

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

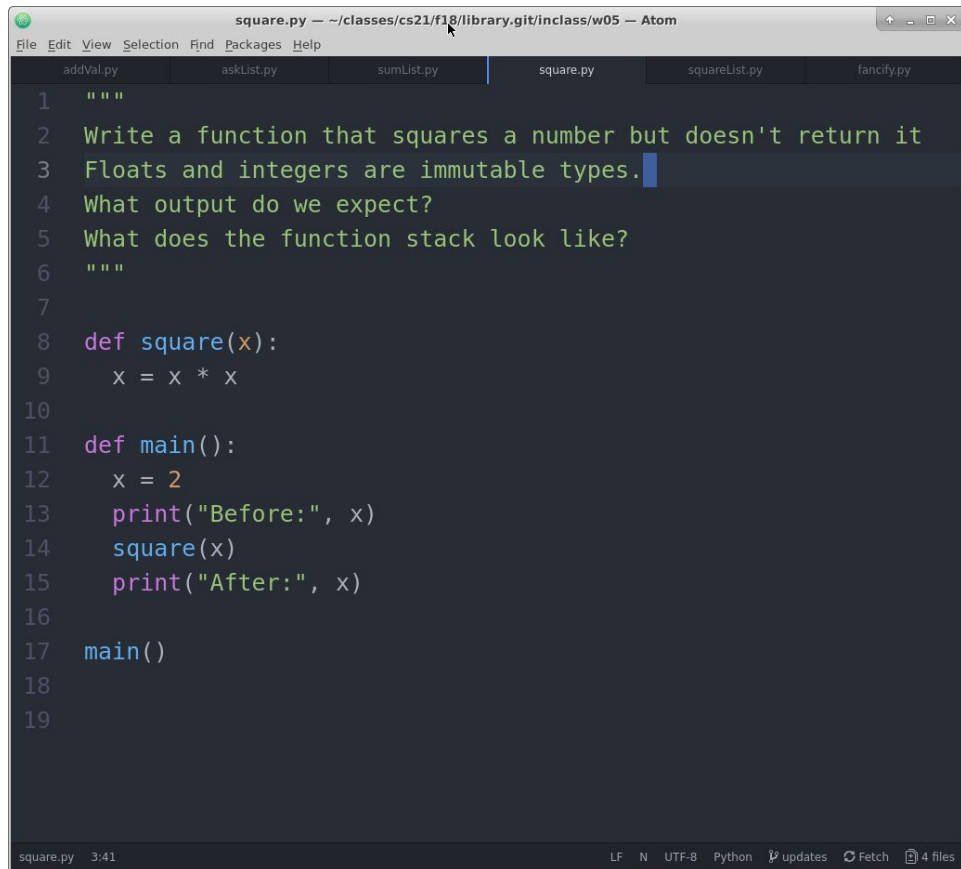
If a variable is **mutable** and you change its value inside a function, the change is permanent

Example: square.py

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

Before: 2

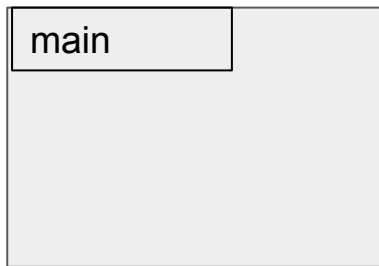
After: 2



```
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 squareList.py fancify.py
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
square.py 3:41 LF N UTF-8 Python updates Fetch 4 files
```

Function stack diagram

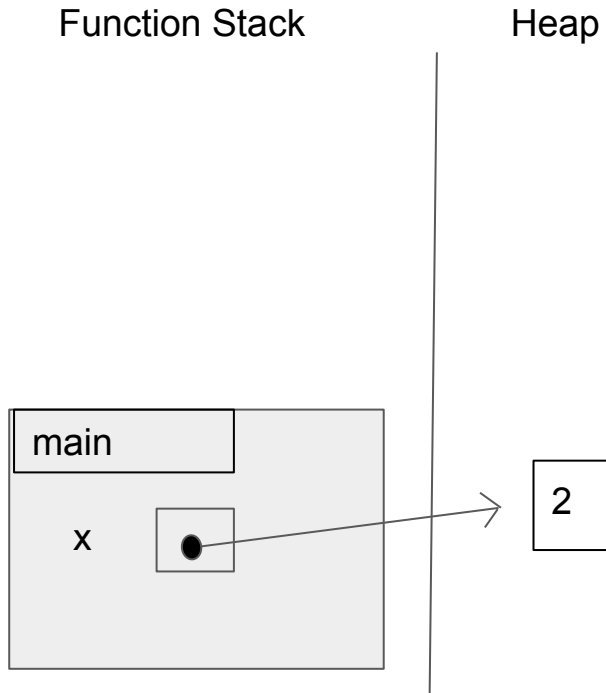
Function Stack



Heap

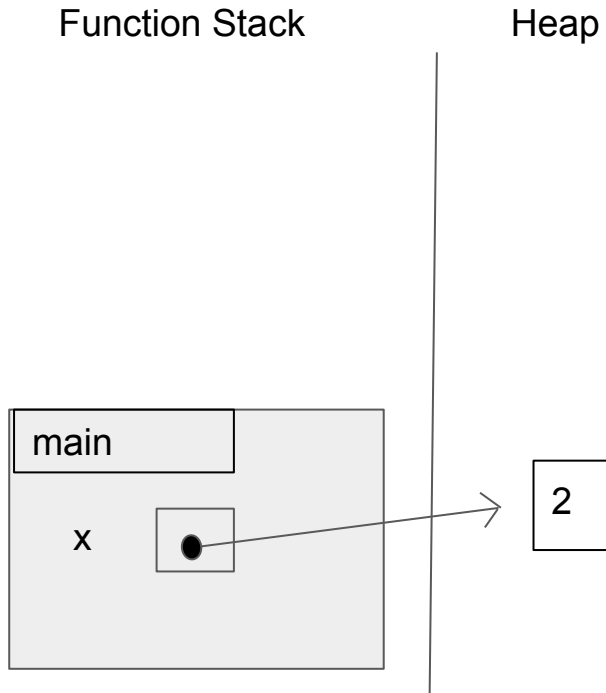
```
square.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squa
1 """
2 Write a function that squares a number but do
3 Floats and integers are immutable types.
4 What output do we expect?
5 What does the function 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



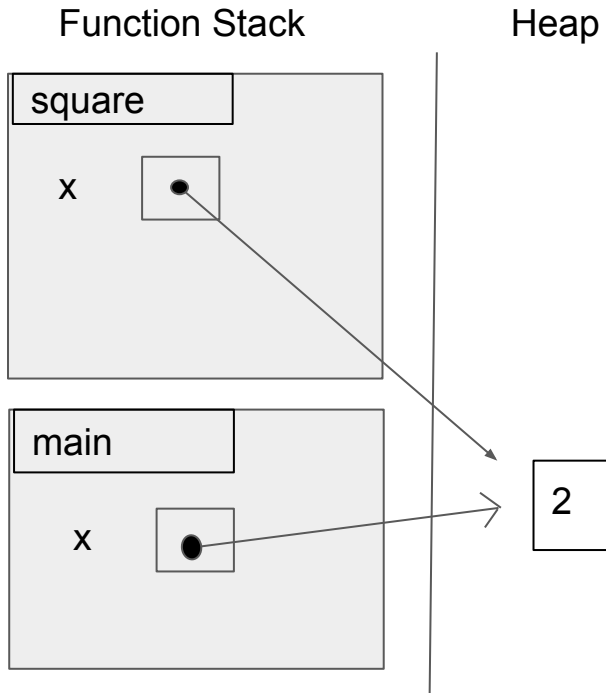
```
square.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squa
1 """
2 Write a function that squares a number but do
3 Floats and integers are immutable types.
4 What output do we expect?
5 What does the function 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



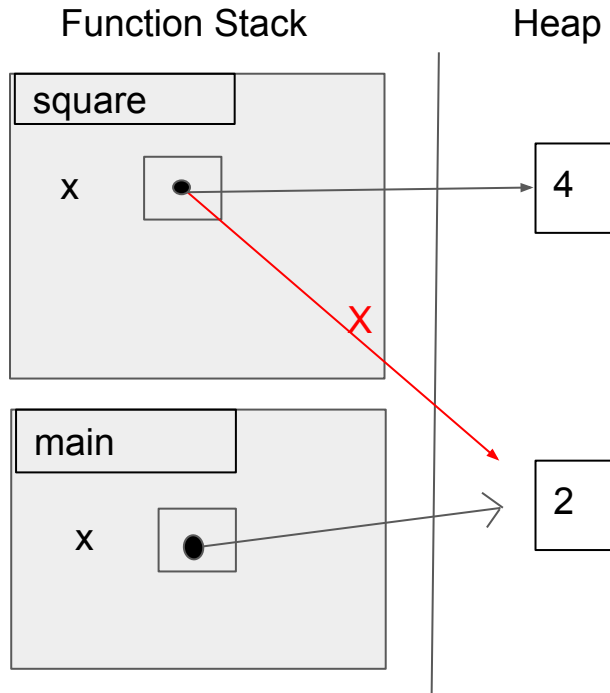
```
square.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squa
1 """
2 Write a function that squares a number but do
3 Floats and integers are immutable types.
4 What output do we expect?
5 What does the function 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 squa
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

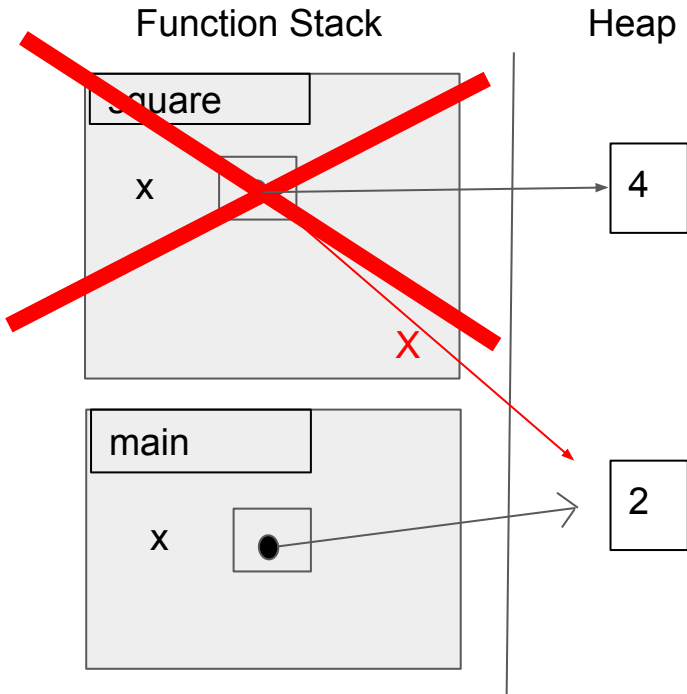


```
square.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squa

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

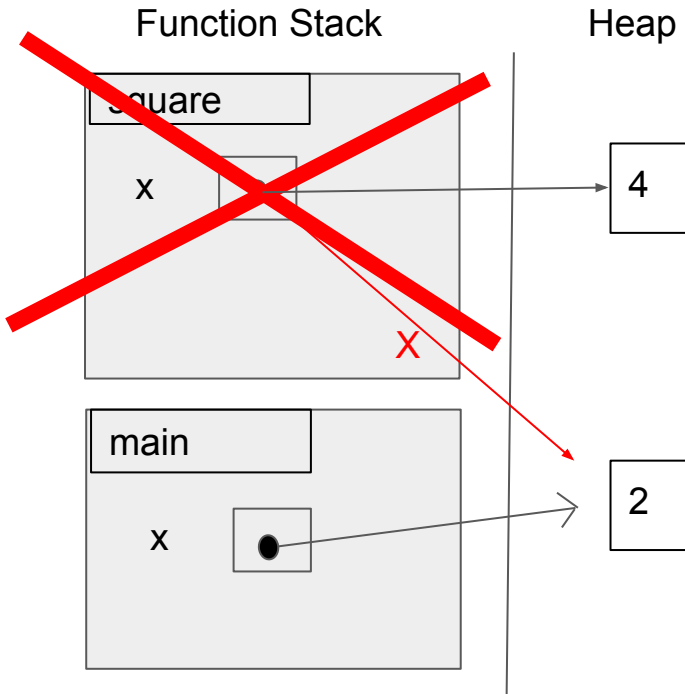


```
square.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squa

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 squa

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

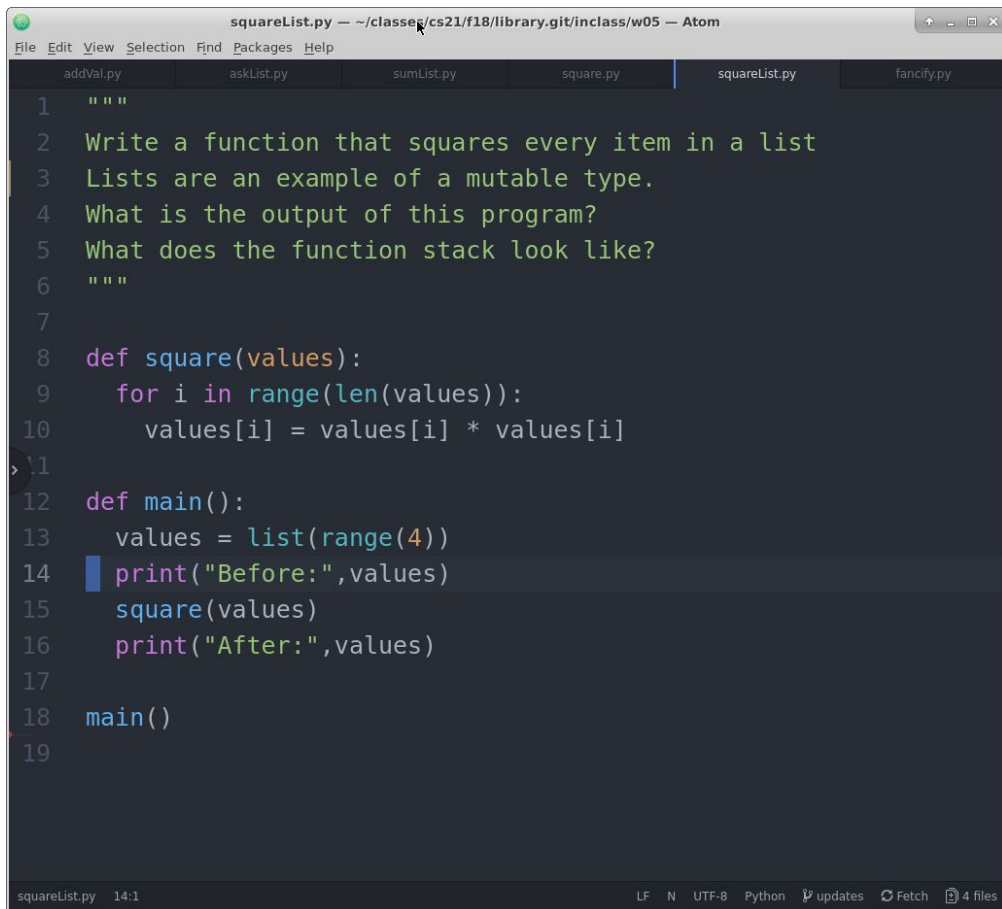
square.py 3:41 LF N UTF-8
```


Example: squareList.py

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

Before: [0, 1, 2, 3]

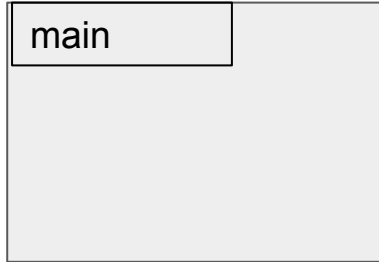
After: [0, 1, 4, 9]



```
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 updates Fetch 4 files
```

Function Stack diagram

Function Stack

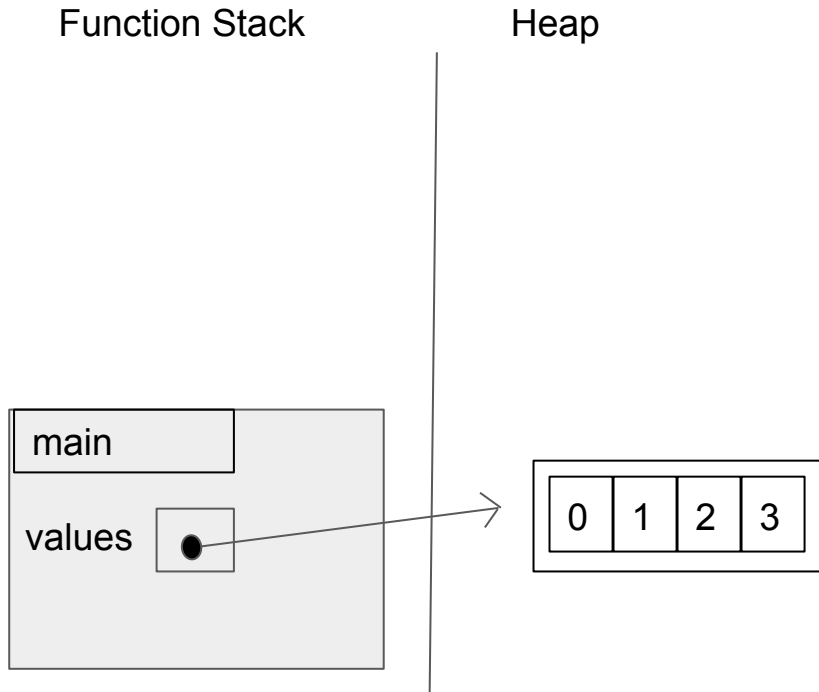


Heap

```
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.p
1 """
2 Write a function that squares every item in a li
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
```

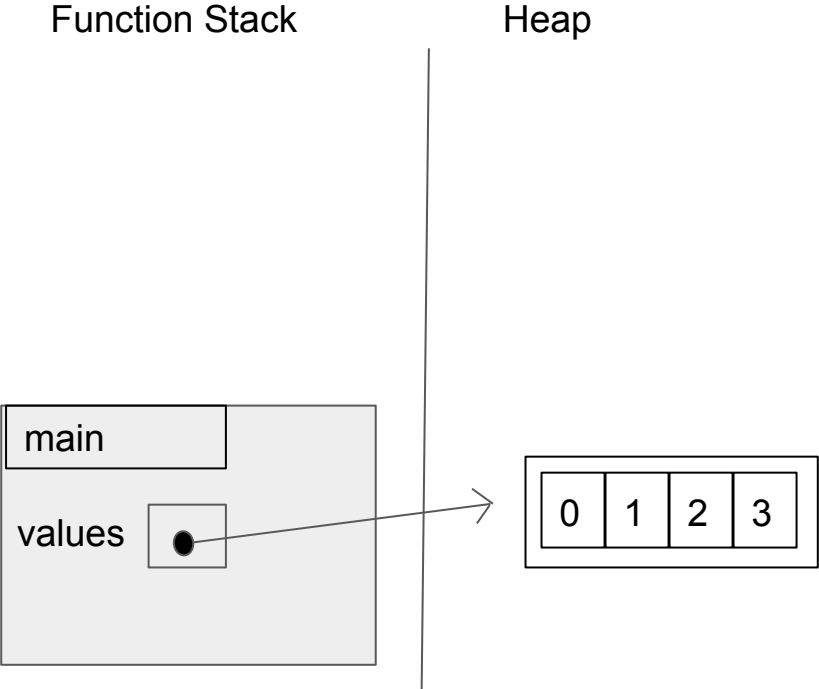
squareList.py 14:1 LF N UTF-8 Python

Function Stack diagram



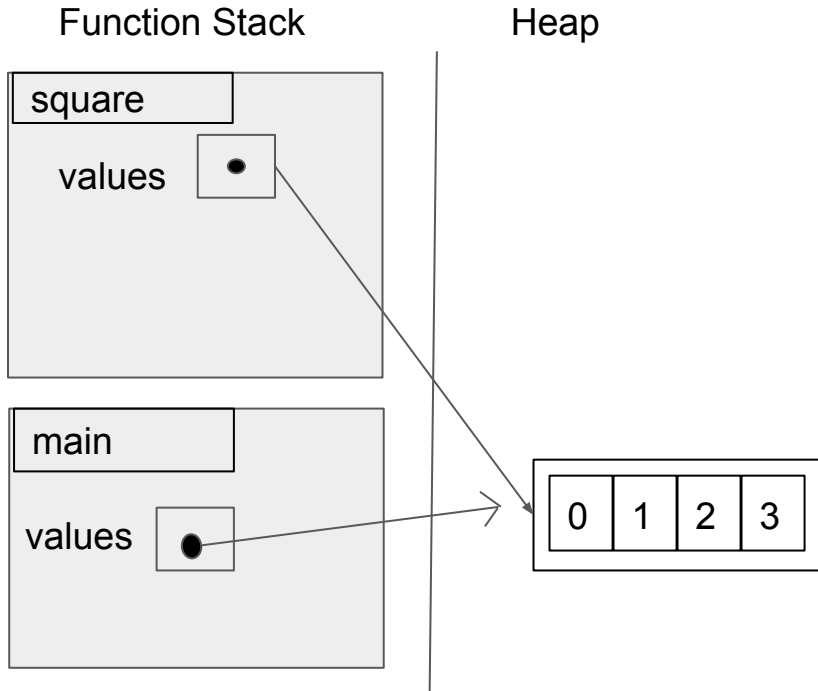
```
squareList.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squareList.py
1 """
2 Write a function that squares every item in a li
3 Lists are an example of a mutable type.
4 What is the output of this program?
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



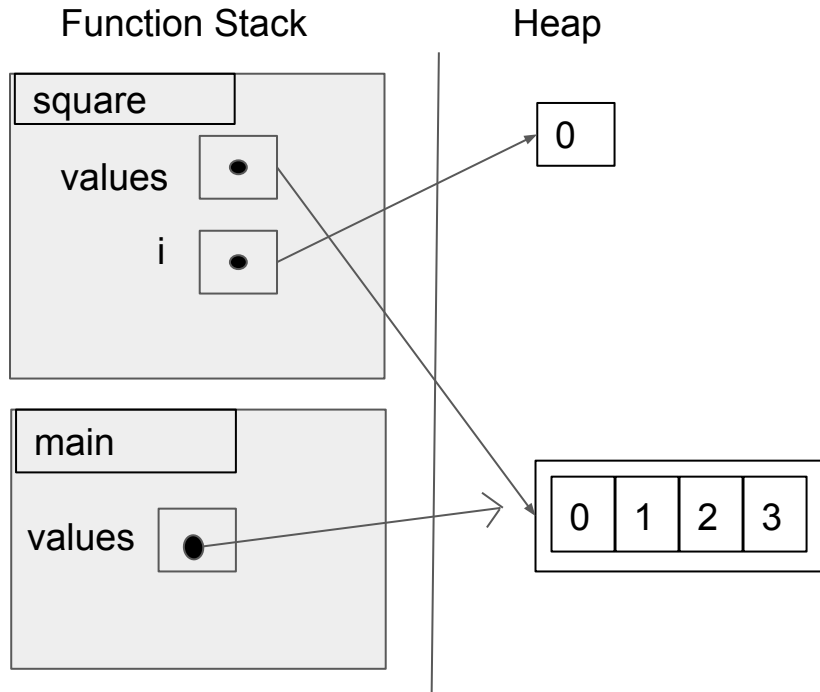
```
squareList.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squareList.py
1 """
2 Write a function that squares every item in a list.
3 Lists are an example of a mutable type.
4 What is the output of this 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



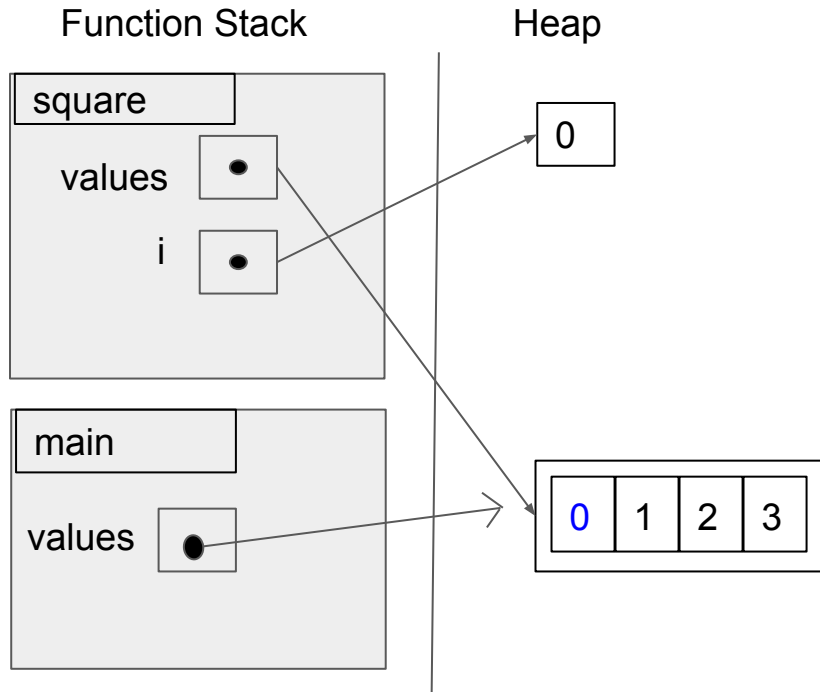
```
squareList.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squareList.py
1 """
2 Write a function that squares every item in a list.
3 Lists are an example of a mutable type.
4 What is the output of this 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
squareList.py 14:1 LF N UTF-8 Python
```

Function Stack diagram



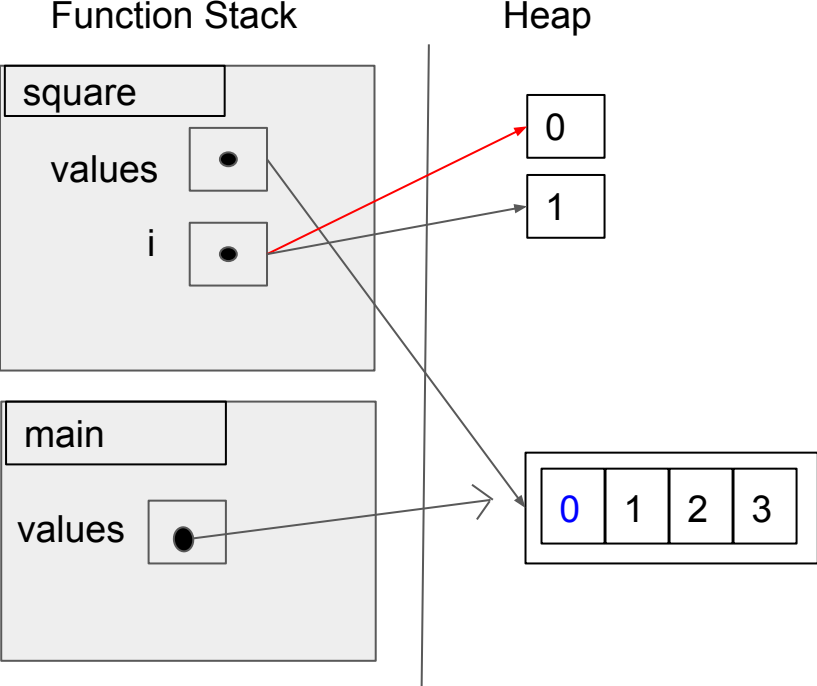
```
squareList.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squareList.py
1 """
2 Write a function that squares every item in a list.
3 Lists are an example of a mutable type.
4 What is the output of this 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
squareList.py 14:1 LF N UTF-8 Python
```


Function Stack diagram



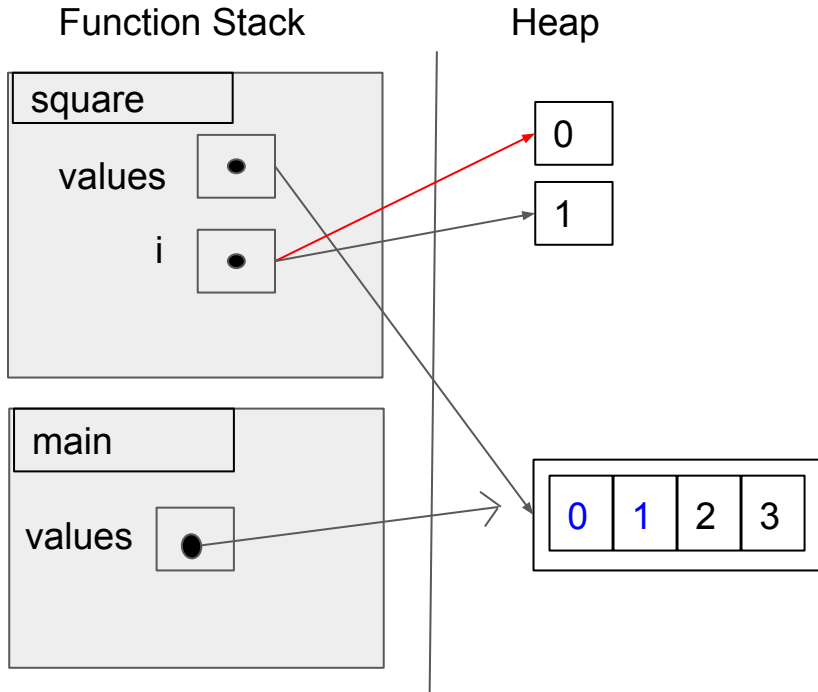
```
squareList.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squareList.py
1 """
2 Write a function that squares every item in a list.
3 Lists are an example of a mutable type.
4 What is the output of this 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
squareList.py 14:1 LF N UTF-8 Python
```

Function Stack diagram



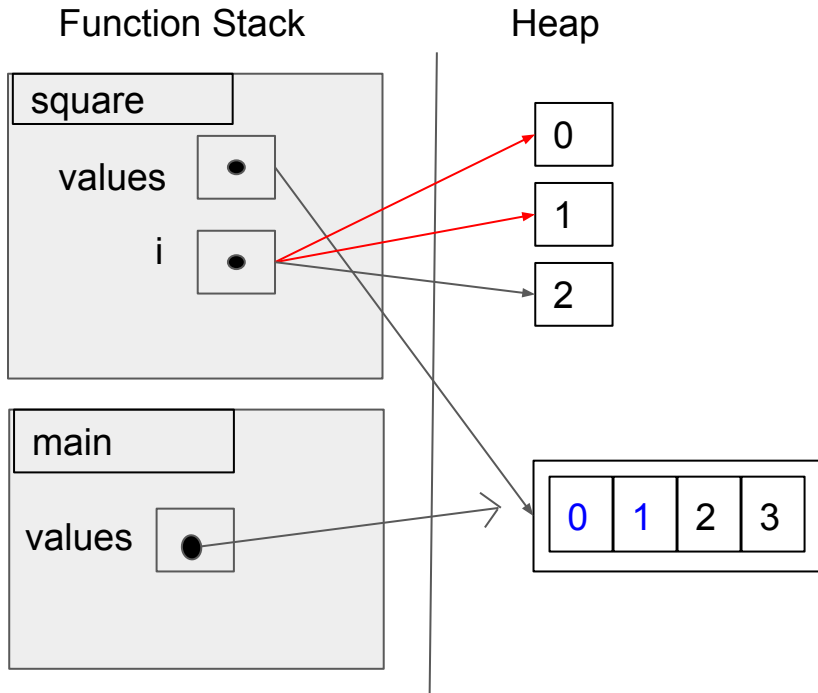
```
squareList.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squareList.py
1 """
2 Write a function that squares every item in a list.
3 Lists are an example of a mutable type.
4 What is the output of this 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
squareList.py 14:1 LF N UTF-8 Python
```

Function Stack diagram



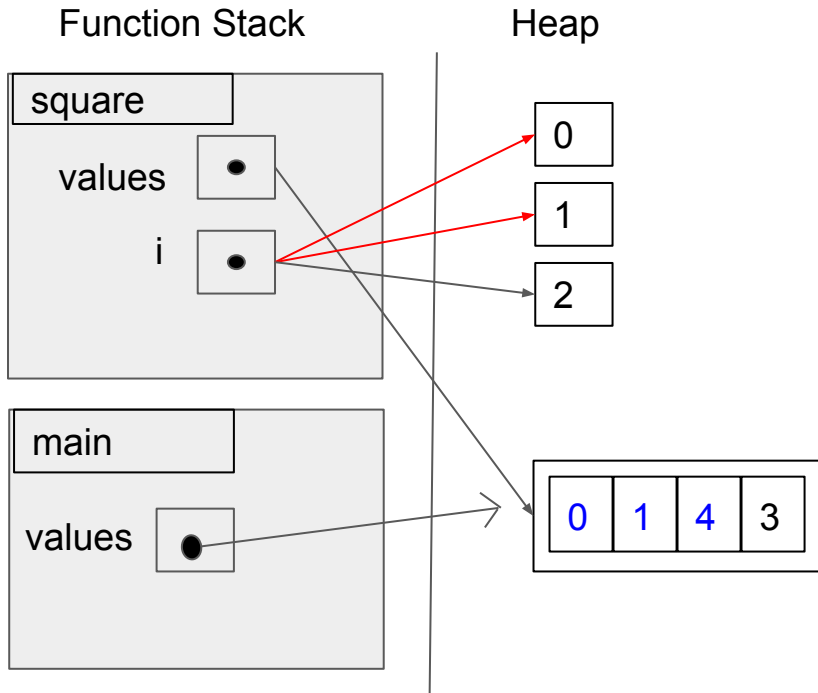
```
squareList.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squareList.py
1 """
2 Write a function that squares every item in a li
3 Lists are an example of a mutable type.
4 What is the output of this program?
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



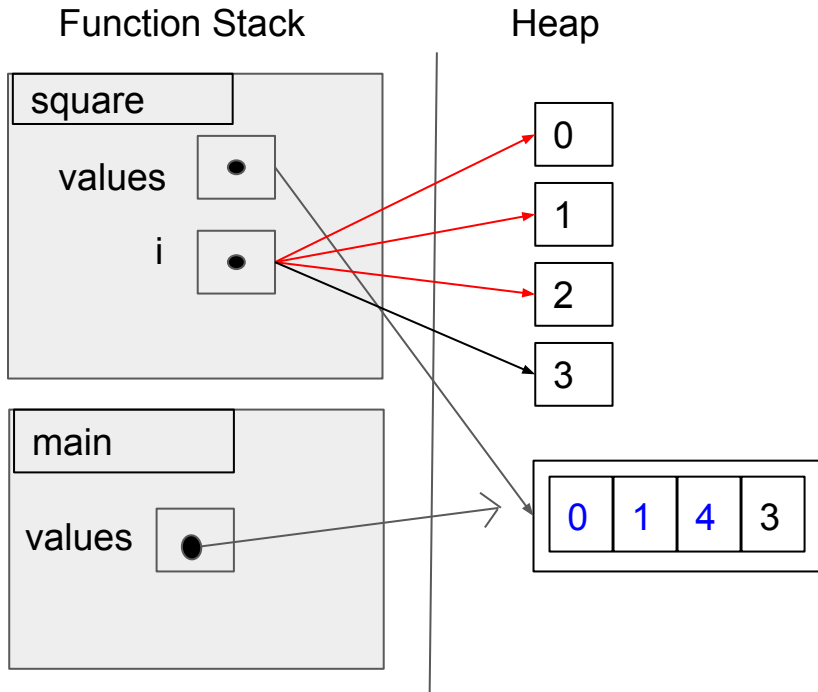
```
squareList.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squareList.py
1 """
2 Write a function that squares every item in a li
3 Lists are an example of a mutable type.
4 What is the output of this program?
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



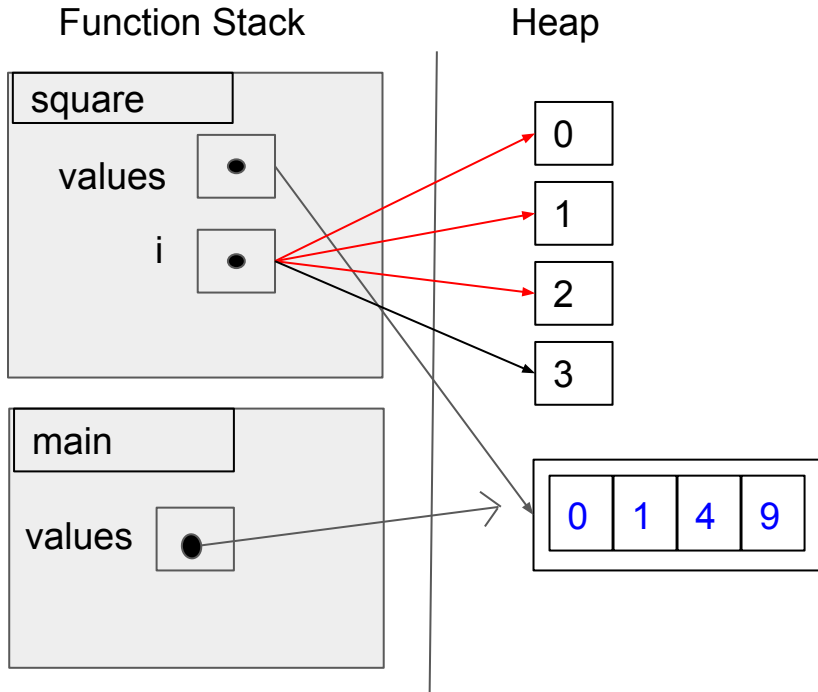
```
squareList.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squareList.py
1 """
2 Write a function that squares every item in a list.
3 Lists are an example of a mutable type.
4 What is the output of this 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
squareList.py 14:1 LF N UTF-8 Python
```

Function Stack diagram



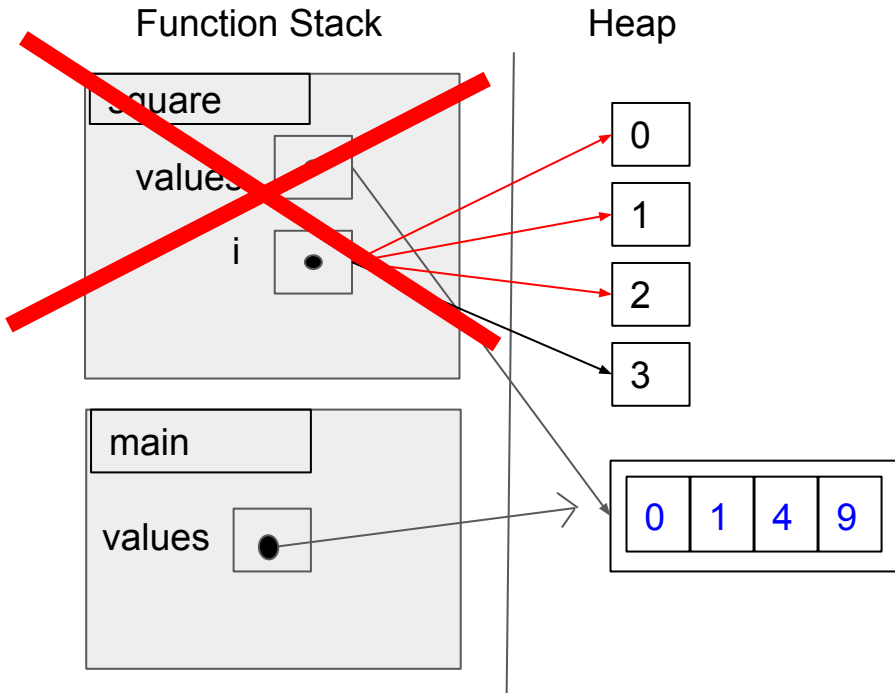
```
squareList.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squareList.py
1 """
2 Write a function that squares every item in a list.
3 Lists are an example of a mutable type.
4 What is the output of this 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
squareList.py 14:1 LF N UTF-8 Python
```

Function Stack diagram



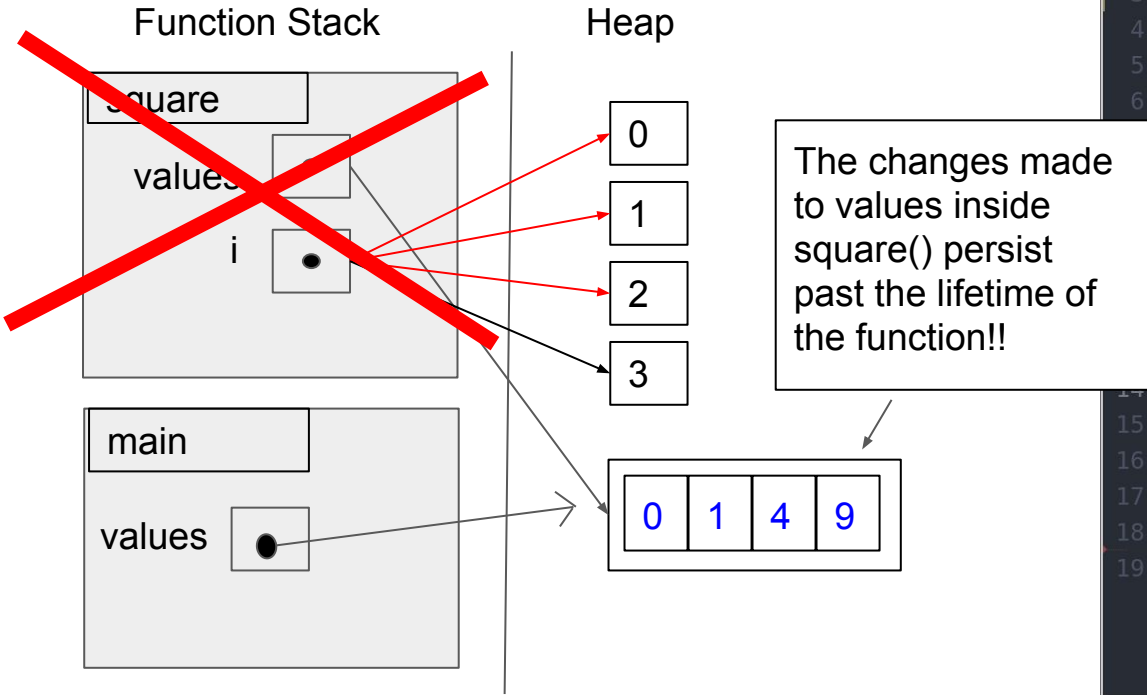
```
squareList.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squareList.py
1 """
2 Write a function that squares every item in a li
3 Lists are an example of a mutable type.
4 What is the output of this program?
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
squareList.py 14:1 LF N UTF-8 Python
```


Function Stack diagram



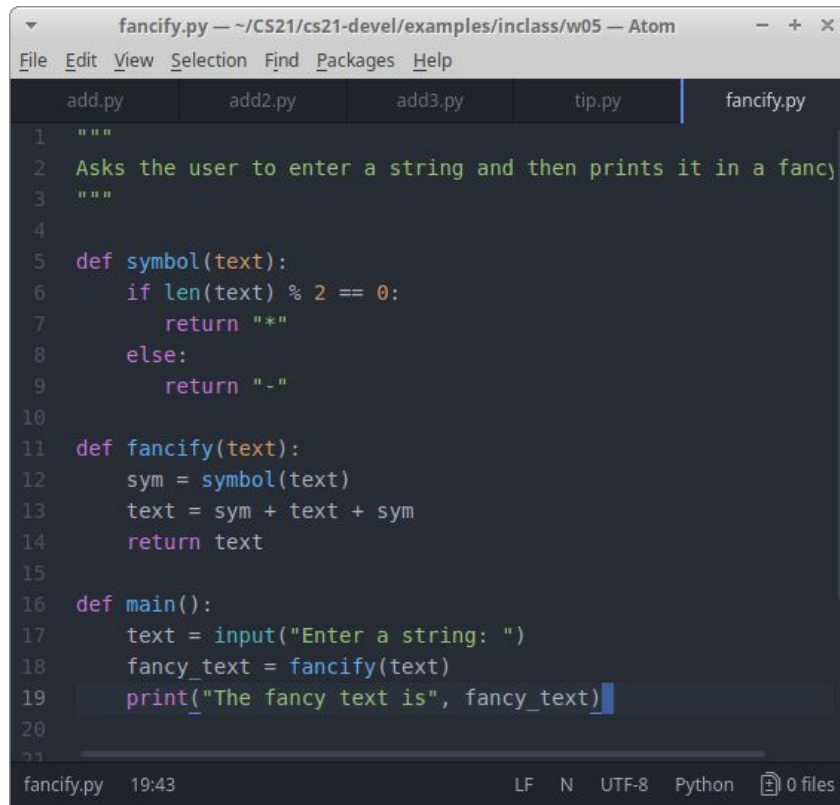
```
squareList.py -- ~/classes/cs21/f18/library.git/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
addVal.py askList.py sumList.py square.py squareList.py
1 """
2 Write a function that squares every item in a list.
3 Lists are an example of a mutable type.
4 What is the output of this 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
squareList.py 14:1 LF N UTF-8 Python
```


fancyfy.py

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

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

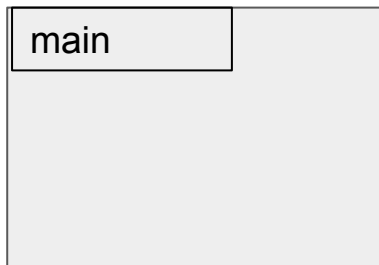
```
almond[w05]$ python3 fancyfy.py
Enter a string: cats
The fancy text is *cats*
```



```
fancyfy.py -- ~/CS21/cs21-devel/examples/inclass/w05 — Atom
File Edit View Selection Find Packages Help
add.py add2.py add3.py tip.py fancyfy.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 fancyfy(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 = fancyfy(text)
19     print("The fancy text is", fancy_text)
20
21
fancyfy.py 19:43 LF N UTF-8 Python 0 files
```

Function stack diagram

Function Stack



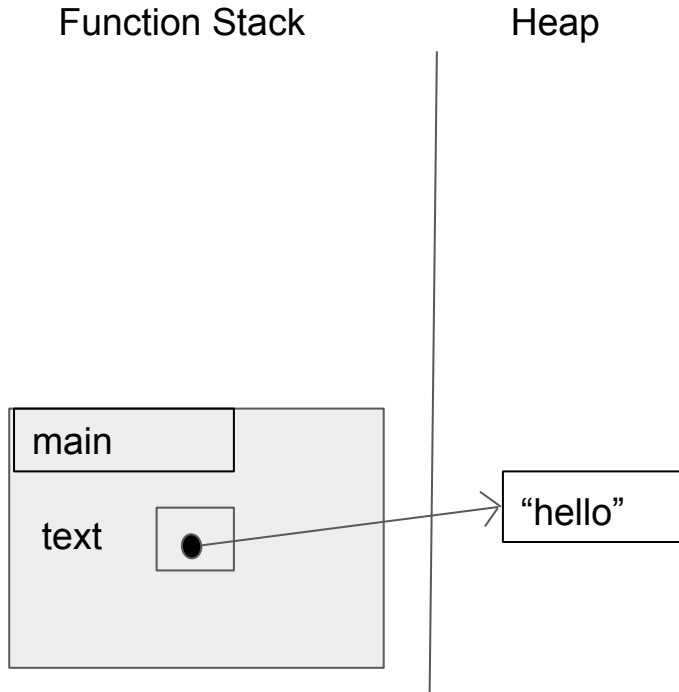
Heap

A screenshot of a code editor window titled "fancy.py". The code is as follows:

```
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 fancy(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 = fancy(text)
19     print("The fancy text is", fancy_text)
20
21
```

The code defines a `symbol` function that returns "*" for even-length strings and "_" for odd-length strings. It then defines a `fancy` function that uses `symbol` to create a decorative border around the input text. Finally, it defines a `main` function that prompts the user for input and prints the result of `fancy`. The `def main():` line is highlighted with a red box.

Function stack diagram

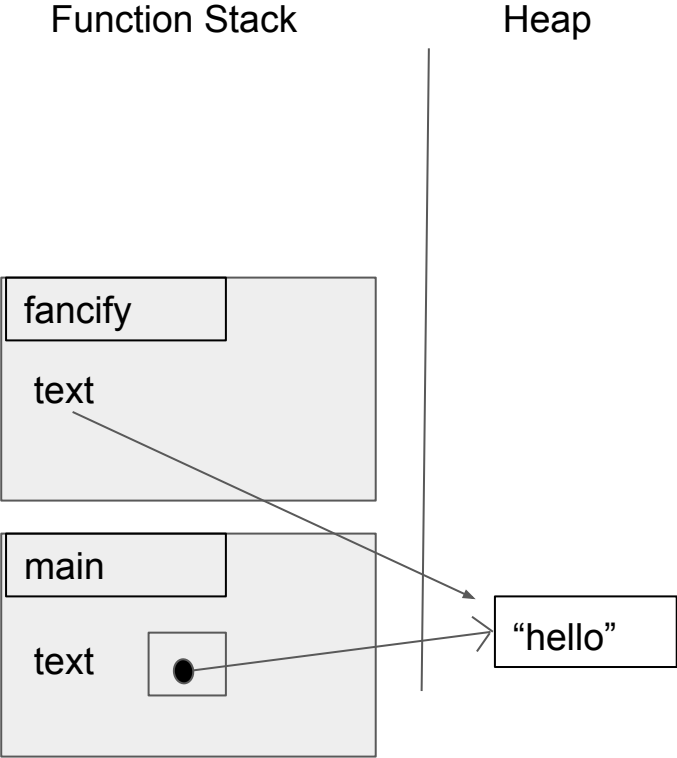


```
fancy.py -- ~/CS21-devel/examples/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
add.py add2.py add3.py tip.py fancy.py
1 """
2 Asks the user to enter a string and then prints it in a fancy
3 """
4
5 def symbol(text):
6     if len(text) % 2 == 0:
7         return "*"
8     else:
9         return "_"
10
11 def fancy(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 = fancy(text)
19     print("The fancy text is", fancy_text)
20
21
```

Assume user enters "hello"

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

Function stack diagram

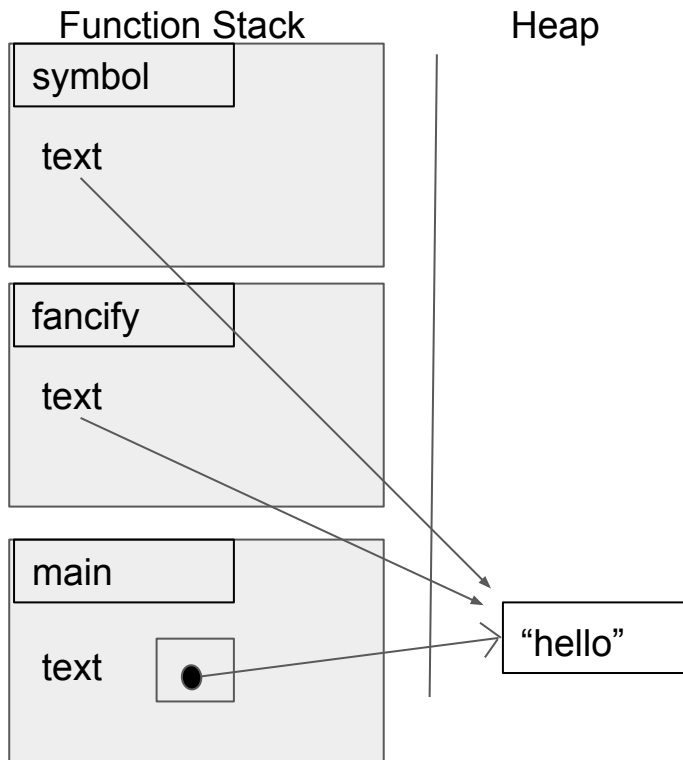


The screenshot shows a code editor window titled `fancify.py` with the following Python code:

```
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
```

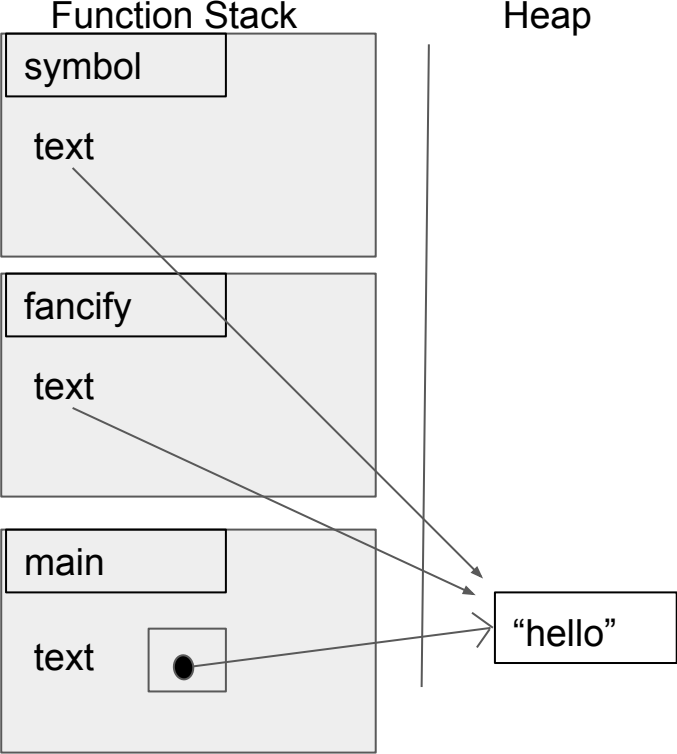
The `def fancify(text):` line is highlighted with a red box. The status bar at the bottom shows `fancify.py 19:43` and `LF N UTF-8 Python 0 files`.

Function stack diagram



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

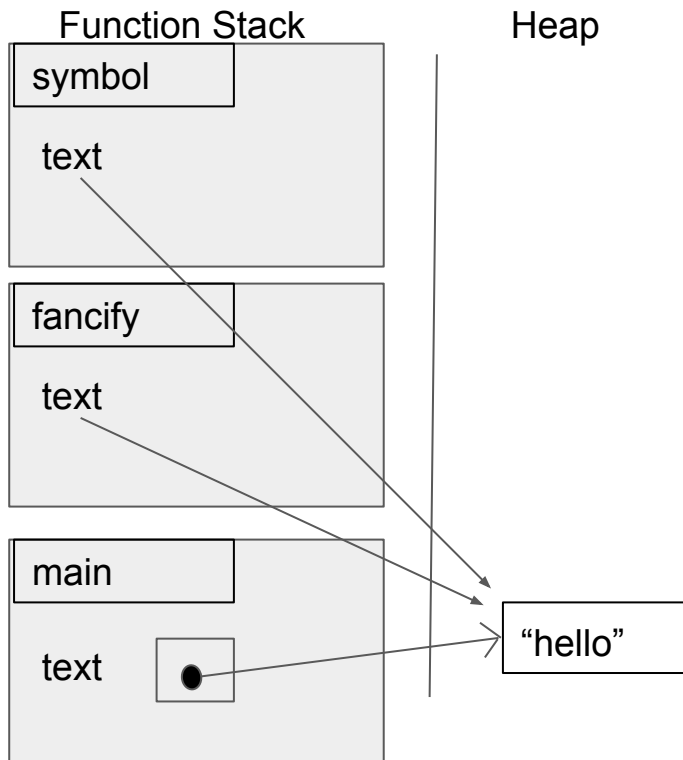
Function stack diagram



```
fancify.py --~/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
```

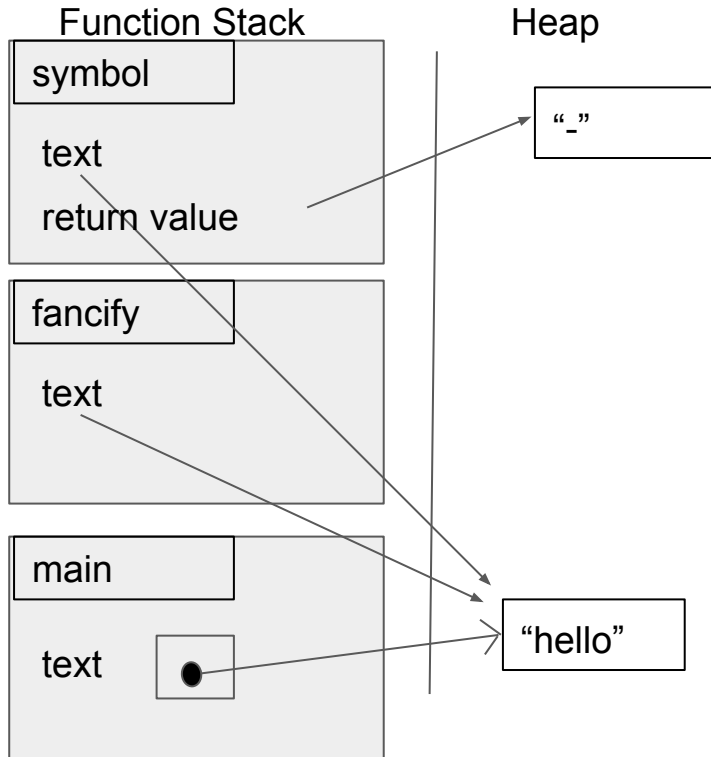
condition is False

Function stack diagram



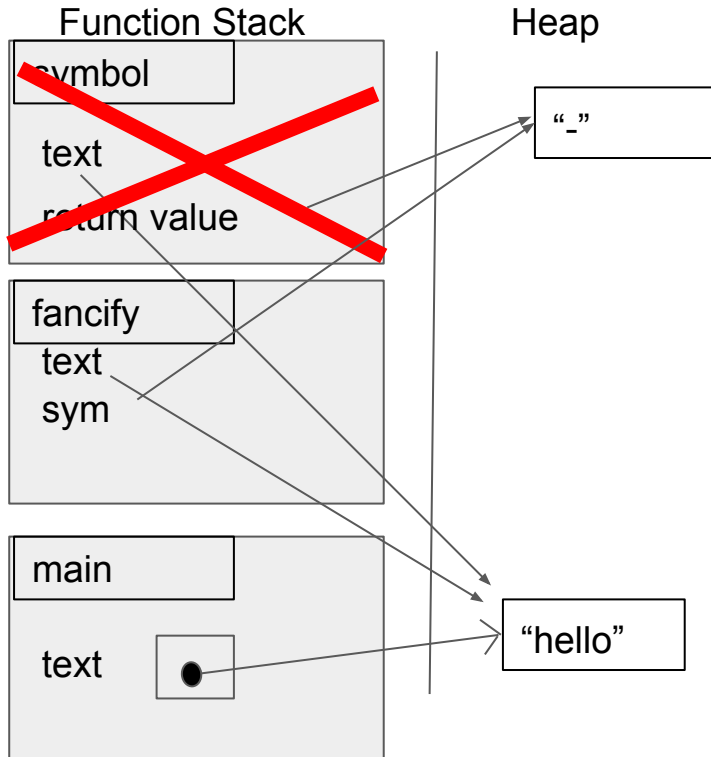
```
fancy.py -- ~/CS21/cs21-devel/examples/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
add.py add2.py add3.py tip.py fancy.py
1 """
2 Asks the user to enter a string and then prints it in a fancy
3 """
4
5 def symbol(text):
6     if len(text) % 2 == 0:
7         return "*"
8     else:
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
```

Function stack diagram



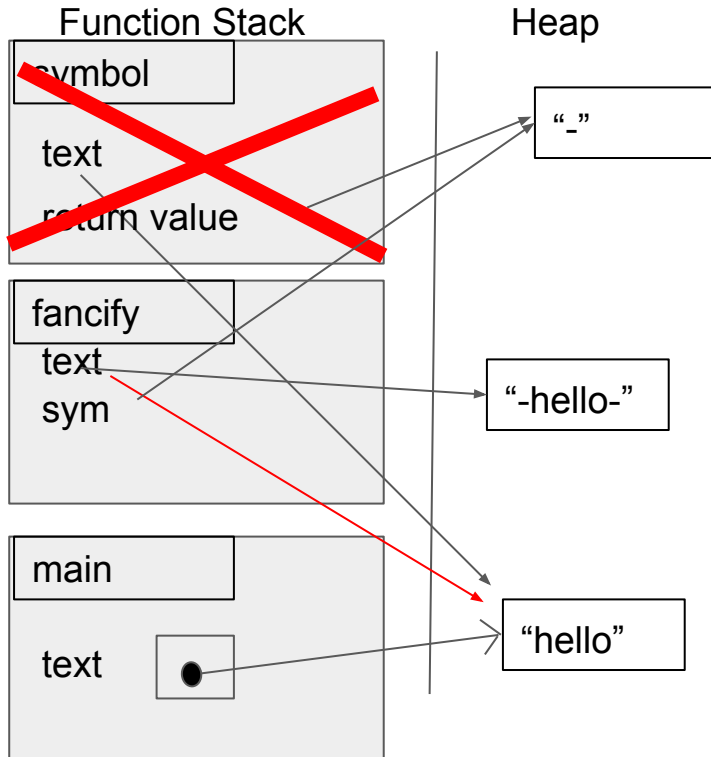
```
fancyfy.py --~/CS21/cs21-devel/examples/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
add.py add2.py add3.py tip.py fancyfy.py
1 """
2 Asks the user to enter a string and then prints it in a fancy
3 """
4
5 def symbol(text):
6     if len(text) % 2 == 0:
7         return "*"
8     else:
9         return "_"
10
11 def fancyfy(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 = fancyfy(text)
19     print("The fancy text is", fancy_text)
20
21
```

Function stack diagram



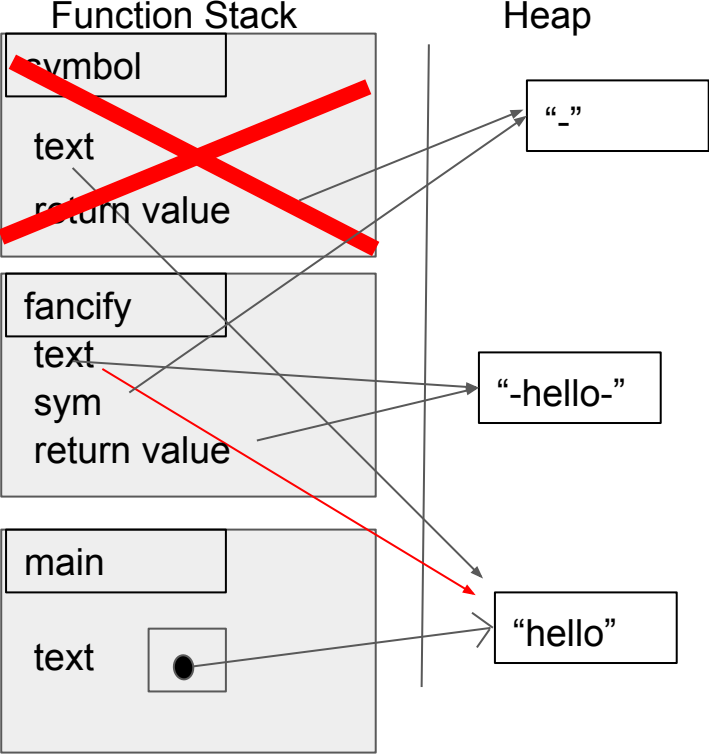
```
fancy.py -- ~/CS21/cs21-devel/examples/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
add.py add2.py add3.py tip.py fancy.py
1 """
2 Asks the user to enter a string and then prints it in a fancy
3 """
4
5 def symbol(text):
6     if len(text) % 2 == 0:
7         return "*"
8     else:
9         return "_"
10
11 def fancy(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 = fancy(text)
19     print("The fancy text is", fancy_text)
20
21
```

Function stack diagram



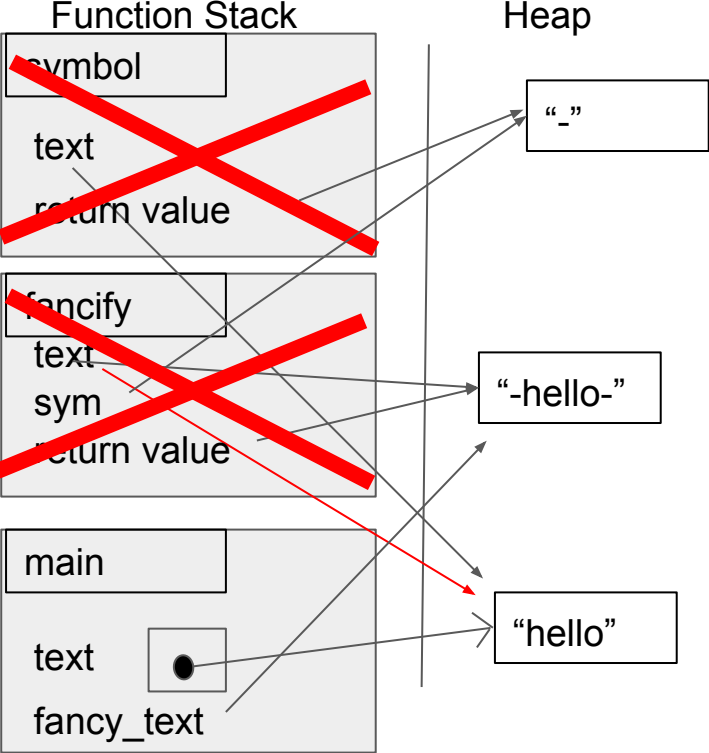
```
fancy.py -- ~/CS21/cs21-devel/examples/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
add.py add2.py add3.py tip.py fancy.py
1 """
2 Asks the user to enter a string and then prints it in a fancy
3 """
4
5 def symbol(text):
6     if len(text) % 2 == 0:
7         return "*"
8     else: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

Function stack diagram



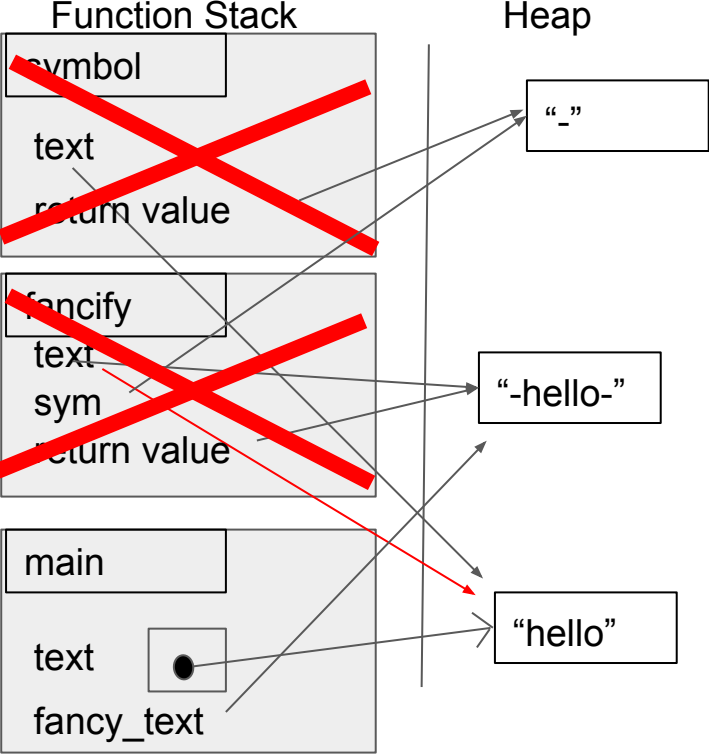
```
fancyfy.py -- ~/CS21/cs21-devel/examples/inclass/w05 -- Atom
File Edit View Selection Find Packages Help
add.py add2.py add3.py tip.py fancyfy.py
1 """
2 Asks the user to enter a string and then prints it in a fancy
3 """
4
5 def symbol(text):
6     if len(text) % 2 == 0:
7         return "*"
8     else:
9         return "_"
10
11 def fancyfy(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 = fancyfy(text)
19     print("The fancy text is", fancy_text)
20
21
```

Function stack diagram



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