Function Examples

Announcements

Review

What Would Python Display?

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

```
from operator import add, mul
def square(x):
    return mul(x, x)
```

A function that takes any argument and returns a function that returns that arg

```
def delay(arg):
    print('delayed')
    def g():
        return arg
    return g
```

Names in nested def statements can refer to their enclosing scope

This expression	Evaluates to	Interactive Output
`5`	`5`	`5`
`print(5)`	`None`	`5`
`print(print(5))`	`None`	`5` `None`

`delay(delay)()(6)()`	`6`	`delayed`
		`delayed`
		`6`

`print(delay(print)()(4))`	`None`	`delayed`
		`4`
		`None`

None

What Would Python Print?

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

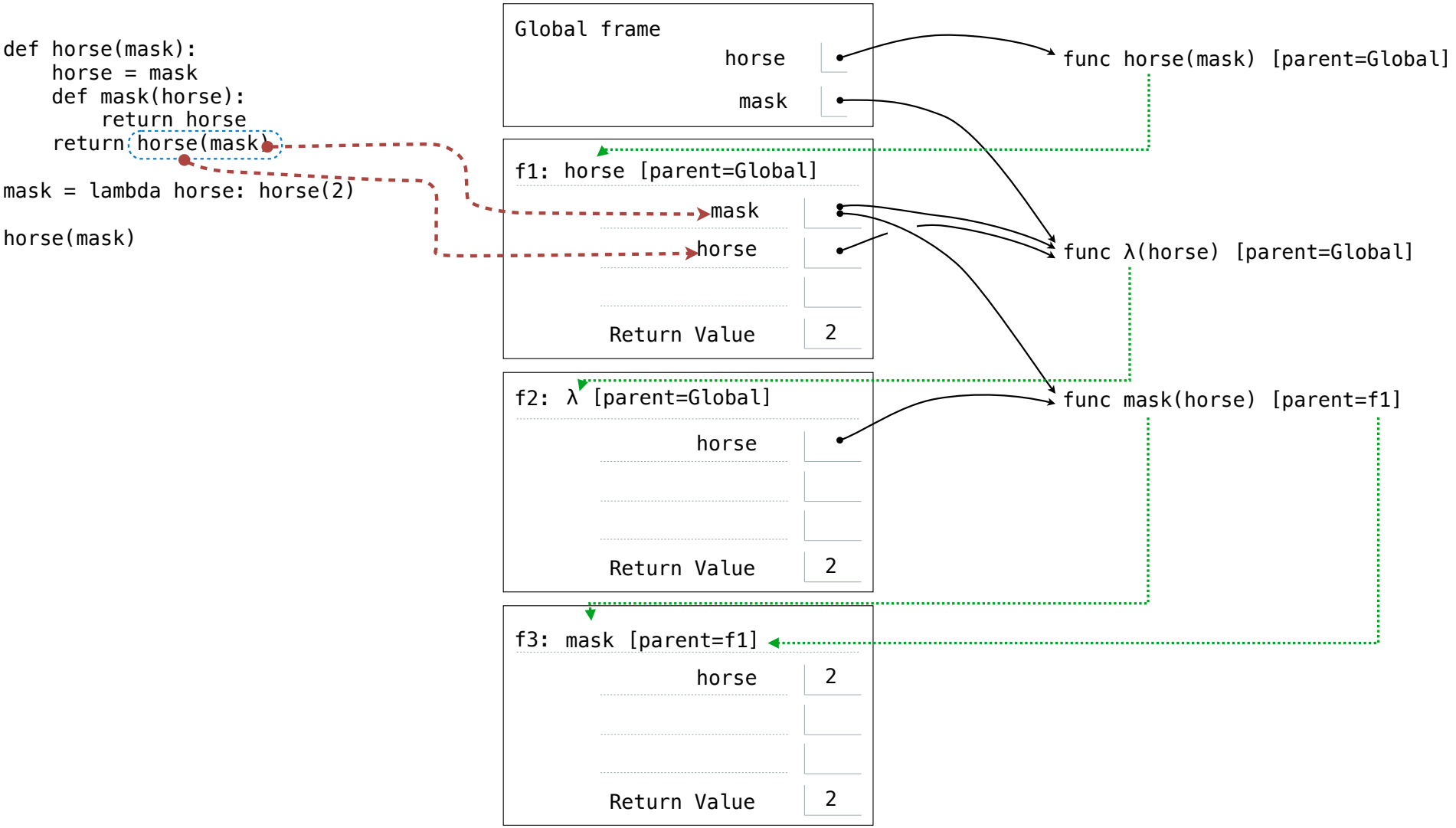
```
from operator import add, mul
def square(x):
    return mul(x, x)
```

A function that always returns the identity function

```
def pirate(arggg):
    print('matey')
    def plunder(arggg):
        return arggg
    return plunder
```

This expression	Evaluates to	Interactive Output
`add(pirate(3)(square)(4), 1)` *func square(x)* *16*	17	Matey 17
`pirate(pirate(pirate))(5)(7)` *Identity function* *5*	Error	Matey Matey Error

A name evaluates to the value bound to that name in the earliest frame of the current environment in which that name is found.


def horse(mask):
horse = mask
def mask(horse):
return horse
return horse(mask)
mask = lambda horse: horse(2)
horse(mask)
Global frame
horse
mask
func horse(mask) [parent=Global]
f1: horse [parent=Global]
mask
horse
Return Value
2
func λ(horse) [parent=Global]
f2: λ [parent=Global]
horse
Return Value
2
func mask(horse) [parent=f1]
f3: mask [parent=f1]
horse
2
Return Value
2

Implementing Functions

Implementing a Function

```
def remove(n, digit):
    """Return all digits of non-negative N
    that are not DIGIT, for some
    non-negative DIGIT less than 10.

    >>> remove(231, 3)
    21
    >>> remove(243132, 2)
    4313
    """
    kept, digits = 0, 0

    while _____:

        n, last = n // 10, n % 10

        if _____:

            kept = _____

            digits = _____

    return _____
```

231

4

1	1
+ 20	+ 30
	+ 200
21	231

n > 0

last != digit

~~10*~~kept + last*10**digits

digits + 1

231

kept

Read the description

Verify the examples & pick a simple one

Read the template

Implement without the template, then change your implementation to match the template.
OR
If the template is helpful, use it.

Annotate names with values from your chosen example

Write code to compute the result

Did you really return the right thing?

Check your solution with the other examples

Implementing a Function

```
def remove(n, digit):
    """Return all digits of non-negative N
    that are not DIGIT, for some
    non-negative DIGIT less than 10.

    >>> remove(231, 3)
    21
    >>> remove(243132, 2)
    4313
    """
    kept, digits = 0, 0
    while ____________n > 0____________:
        n, last = n // 10, n % 10
        if ____last != digit____:
            kept = ____kept/10 +  last____
            digits = ____digits + 1____
    return ____round(kept * 10 ** (digits-1))____
```

231

3

21

Read the description

Verify the examples & pick a simple one

Read the template

Implement without the template, then change your implementation to match the template.
OR
If the template is helpful, use it.

Annotate names with values from your chosen example

Write code to compute the result

Did you really return the right thing?

Check your solution with the other examples

Decorators

Function Decorators

(Demo)

Function decorator

```
@trace1
def triple(x):
    return 3 * x
```

Decorated function

is identical to

Why not just use this?

```
def triple(x):
    return 3 * x
triple = trace1(triple)
```