

LAMDAS AND ENVIRONMENT DIAGRAMS 4

COMPUTER SCIENCE 88

September 22, 2021

1 Lambdas

Lambda expressions are one-line functions that specify two things: the parameters and the return expression.

A lambda expression that takes in no arguments and returns 8:

lambda : $\underbrace{8}_{\text{return value}}$

A lambda expression that takes two arguments and returns their product:

lambda $\underbrace{x, y}_{\text{parameters}}$: $\underbrace{x * y}_{\text{return expression}}$

Unlike functions created by a `def` statement, the function object that a lambda expression creates has no intrinsic name and is not bound to any variable. In fact, nothing changes in the current environment when we evaluate a lambda expression unless we do something with this expression, such as assign it to a variable or pass it as an argument to a higher order function.

1. What would Python print?

```
>>> a = lambda: 5
>>> a()
```

Solution:

```
5
```

```
>>> a(5)
```

Solution:

```
TypeError: <lambda>() takes 0 positional arguments but 1
was given
```

```
>>> b = lambda: lambda x: 3
>>> b()(15)
```

Solution:

```
3
```

```
>>> c = lambda x, y: x + y
>>> c(4, 5)
```

Solution:

```
9
```

```
>>> d = lambda x: lambda y: x * y
>>> d(3)
```

Solution:

```
<function ...>
```

```
>>> d(3)(3)
```

Solution:

```
9
```

```
>>> e = d(2)
>>> e(5)
```

Solution:

10

```
>>> f = lambda: print(1)
```

Solution:*# No output*

```
>>> g = f()
```

Solution:

1

2 Environment Diagrams

1. Draw the environment diagram for evaluating the following code

```
def mystery_a(lst):  
    def mystery_b(color, count):  
        lst.extend([color] * count)  
    return mystery_b
```

```
colors = ["purple", "pink", "brown"]  
f = mystery_a(colors)  
f("red", 3)  
f("blue", 1)
```

Solution: Solution: [python tutor link](#)

2. If on line 2 and line 4, we replace `mystery_b` with `mystery_a`, what will change in the environment diagram, if anything?

Solution: Solution: Only the name of frame 2 would change to `mystery_a`. Nothing else would change, as `mystery_a` would just be a new variable defined in the scope of the `f1` frame and would point to the same function object as before, the function object defined on line 2.

3. If on line 3, we change `lst.extend([color] * count)` to `lst.append([color] * count)`, what will change, if anything?

Solution: Solution: The list `lst` would grow in length by 1 element, where that one new element would be a list of length 'count' and every item in the list would have value equal to `color`.

4. Draw the environment diagram for evaluating the following code

```
def ross(geller, num):  
    return geller(monica(num))
```

```
def monica(num):  
    if num >= 2:  
        return tup[0]  
    return tup[num]
```

```
f = lambda x: x[-1] == "a"  
tup = ("hola", "there")  
rachel = ross(f, 5)
```

Solution: Solution: [python tutor link](#)

5. Draw the environment diagram for evaluating the following code

```
def anna(olaf):  
    return lambda a, b: olaf or [a] * b
```

```
hans = [1]  
elsa = anna(hans.append(4))  
kristoff = elsa(3, 4)
```

Solution: Solution: <https://tinyurl.com/14zjl83t>