

Lambdas & HOF

Lambda : a way to write a python function in one line
format: lambda param: RV

ex. `>>> def square(x):`
 `return x**2` \rightarrow `>>> square = lambda x : x**2`

`>>> square(2)`
4

`>>> square(2)`
4

Can take multiple arguments

ex. `>>> def multiply(x, y, z):`
 `return x*y*z` \rightarrow `>>> multiply = lambda x, y, z: x*y*z`

`>>> multiply(1, 2, 3)`
6

`>>> multiply(1, 2, 3)`
6

Can also have no arguments

ex. `>>> def tru():`
 `return True`
`>>> tru()`
True

\rightarrow `>>> tru = lambda: True`
`>>> tru()`
True

① Write a lambda function for the function below :

`def division(x, y):`
 `return x/y`

`>>> 3 / 2`
1.5

`>>> 6 / 2`
3.0

`>>> 3 // 2`
1

`>>> 6 // 2`
3

division = lambda x, y: x/y ,

In Env. Diagrams:

$x = \text{lambda } n: n * 5$

$z = x(5)$

$y = x(3)$ CP: G

primitive types [P = F2]

- integers
- floats
- strings

} anything that's not a primitive type is an object

Fill Out!

GF | $x \rightarrow \lambda(n) [P = G]$
 $z \rightarrow 25$
 $y \rightarrow 15$
 parameter

f1 | $\lambda(n) [P = G]$

n | 5

RV | 25

f2 | $\lambda(n) [P = G]$

n | 3

RV | 15

★ Open a frame every time a function is called

Higher Order Functions: a type of function

which takes another function(s) as a parameter and/or returns another function(s).

* In other words just a function that contains another function

ex. $\text{def operation}(func, x, y):$
 $\text{result} = func(x, y)$
 return result

$\text{def mult}(x, y):$
 $\text{return } x * y$


which function is the higher order function & why?

$\ggg \text{operation}(\text{mult}, 6, 7)$
 42

Lambdas come in handy with higher-order functions

```
ex. def print-word(string):  
    return lambda: print(string)  
        lambda x: print(x)
```

def print-word(string)
def p():
 print(string)
 return p



```
>>> hi = print-word("hi")  
>>> hi()  
hi
```

hi is the lambda function
function call to hi

② Write a higher-order function which takes in an operation function (any function that takes in two numbers as parameters and returns a number) and returns an operation function which divides the result of the input operation function by 2.

```
def divide-by-two(operation):
```

```
    """  
    >>> def mult(x,y)  
        return x*y
```

```
    >>> a = divide-by-two(mult)
```

```
    >>> a(2,3)
```

```
    3.0
```

```
    """
```

```
    return lambda x,y: operation(x,y) / 2
```



Env Diagram to above Problem

