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	None	5 None
delay(delay)()(6)()	6	delayed delayed 6
print(delay(print)()(4))	None	delayed 4 None

Slide 1: 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
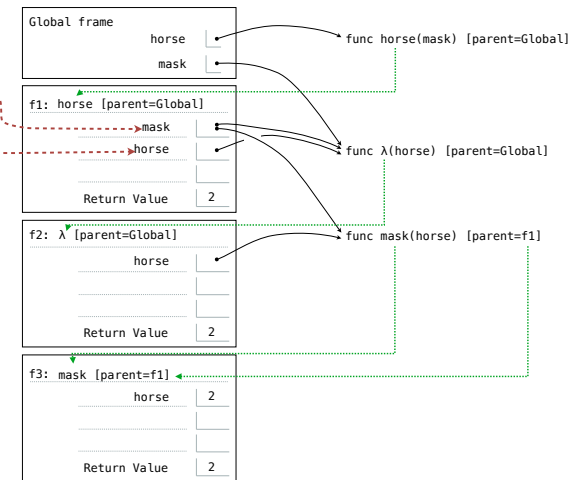always returns the
identity function

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

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

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

6

Slide 2: Environment Diagram

```
def horse(mask):
    horse = mask
    def mask(horse):
        return horse
    return horse(mask)

mask = lambda horse: horse(2)

horse(mask)
```

Global frame
- horse → func horse(mask) [parent=Global]
- mask → func λ(horse) [parent=Global]

f1: horse [parent=Global]
- mask
- horse
- Return Value: 2

func mask(horse) [parent=f1]

f2: λ [parent=Global]
- horse
- Return Value: 2

f3: mask [parent=f1]
- horse
- Return Value: 2

Slide 3: Implementing Functions

Slide 4: 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 = 10*kept + last*10**digits
            digits = ____digits + 1____
    return ____kept____
```

```
231    4
       1       1
       + 20    + 30
               + 200
       ___     ___
       21      231
```

231 (kept annotation)

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

6

Implementing a Function

```
def remove(n, digit):
    """Return all digits of non-negative N
       [231]  re    [3]  IT, for some
           ga       IT 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_____
    [21]
            digits = _____digits + 1_____

    return ___round(kept * 10 ** (digits−1))___
```

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
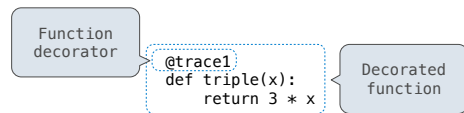example

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          Decorated
                def triple(x):   function
                    return 3 * x
```

is identical to

```
Why not just    def triple(x):
use this?           return 3 * x
                triple = trace1(triple)
```