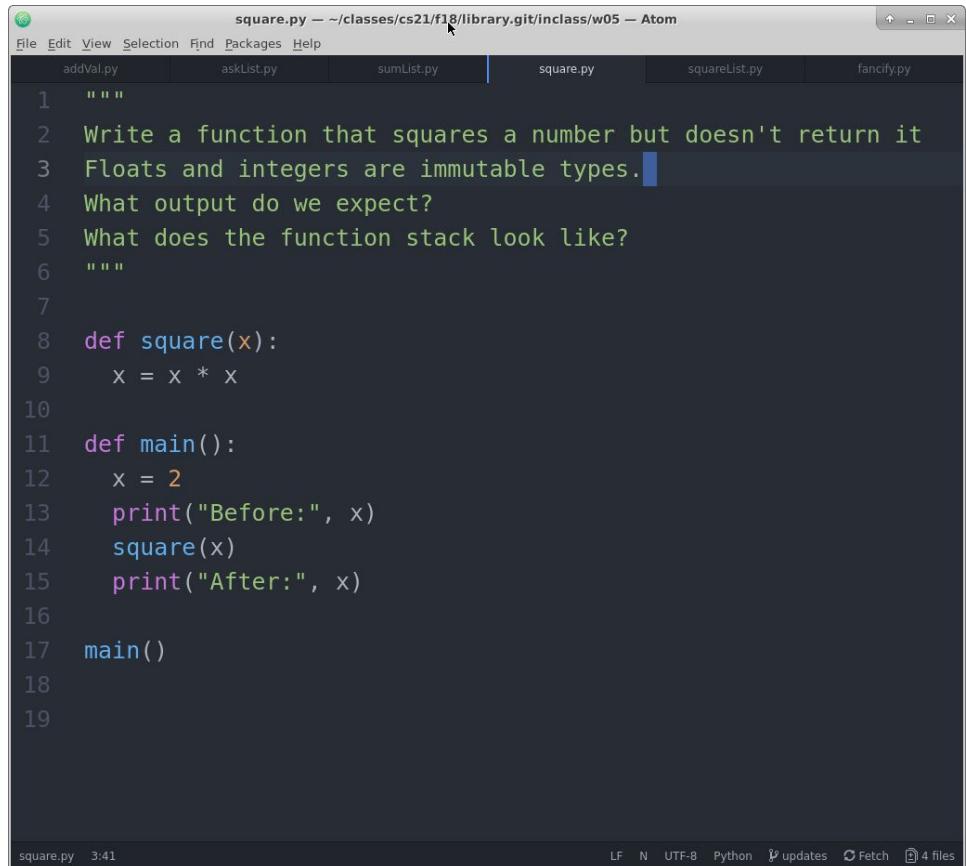
If a variable is **mutable** and you change its value inside a function, the change is permanent

# Example: square.py

```
almond[w05]$ python3 square.py
```

Before: 2

After: 2



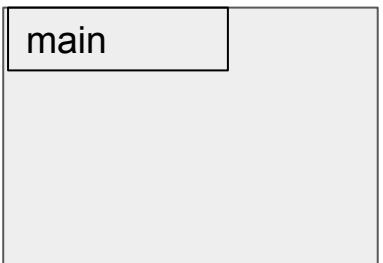
The screenshot shows a dark-themed Atom code editor window titled "square.py — ~/classes/cs21/f18/library.git/inclass/w05 — Atom". The file tab bar includes "addVal.py", "askList.py", "sumList.py", "square.py" (which is the active tab), "squareList.py", and "fancify.py". The code in "square.py" is as follows:

```
1 """
2 Write a function that squares a number but doesn't return it
3 Floats and integers are immutable types.
4 What output do we expect?
5 What does the function stack look like?
6 """
7
8 def square(x):
9     x = x * x
10
11 def main():
12     x = 2
13     print("Before:", x)
14     square(x)
15     print("After:", x)
16
17 main()
18
19
```

The status bar at the bottom shows "square.py 3:41" and icons for LF, N, UTF-8, Python, updates, Fetch, and 4 files.

# Function stack diagram

Function Stack



Heap



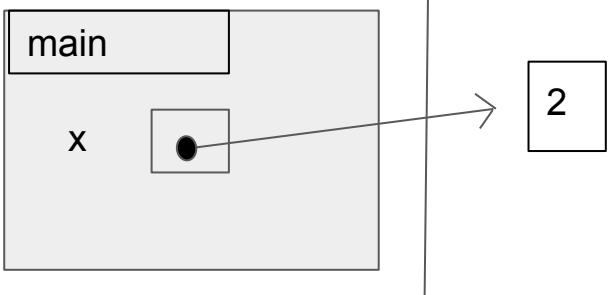
The screenshot shows a code editor window for the file "square.py". The code defines a square function that takes a parameter x and returns its square. It also defines a main function that initializes x to 2, prints its initial value, calls square(x), and then prints its value again after the square operation.

```
square.py — ~/classes/cs21/f18/library.git/inclass/w05 — Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py
square.py
1 """
2 Write a function that squares a number but do
3 Floats and integers are immutable types.
4 What output do we expect?
5 What does the function stack look like?
6 """
7
8 def square(x):
9     x = x * x
10
11 def main():
12     x = 2
13     print("Before:", x)
14     square(x)
15     print("After:", x)
16
17 main()
18
19
```

square.py 3:41 LF N UTF-8

# Function stack diagram

Function Stack



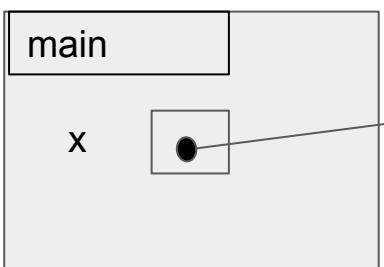
Heap

A screenshot of an Atom code editor showing a file named 'square.py'. The code defines a square function that takes a parameter x and returns its square. It also defines a main function that initializes x to 2, prints its initial value, calls the square function, and then prints its value after being squared. A red box highlights the line 'def main(): x = 2'. The status bar at the bottom shows the file name 'square.py', line number '3:41', and encoding 'UTF-8'.

```
square.py — ~/classes/cs21/f18/library.git/inclass/w05 — Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py
1 """
2 Write a function that squares a number but do
3 Floats and integers are immutable types.
4 What output do we expect?
5 What does the function stack look like?
6
7
8 def square(x):
9     x = x * x
10
11 def main():
12     x = 2
13     print("Before:", x)
14     square(x)
15     print("After:", x)
16
17 main()
18
19
```

# Function stack diagram

Function Stack

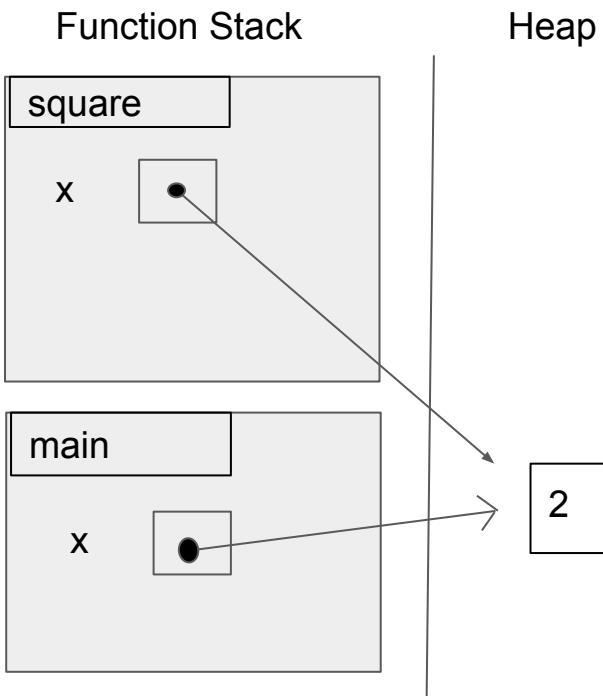


Heap

A screenshot of an Atom code editor showing a file named "square.py". The code defines a function "square" that takes a parameter "x" and returns its square. It also defines a "main" function that prints the value of "x" before and after calling "square". A red box highlights the line "print("Before:", x)".

```
square.py — ~/classes/cs21/f18/library.git/inclass/w05 — Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py
square.py
1 """
2 Write a function that squares a number but do
3 Floats and integers are immutable types.
4 What output do we expect?
5 What does the function stack look like?
6
7
8 def square(x):
9     x = x * x
10
11 def main():
12     x = 2
13     print("Before:", x)
14     square(x)
15     print("After:", x)
16
17 main()
18
19
```

# Function stack diagram

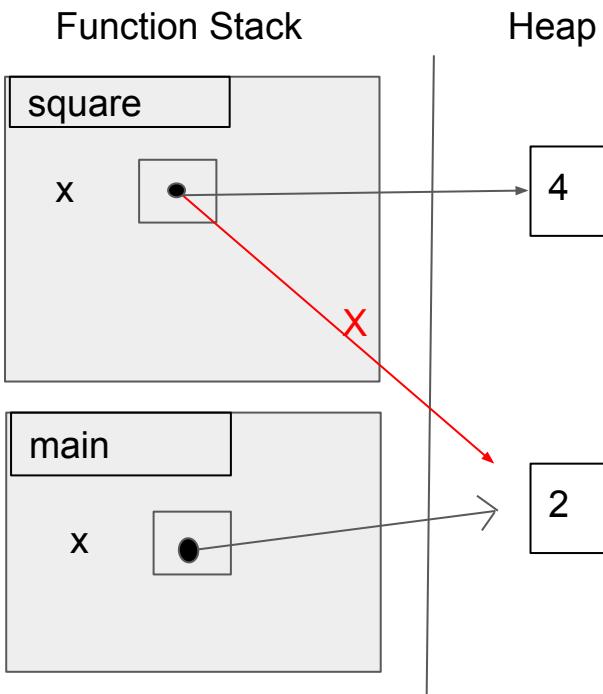


square.py — ~/classes/cs21/f18/library.git/inclass/w05 — Atom

```
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py
square.py
1 """
2 Write a function that squares a number but do
3 Floats and integers are immutable types.
4 What output do we expect?
5 What does the function stack look like?
6 """
7
8 def square(x):
9     x = x * x
10
11 def main():
12     x = 2
13     print("Before:", x)
14     square(x)
15     print("After:", x)
16
17 main()
18
19
```

square.py 3:41 LF N UTF-8

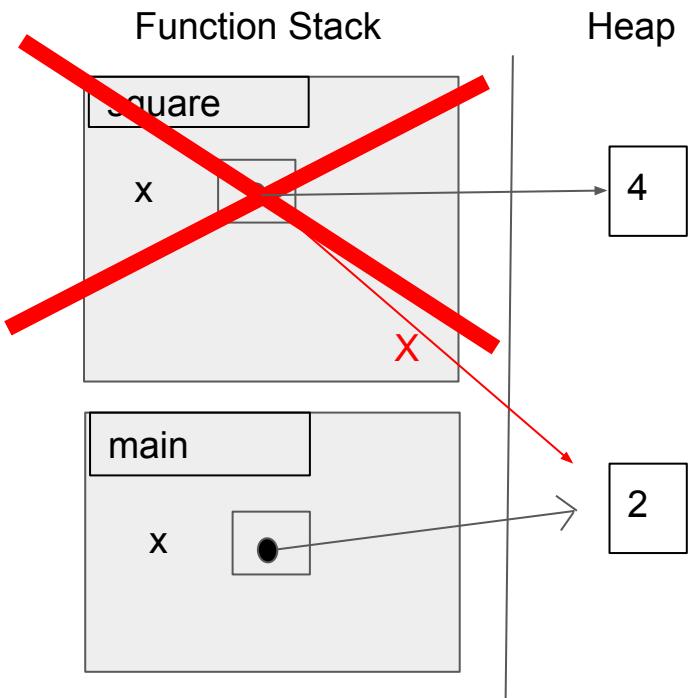
# Function stack diagram



```
File Edit View Selection Find Packages Help
square.py — ~/classes/cs21/f18/library.git/inclass/w05 — Atom
addVal.py askList.py sumList.py square.py
1 """
2 Write a function that squares a number but do
3 Floats and integers are immutable types.
4 What output do we expect?
5 What does the function stack look like?
6 """
7
8 def square(x):
9     x = x * x
10
11 def main():
12     x = 2
13     print("Before:", x)
14     square(x)
15     print("After:", x)
16
17 main()
18
19
```

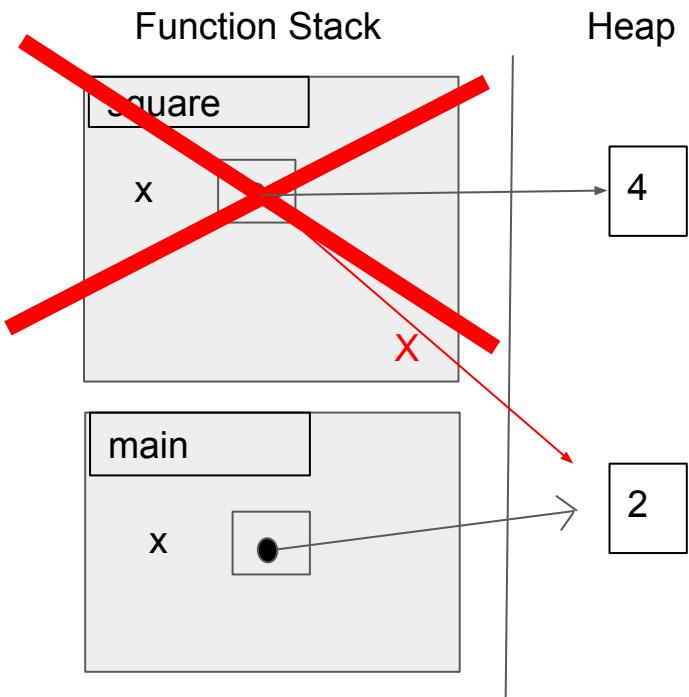
square.py 3:41 LF N UTF-8

# Function stack diagram



```
square.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py
square.py 3:41 LF N UTF-8
1 """
2 Write a function that squares a number but do
3 Floats and integers are immutable types.
4 What output do we expect?
5 What does the function stack look like?
6 """
7
8 def square(x):
9     x = x * x
10
11 def main():
12     x = 2
13     print("Before:", x)
14     square(x)
15     print("After:", x)
16
17 main()
```

# Function stack diagram



square.py — ~/classes/cs21/f18/library.git/inclass/w05 — Atom

```
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py
square.py
1 """
2 Write a function that squares a number but do
3 Floats and integers are immutable types.
4 What output do we expect?
5 What does the function stack look like?
6 """
7
8 def square(x):
9     x = x * x
10
11 def main():
12     x = 2
13     print("Before:", x)
14     square(x)
15     print("After:", x)
16
17 main()
18
19
```

square.py 3:41 LF N UTF-8

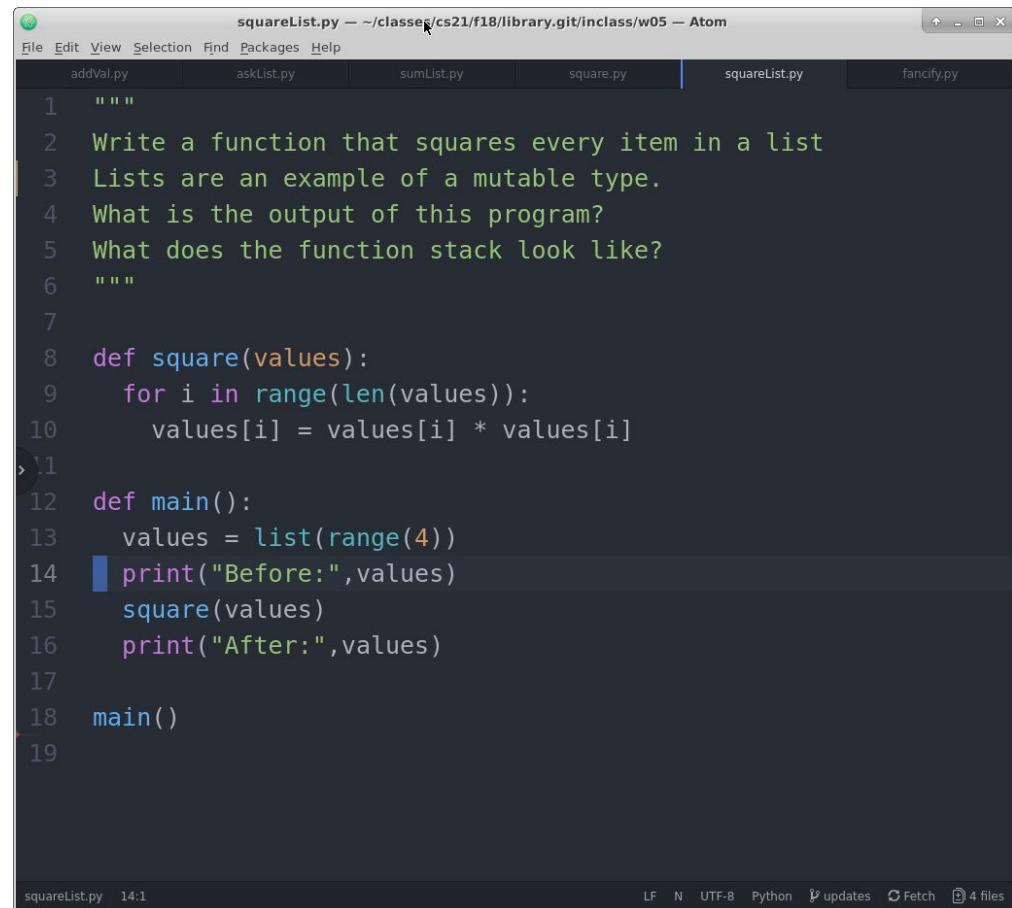


# Example: squareList.py

```
almond[w05]$ python3 squareList.py
```

Before: [0, 1, 2, 3]

After: [0, 1, 4, 9]



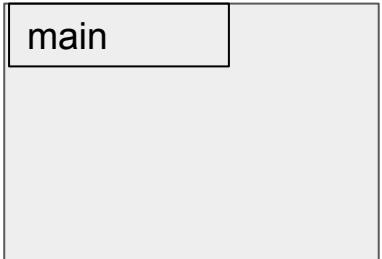
A screenshot of the Atom code editor showing the file `squareList.py`. The code defines a function `square` that takes a list of values and squares each item. It then calls `square` from within the `main` function, printing the original and squared lists.

```
squareList.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
1 """
2 Write a function that squares every item in a list
3 Lists are an example of a mutable type.
4 What is the output of this program?
5 What does the function stack look like?
6 """
7
8 def square(values):
9     for i in range(len(values)):
10         values[i] = values[i] * values[i]
11
12 def main():
13     values = list(range(4))
14     print("Before:", values)
15     square(values)
16     print("After:", values)
17
18 main()
19
```

squareList.py 14:1 LF N UTF-8 Python 4 updates 4 files

# Function Stack diagram

Function Stack



Heap



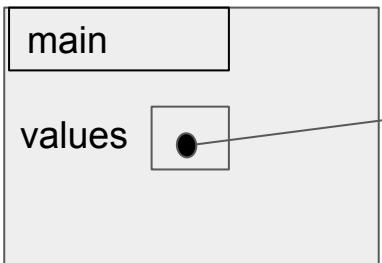
The screenshot shows a code editor window titled 'squareList.py'. The menu bar includes 'File', 'Edit', 'View', 'Selection', 'Find', 'Packages', and 'Help'. The tabs at the top show 'addVal.py', 'askList.py', 'sumList.py', 'square.py', and 'squareList.py'. The code itself is:

```
1 """  
2 Write a function that squares every item in a list.  
3 Lists are an example of a mutable type.  
4 What is the output of this program?  
5 What does the function stack look like?  
6 """  
7  
8 def square(values):  
9     for i in range(len(values)):  
10         values[i] = values[i] * values[i]  
11  
12 def main():  
13     values = list(range(4))  
14     print("Before:", values)  
15     square(values)  
16     print("After:", values)  
17  
18 main()  
19
```

The 'main()' function is highlighted with a thick red rectangular box. The status bar at the bottom indicates 'squareList.py 14:1'.

# Function Stack diagram

Function Stack



Heap

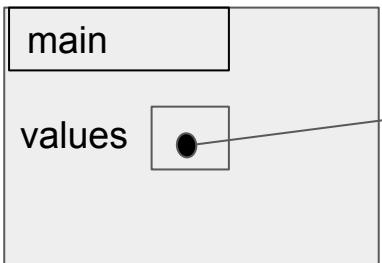
squareList.py — ~/classes/cs21/f18/library.git/inclass/w05 — Atom

```
File Edit View Selection Find Packages Help  
addVal.py askList.py sumList.py square.py squareList.py  
1 """  
2 Write a function that squares every item in a list.  
3 Lists are an example of a mutable type.  
4 What is the output of this program?  
5 What does the function stack look like?  
6 """  
7  
8 def square(values):  
9     for i in range(len(values)):  
10         values[i] = values[i] * values[i]  
11  
12 def main():  
13     values = list(range(4))  
14     print("Before:", values)  
15     square(values)  
16     print("After:", values)  
17  
18 main()  
19
```

squareList.py 14:1 LF N UTF-8 Python

# Function Stack diagram

Function Stack



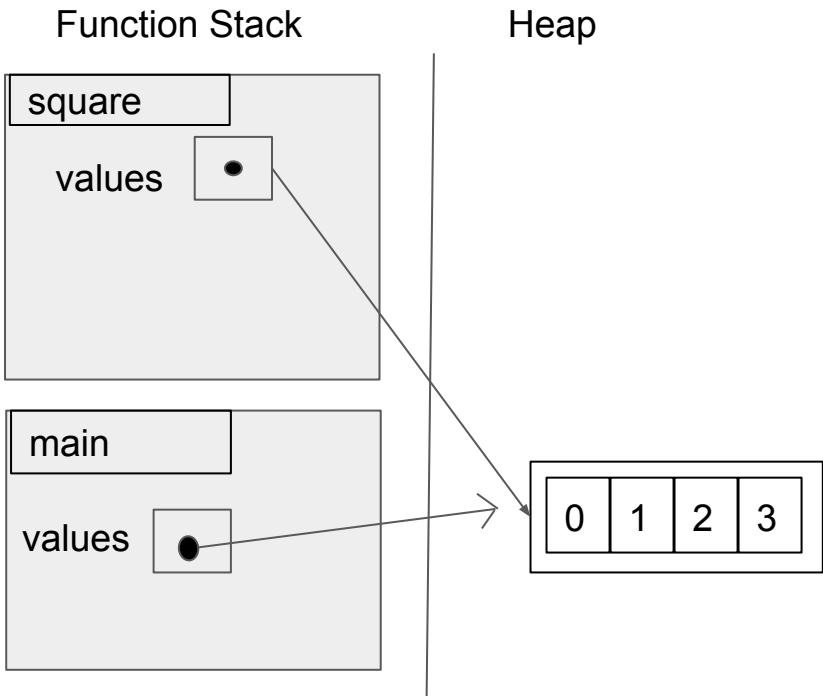
Heap

A screenshot of a code editor window titled "squareList.py". The window shows a Python script with the following content:

```
1 """  
2 Write a function that squares every item in a list.  
3 Lists are an example of a mutable type.  
4 What is the output of this program?  
5 What does the function stack look like?  
6 """  
7  
8 def square(values):  
9     for i in range(len(values)):  
10         values[i] = values[i] * values[i]  
11  
12 def main():  
13     values = list(range(4))  
14     print("Before:", values)  
15     square(values)  
16     print("After:", values)  
17  
18 main()  
19
```

The line `14 print("Before:", values)` is highlighted with a red rectangle.

# Function Stack diagram



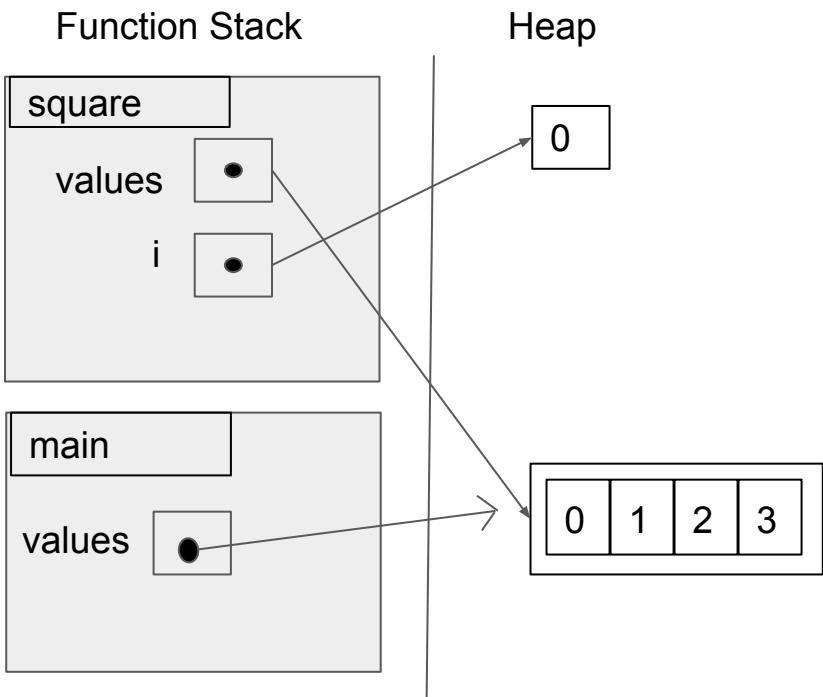
squareList.py — ~/classes/cs21/f18/library.git/inclass/w05 — Atom

File Edit View Selection Find Packages Help  
addVal.py askList.py sumList.py square.py squareList.py

```
1 """
2 Write a function that squares every item in a list.
3 Lists are an example of a mutable type.
4 What is the output of this program?
5 What does the function stack look like?
6 """
7
8 def square(values):
9     for i in range(len(values)):
10         values[i] = values[i] * values[i]
11
12 def main():
13     values = list(range(4))
14     print("Before:", values)
15     square(values)
16     print("After:", values)
17
18 main()
19
```

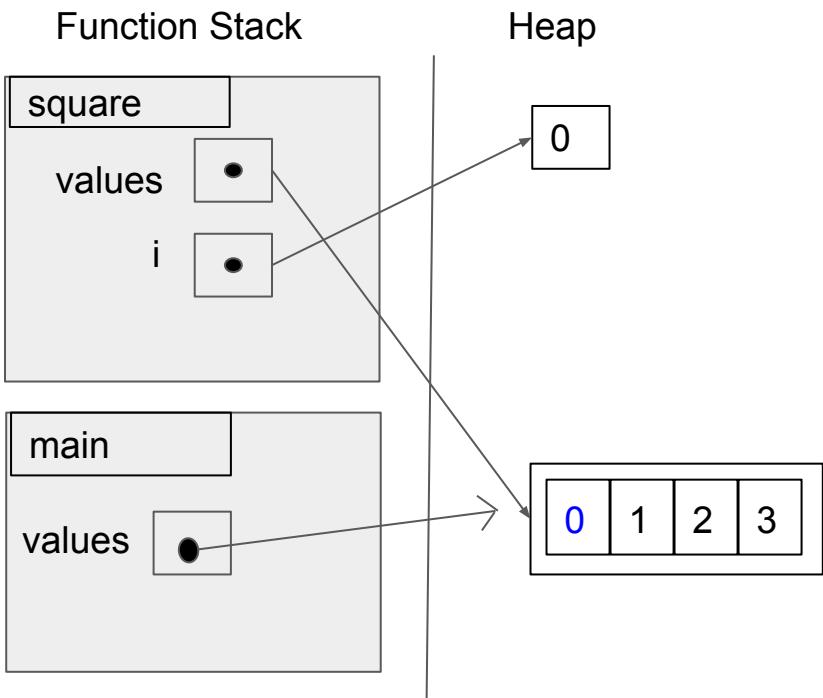
squareList.py 14:1 LF N UTF-8 Python

# Function Stack diagram



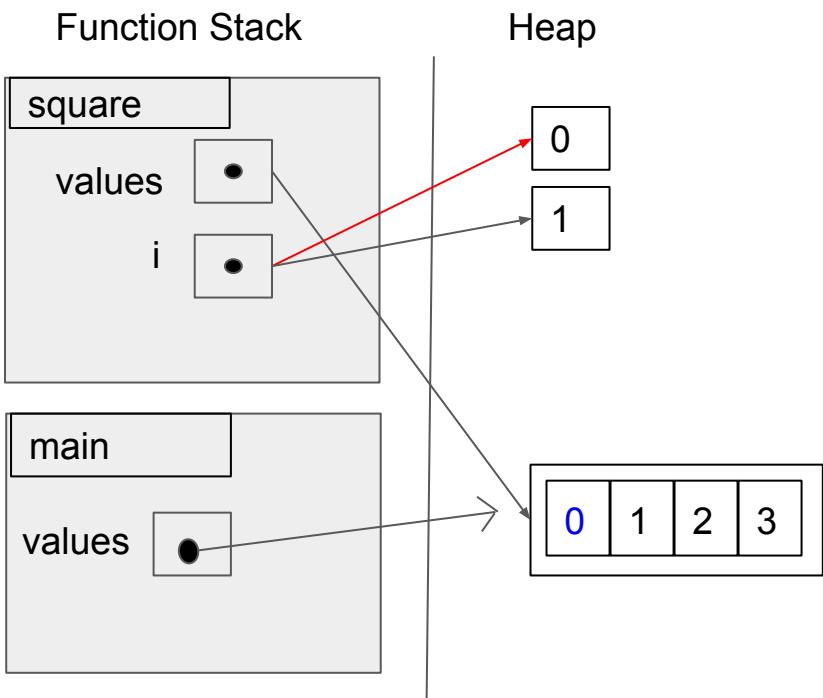
```
squareList.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squareList.py
1 """
2 Write a function that squares every item in a list.
3 Lists are an example of a mutable type.
4 What is the output of this program?
5 What does the function stack look like?
6 """
7
8 def square(values):
9     for i in range(len(values)):
10         values[i] = values[i] * values[i]
11
12 def main():
13     values = list(range(4))
14     print("Before:", values)
15     square(values)
16     print("After:", values)
17
18 main()
19
```

# Function Stack diagram



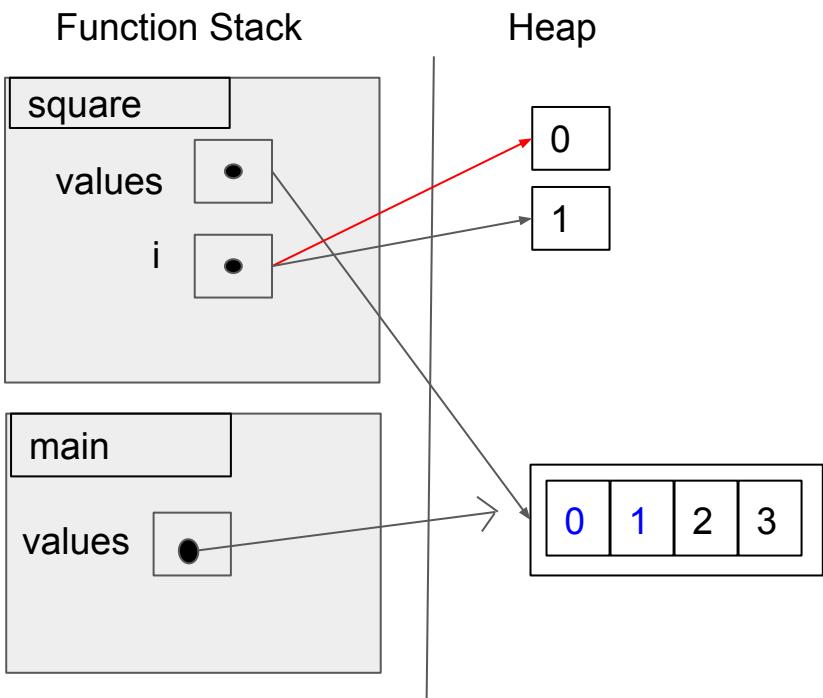
```
squareList.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squareList.py
1 """
2 Write a function that squares every item in a list.
3 Lists are an example of a mutable type.
4 What is the output of this program?
5 What does the function stack look like?
6 """
7
8 def square(values):
9     for i in range(len(values)):
10         values[i] = values[i] * values[i]
11
12 def main():
13     values = list(range(4))
14     print("Before:", values)
15     square(values)
16     print("After:", values)
17
18 main()
19
```

# Function Stack diagram



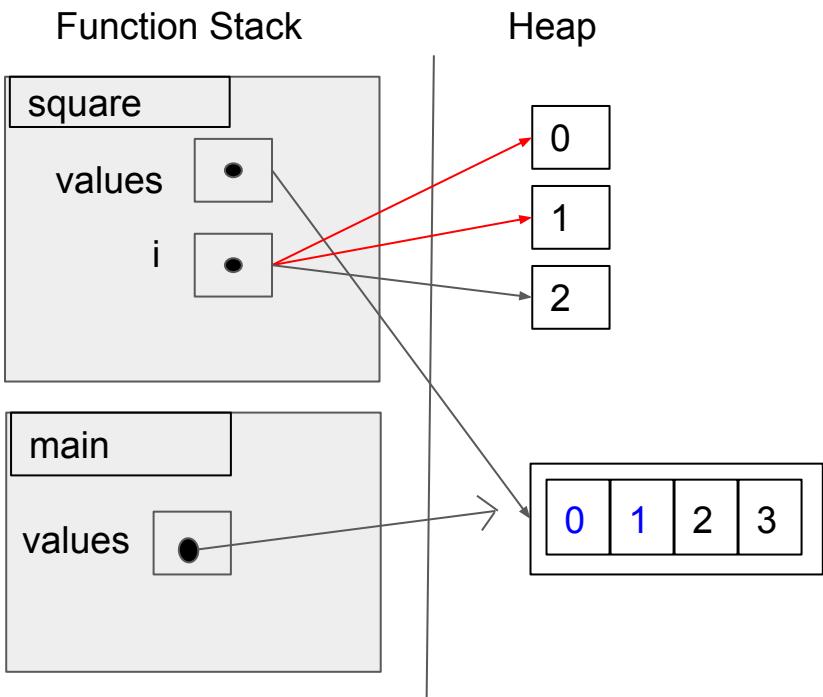
```
squareList.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squareList.py
1 """
2 Write a function that squares every item in a list.
3 Lists are an example of a mutable type.
4 What is the output of this program?
5 What does the function stack look like?
6 """
7
8 def square(values):
9     for i in range(len(values)):
10         values[i] = values[i] * values[i]
11
12 def main():
13     values = list(range(4))
14     print("Before:", values)
15     square(values)
16     print("After:", values)
17
18 main()
19
```

# Function Stack diagram



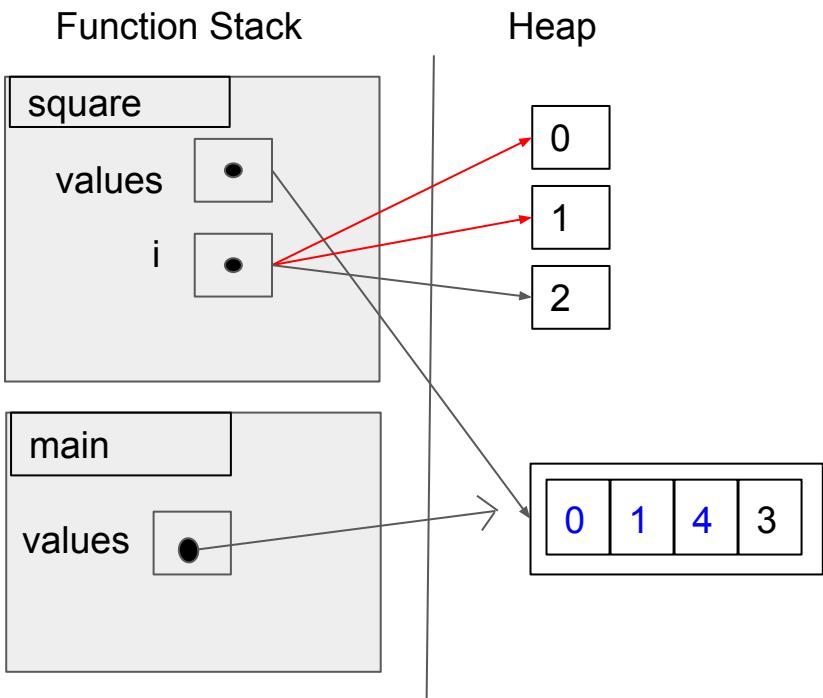
```
squareList.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squareList.py
1 """
2 Write a function that squares every item in a list.
3 Lists are an example of a mutable type.
4 What is the output of this program?
5 What does the function stack look like?
6 """
7
8 def square(values):
9     for i in range(len(values)):
10         values[i] = values[i] * values[i]
11
12 def main():
13     values = list(range(4))
14     print("Before:", values)
15     square(values)
16     print("After:", values)
17
18 main()
19
```

# Function Stack diagram



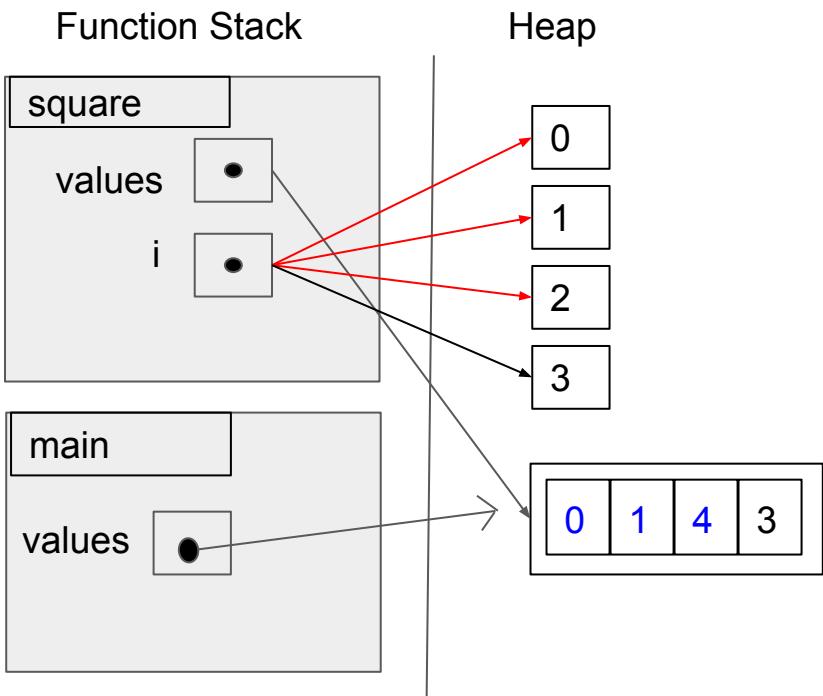
```
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py
squareList.py — ~/classes/cs21/f18/library.git/inclass/w05 — Atom
1 """
2 Write a function that squares every item in a list.
3 Lists are an example of a mutable type.
4 What is the output of this program?
5 What does the function stack look like?
6 """
7
8 def square(values):
9     for i in range(len(values)):
10         values[i] = values[i] * values[i]
11
12 def main():
13     values = list(range(4))
14     print("Before:", values)
15     square(values)
16     print("After:", values)
17
18 main()
19
```

# Function Stack diagram



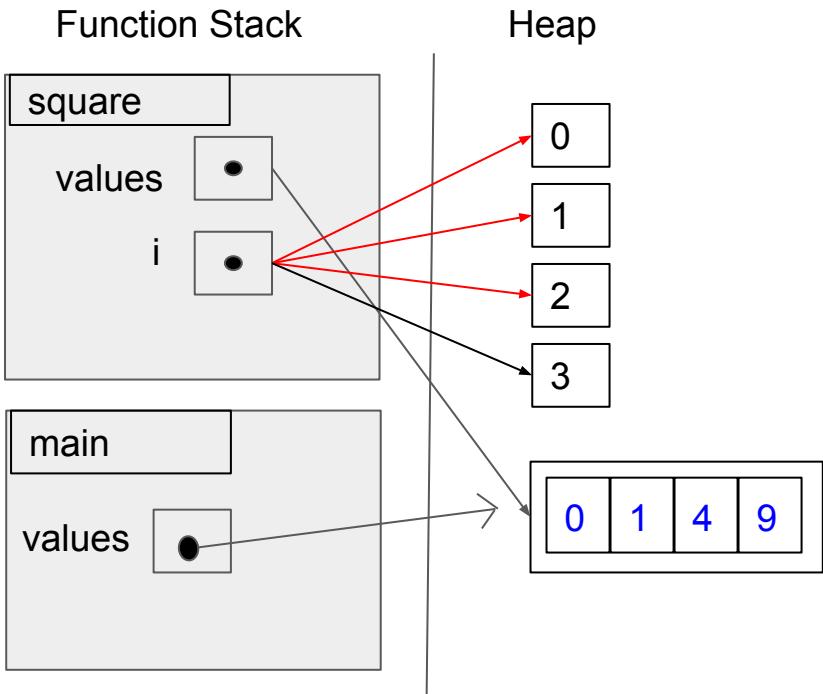
```
squareList.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squareList.py
1 """
2 Write a function that squares every item in a list.
3 Lists are an example of a mutable type.
4 What is the output of this program?
5 What does the function stack look like?
6 """
7
8 def square(values):
9     for i in range(len(values)):
10         values[i] = values[i] * values[i]
11
12 def main():
13     values = list(range(4))
14     print("Before:", values)
15     square(values)
16     print("After:", values)
17
18 main()
19
```

# Function Stack diagram



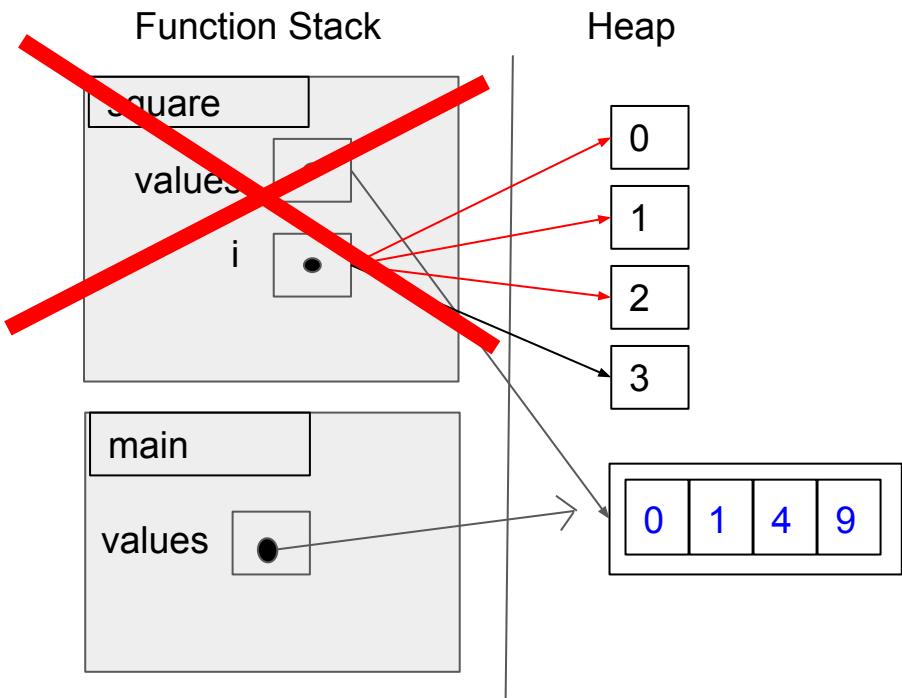
```
squareList.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squareList.py
1 """
2 Write a function that squares every item in a list.
3 Lists are an example of a mutable type.
4 What is the output of this program?
5 What does the function stack look like?
6 """
7
8 def square(values):
9     for i in range(len(values)):
10         values[i] = values[i] * values[i]
11
12 def main():
13     values = list(range(4))
14     print("Before:", values)
15     square(values)
16     print("After:", values)
17
18 main()
19
```

# Function Stack diagram



```
squareList.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
1 """
2 Write a function that squares every item in a list.
3 Lists are an example of a mutable type.
4 What is the output of this program?
5 What does the function stack look like?
6 """
7
8 def square(values):
9     for i in range(len(values)):
10         values[i] = values[i] * values[i]
11
12 def main():
13     values = list(range(4))
14     print("Before:", values)
15     square(values)
16     print("After:", values)
17
18 main()
19
```

# Function Stack diagram



squareList.py — ~/classes/cs21/f18/library.git/inclass/w05 — Atom

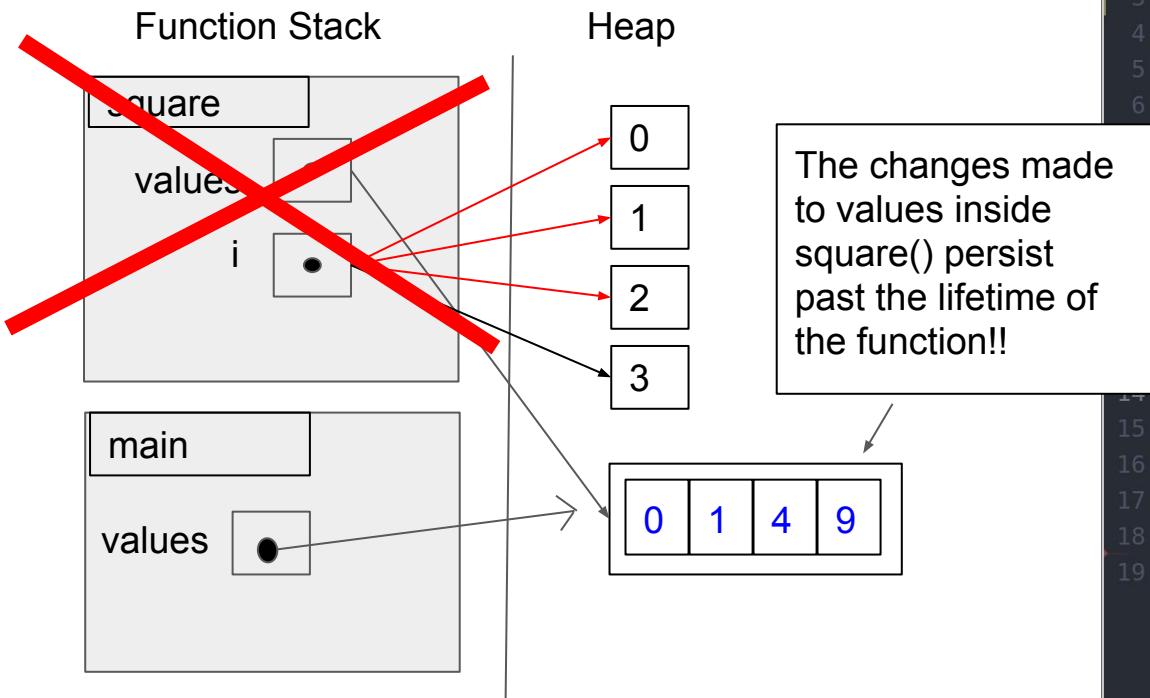
File Edit View Selection Find Packages Help

addVal.py askList.py sumList.py square.py squareList.py

```
1 """
2 Write a function that squares every item in a list.
3 Lists are an example of a mutable type.
4 What is the output of this program?
5 What does the function stack look like?
6 """
7
8 def square(values):
9     for i in range(len(values)):
10         values[i] = values[i] * values[i]
11
12 def main():
13     values = list(range(4))
14     print("Before:", values)
15     square(values)
16     print("After:", values)
17
18 main()
19
```

squareList.py 14:1 LF N UTF-8 Python

# Function Stack diagram



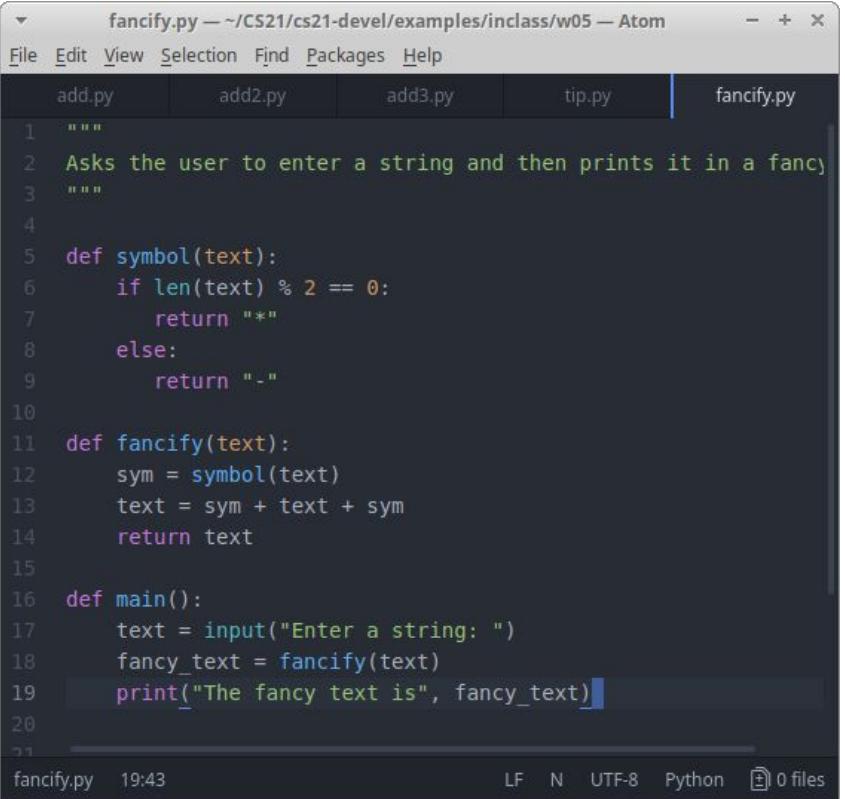


# fancify.py

strings are immutable types, so they work on the heap similarly to ints, floats, and booleans

```
almond[w05]$ python3 fancify.py  
Enter a string: hello  
The fancy text is -hello-
```

```
almond[w05]$ python3 fancify.py  
Enter a string: cats  
The fancy text is *cats*
```



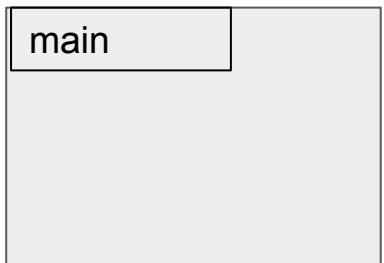
The screenshot shows the Atom code editor with the file 'fancify.py' open. The code defines a function 'fancify' that takes a string and returns it with a symmetrical border of either '-' or '\*' characters. It uses a helper function 'symbol' to determine the border character based on the string's length.

```
1  """  
2  Asks the user to enter a string and then prints it in a fancy  
3  """  
4  
5  def symbol(text):  
6      if len(text) % 2 == 0:  
7          return "*"  
8      else:  
9          return "-"  
10  
11 def fancify(text):  
12     sym = symbol(text)  
13     text = sym + text + sym  
14     return text  
15  
16 def main():  
17     text = input("Enter a string: ")  
18     fancy_text = fancify(text)  
19     print("The fancy text is", fancy_text)  
20  
21
```

The status bar at the bottom shows the file name 'fancify.py', the current time '19:43', and the encoding 'UTF-8'. There are also tabs for 'add.py', 'add2.py', 'add3.py', and 'tip.py'.

# Function stack diagram

Function Stack



Heap

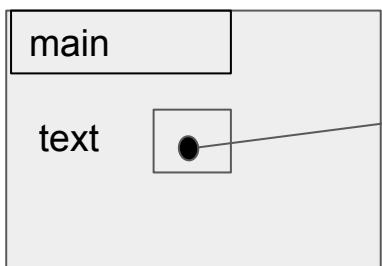
A screenshot of a code editor window titled "fancify.py — ~/CS21/cs21-devel/examples/inclass/w05 — Atom". The window shows a Python script with several functions:

```
1 """
2 Asks the user to enter a string and then prints it in a fancy
3 """
4
5 def symbol(text):
6     if len(text) % 2 == 0:
7         return "*"
8     else:
9         return "-"
10
11 def fancify(text):
12     sym = symbol(text)
13     text = sym + text + sym
14     return text
15
16 def main():
17     text = input("Enter a string: ")
18     fancy_text = fancify(text)
19     print("The fancy text is", fancy_text)
20
21
```

The line "def main():" is highlighted with a red rectangle.

# Function stack diagram

Function Stack



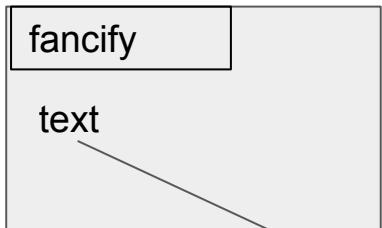
Heap

A screenshot of a code editor window titled "fancify.py — ~/CS21/cs21-devel/examples/inclass/w05 — Atom". The window shows a Python script with several functions: `symbol`, `fancify`, and `main`. A red rectangle highlights the line `text = input("Enter a string: ")` in the `main` function. A callout box to the right of the code states "Assume user enters ‘hello’".

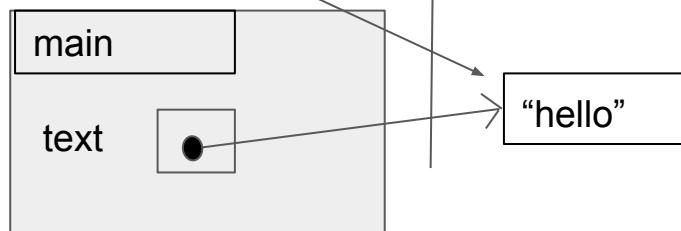
```
1 """
2 Asks the user to enter a string and then prints it in a fancy
3 """
4
5 def symbol(text):
6     if len(text) % 2 == 0:
7         return "*"
8     else:
9         return "-"
10
11 def fancify(text):
12     sym = symbol(text)
13     text = sym + text + sym
14     return text
15
16 if __name__ == "__main__":
17     text = input("Enter a string: ")
18     fancy_text = fancify(text)
19     print("The fancy text is", fancy_text)
```

# Function stack diagram

Function Stack



Heap

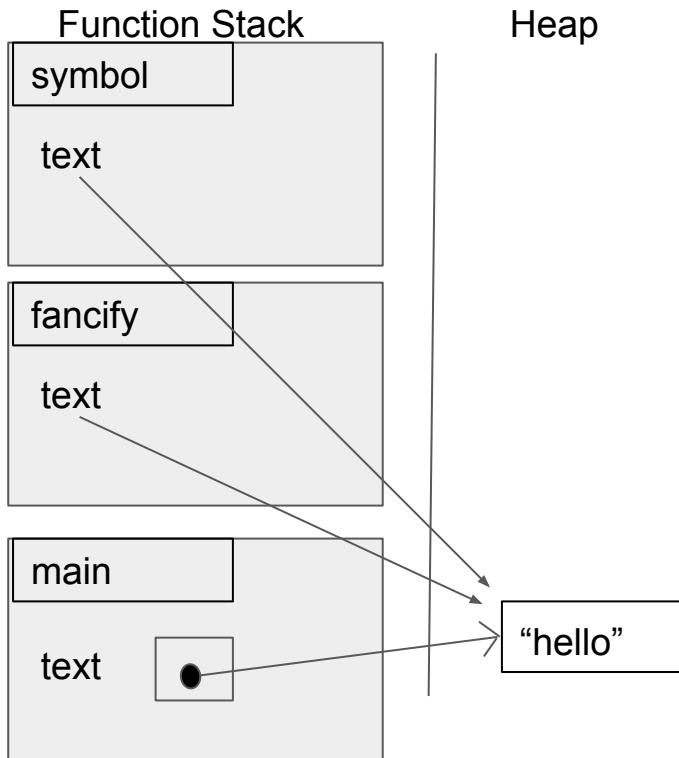


```
fancify.py -- ~/CS21/cs21-devel/examples/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
add.py add2.py add3.py tip.py fancify.py

1 """
2 Asks the user to enter a string and then prints it in a fancy
3 """
4
5 def symbol(text):
6     if len(text) % 2 == 0:
7         return "*"
8     else:
9         return "-"
10
11 def fancify(text):
12     sym = symbol(text)
13     text = sym + text + sym
14     return text
15
16 def main():
17     text = input("Enter a string: ")
18     fancy_text = fancify(text)
19     print("The fancy text is", fancy_text)
20
21

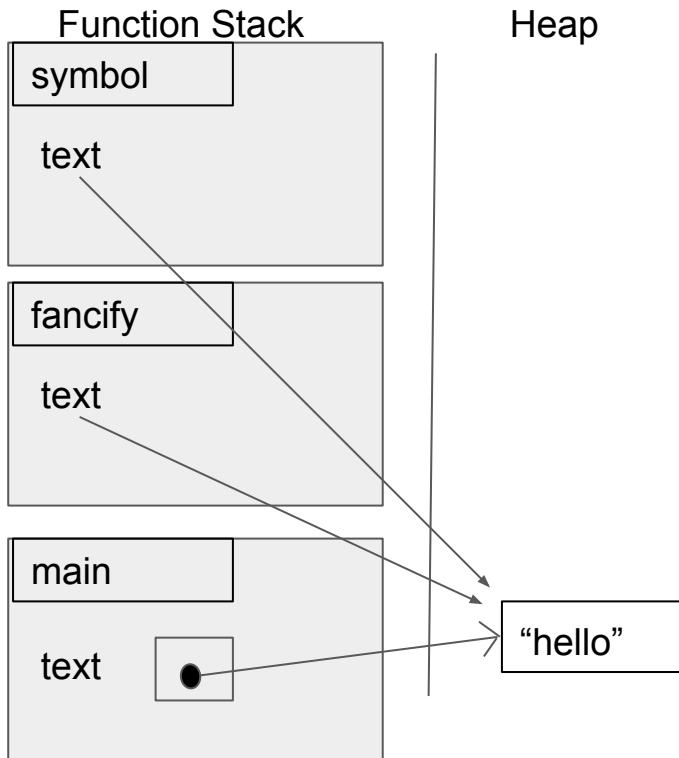
fancify.py 19:43 LF N UTF-8 Python 0 files
```

# Function stack diagram



```
fancify.py — ~/CS21/cs21-devel/examples/inclass/w05 — Atom
File Edit View Selection Find Packages Help
add.py add2.py add3.py tip.py fancify.py
1 """
2 Asks the user to enter a string and then prints it in a fancy
3 """
4
5 def symbol(text):
6     if len(text) < 2:
7         return "*"
8     else:
9         return "-"
10
11 def fancify(text):
12     sym = symbol(text)
13     text = sym + text + sym
14     return text
15
16 def main():
17     text = input("Enter a string: ")
18     fancy_text = fancify(text)
19     print("The fancy text is", fancy_text)
20
21
fancify.py 19:43 LF N UTF-8 Python 0 files
```

# Function stack diagram



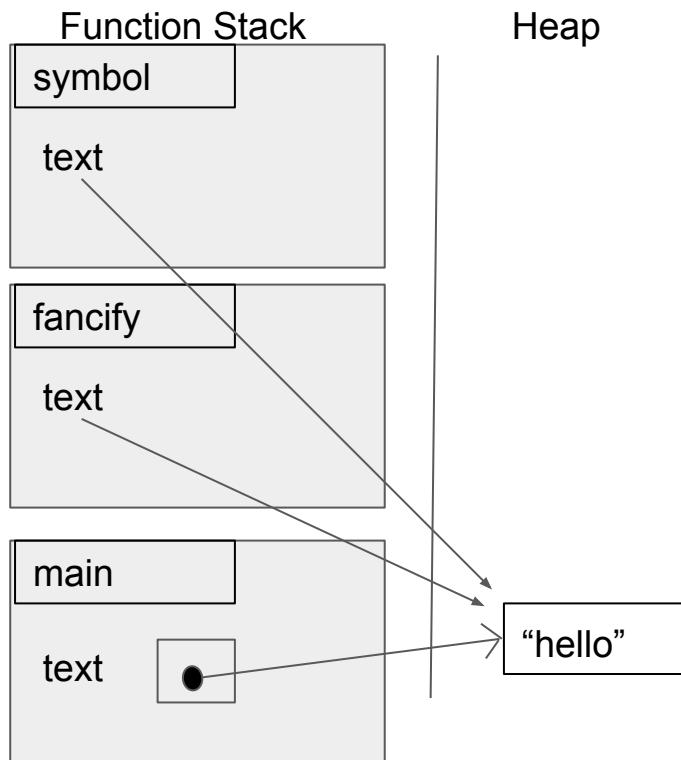
fancify.py — ~/CS21/cs21-devel/examples/inclass/w05 — Atom

```
File Edit View Selection Find Packages Help  
add.py add2.py add3.py tip.py fancify.py  
1 """  
2 Asks the user to enter a string and then prints it in a fancy  
3 """  
4  
5 def symbol(text):  
6     if len(text) % 2 == 0:  
7         return ""  
8     else:  
9         return "--"  
10  
11 def fancify(text):  
12     sym = symbol(text)  
13     text = sym + text + sym  
14     return text  
15  
16 def main():  
17     text = input("Enter a string: ")  
18     fancy_text = fancify(text)  
19     print("The fancy text is", fancy_text)  
20  
21
```

condition is False

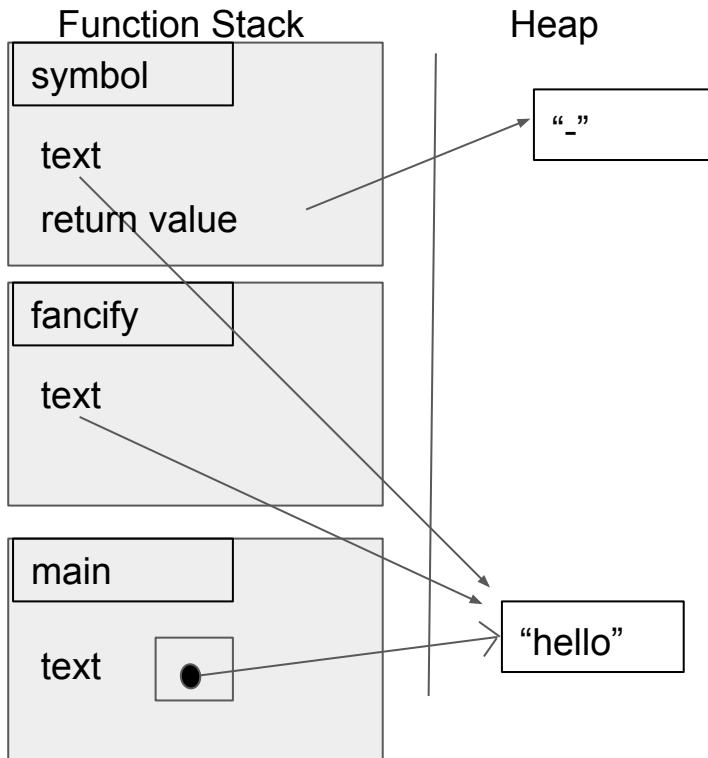
fancify.py 19:43 LF N UTF-8 Python 0 files

# Function stack diagram



```
fancify.py — ~/CS21/cs21-devel/examples/inclass/w05 — Atom
File Edit View Selection Find Packages Help
add.py add2.py add3.py tip.py fancify.py
1 """
2 Asks the user to enter a string and then prints it in a fancy
3 """
4
5 def symbol(text):
6     if len(text) % 2 == 0:
7         return "*"
8     else:
9         return " "
10
11 def fancify(text):
12     sym = symbol(text)
13     text = sym + text + sym
14     return text
15
16 def main():
17     text = input("Enter a string: ")
18     fancy_text = fancify(text)
19     print("The fancy text is", fancy_text)
20
21
fancify.py 19:43 LF N UTF-8 Python 0 files
```

# Function stack diagram

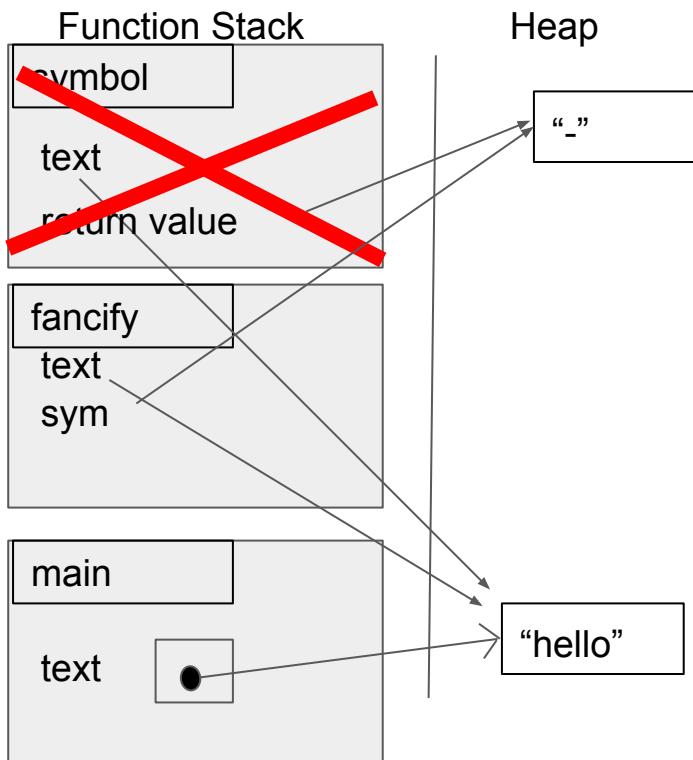


fancify.py — ~/CS21/cs21-devel/examples/inclass/w05 — Atom

```
File Edit View Selection Find Packages Help  
add.py add2.py add3.py tip.py fancify.py  
1 """  
2 Asks the user to enter a string and then prints it in a fancy  
3 """  
4  
5 def symbol(text):  
6     if len(text) % 2 == 0:  
7         return "*"  
8     else:  
9         return "-"  
10  
11 def fancify(text):  
12     sym = symbol(text)  
13     text = sym + text + sym  
14     return text  
15  
16 def main():  
17     text = input("Enter a string: ")  
18     fancy_text = fancify(text)  
19     print("The fancy text is", fancy_text)  
20  
21
```

fancify.py 19:43 LF N UTF-8 Python 0 files

# Function stack diagram



fancify.py — ~/CS21/cs21-devel/examples/inclass/w05 — Atom

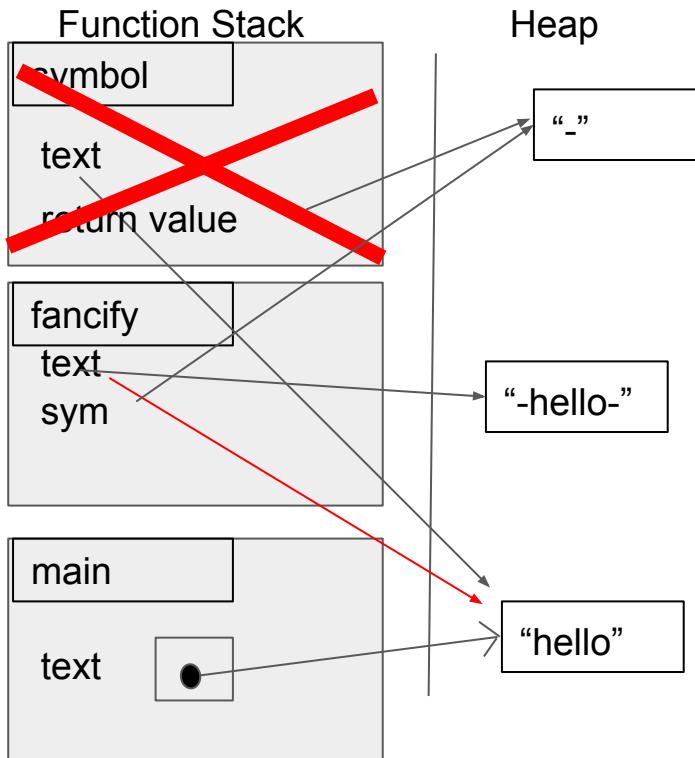
File Edit View Selection Find Packages Help

add.py add2.py add3.py tip.py fancify.py

```
1 """
2 Asks the user to enter a string and then prints it in a fancy
3 """
4
5 def symbol(text):
6     if len(text) % 2 == 0:
7         return "*"
8     else:
9         return "-"
10
11 def fancify(text):
12     sym = symbol(text)
13     text = Sym + text + Sym
14     return text
15
16 def main():
17     text = input("Enter a string: ")
18     fancy_text = fancify(text)
19     print("The fancy text is", fancy_text)
20
21
```

fancify.py 19:43 LF N UTF-8 Python 0 files

# Function stack diagram



fancify.py — ~/CS21/cs21-devel/examples/inclass/w05 — Atom

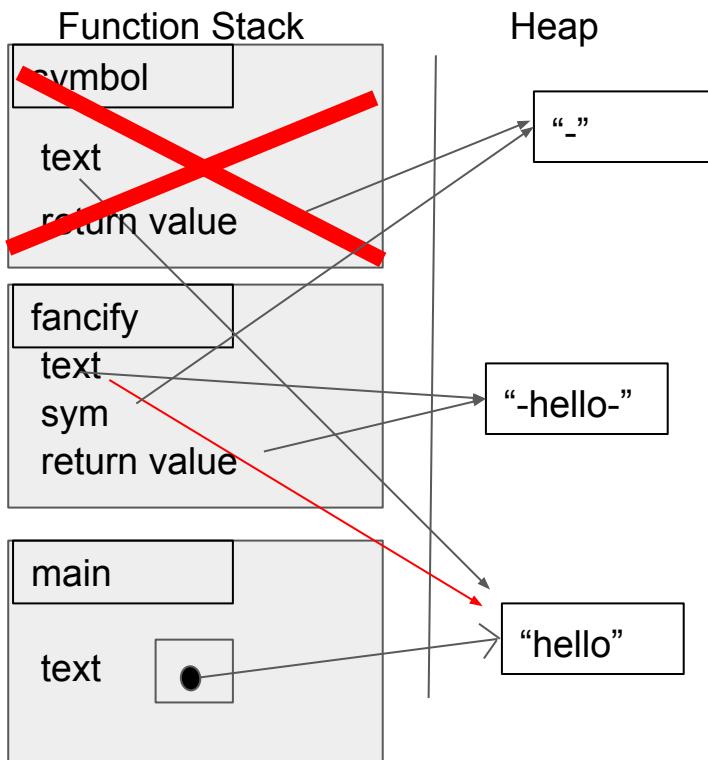
File Edit View Selection Find Packages Help

add.py add2.py add3.py tip.py fancify.py

```
1 """
2 Asks the user to enter a string and then prints it in a fancy
3 """
4
5 def symbol(text):
6     if len(text) % 2 == 0:
7         return "*"
8     else:
9         return "-"
10
11 def fancify(text):
12     sym = symbol(text)
13     text = sym + text + sym
14     return text
15
16 def main():
17     text = input("Enter a string: ")
18     fancy_text = fancify(text)
19     print("The fancy text is", fancy_text)
20
21
```

fancify.py 19:43 LF N UTF-8 Python 0 files

# Function stack diagram



fancify.py — ~/CS21/cs21-devel/examples/inclass/w05 — Atom

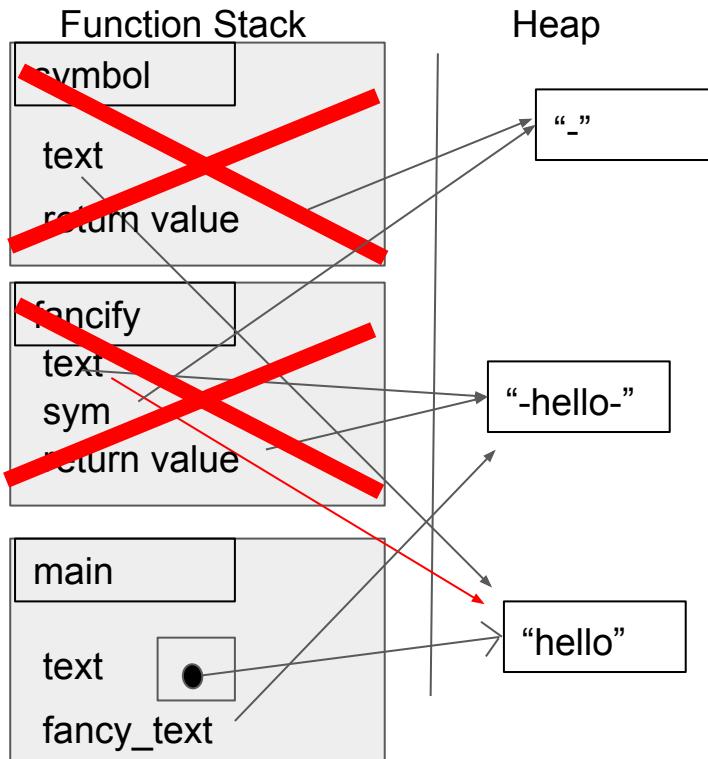
File Edit View Selection Find Packages Help

add.py add2.py add3.py tip.py fancify.py

```
1 """
2 Asks the user to enter a string and then prints it in a fancy
3 """
4
5 def symbol(text):
6     if len(text) % 2 == 0:
7         return "*"
8     else:
9         return "-"
10
11 def fancify(text):
12     sym = symbol(text)
13     text = sym + text + sym
14     return text
15
16 def main():
17     text = input("Enter a string: ")
18     fancy_text = fancify(text)
19     print("The fancy text is", fancy_text)
20
21
```

fancify.py 19:43 LF N UTF-8 Python 0 files

# Function stack diagram



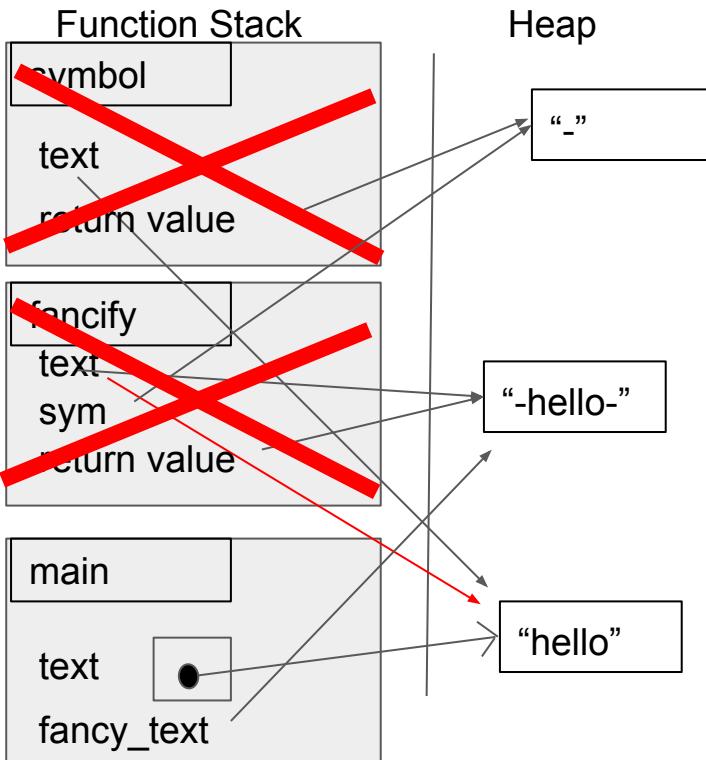
fancify.py — ~/CS21/cs21-devel/examples/inclass/w05 — Atom

File Edit View Selection Find Packages Help

```
1 """
2 Asks the user to enter a string and then prints it in a fancy
3 """
4
5 def symbol(text):
6     if len(text) % 2 == 0:
7         return "*"
8     else:
9         return "-"
10
11 def fancify(text):
12     sym = symbol(text)
13     text = sym + text + sym
14     return text
15
16 def main():
17     text = input("Enter a string. ")
18     fancy_text = fancify(text)
19     print("The fancy text is", fancy_text)
```

fancify.py 19:43 LF N UTF-8 Python 0 files

# Function stack diagram



fancify.py — ~/CS21/cs21-devel/examples/inclass/w05 — Atom

File Edit View Selection Find Packages Help

add.py add2.py add3.py tip.py fancify.py

```
1 """
2 Asks the user to enter a string and then prints it in a fancy
3 """
4
5 def symbol(text):
6     if len(text) % 2 == 0:
7         return "*"
8     else:
9         return "_"
10
11 def fancify(text):
12     sym = symbol(text)
13     text = sym + text + sym
14     return text
15
16 def main():
17     text = input("Enter a string: ")
18     fancy_text = fancify(text)
19     print("The fancy text is", fancy_text)
20
21
```

fancify.py 19:43 LF N UTF-8 Python 0 files