References

CMPT 145

Variables

A variable has 3 aspects:

1. Its name
2. Its value
3. Its location (or *address*)

The Call Stack, Frames, and the Heap

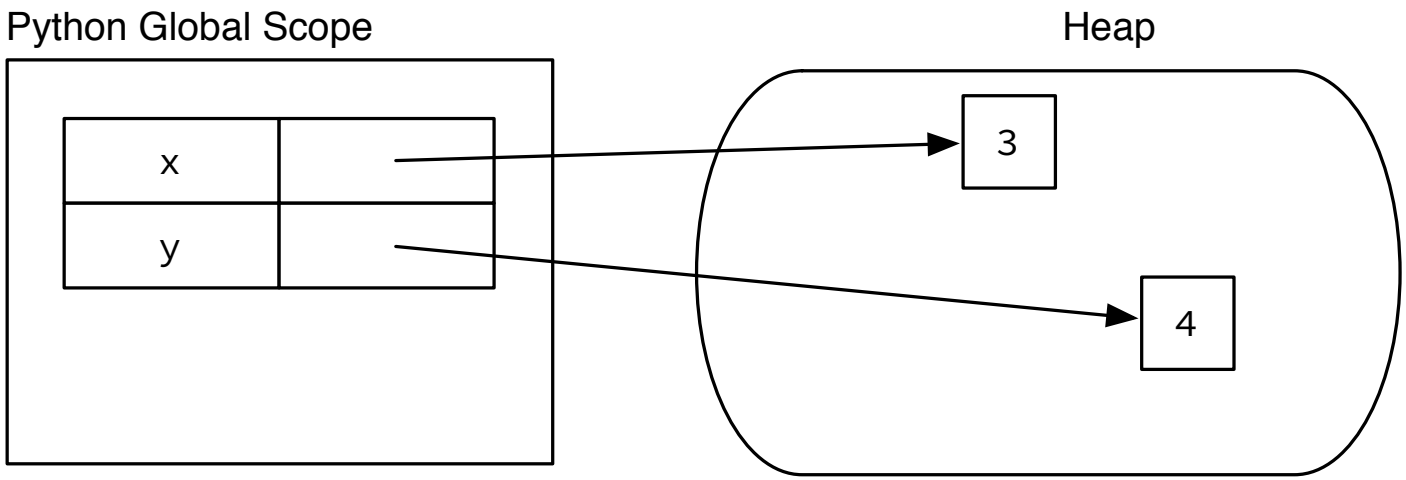
In Python:

- All values (e.g., numbers, strings, lists, objects, ...) are stored on the heap
- Variables are stored in a table called a frame
- Frames are stored on the call stack
- A frame associates a variable name with its value using a reference

References

In Python:

- A frame associates a variable name with its value using a reference
- A reference is an address, which gives the location of a value on the heap.
- We can only manipulate references by assignment statements.
- It is more helpful to think of a reference as an arrow to a value on the heap.


Python Global Scope
Heap
x
y
3
4

Exercise 1

Draw a diagram that shows the frames, variables, values and references for the following Python code:

```
1 x = 3
2 y = x + 1
```

Exercise 2

Draw a diagram that shows the frames, variables, values and references for the following Python code:

```
1 x = 3
2 y = x
3 x = 4
```

Python equality

In Python:

- `x == y` is `True` if the values are equal.
- `x is y` is `True` if the references are equal.

Exercise 3

Draw a diagram that shows the frames, variables, values and references for the following Python code:

```
1 x = 3.1
2 y = x
3 x = y + 0.0
4 print(x == y)
5 print(x is y)
```

Exercise 4

Draw a diagram that shows the frames, variables, values and references for the following Python code:

```
1 x = [1, 2, 3]
2 y = x
3 y.append('four')
```

Exercise 5

Draw a diagram that shows the frames, variables, values and references for the following Python code:

```
1 x = [1, 2, 3]
2 y = x + ['four']
```

Exercise 6

Draw a diagram that shows the frames, variables, values and references for the following Python code:

```
1 alist = [1,2,3]
2 total = 0
3 for v in alist:
4     total = total + v
```

Exercise 7

Draw a diagram that shows the frames, variables, values and references for the following Python code:

```
1 alist = [1, '2', {'three':3}]
2 copy = []
3 for v in alist:
4     copy.append(v)
```

Functions and the Stack

In Python:

- Calling a function pushes a new frame on the call stack.
- The function's parameters are variables in the frame.
- The parameter's values are copies of references to the arguments in the function call.
- New variables in the body of the function are added to the (new) frame.
- When the function returns, the function's stack frame is removed from the stack.

Exercise 8

Draw a diagram that shows the frames, variables, values and references for the following Python code:

```
1  x = 5
2
3  def fun(a, b):
4      x = a + b
5      return x * 2
6
7  y = fun(x, x + 1)
```

Exercise 9

Draw a diagram that shows the frames, variables, values and references for the following Python code:

```
1  x = [1, 2]
2
3  def fun2(a, b):
4      a.append(b)
5      a.append(b)
6      return a
7
8  y = fun2(x, x[0])
```

Exercise 10

Draw a diagram that shows the frames, variables, values and references for the following Python code:

```
1 x = {'first': 1, 'second': None}
2 x['third'] = 3
```

Exercise 11

Draw a diagram that shows the frames, variables, values and references for the following Python code:

```
1 x = {'first': 1, 'second': None}
2 y = {'first': 2, 'second': x}
```