```
#########################
## EXAMPLE: combinations of print and return
## Python Tutor link:
http://www.pythontutor.com/visualize.html#code=def%20is_even_with_return(%20i%20%29%3A%0A%20%20
%20%20%22%22%22%20%0A%20%20%20%20Input%3A%20i,%20a%20positive%20int%0A%20%20%20%20Returns%20Tru
e%20if%20i%20is%20even,%20otherwise%20False%0A%20%20%20%20%22%22%22%0A%20%20%20%20print('with%2
0return'%29%0A%20%20%20%20remainder%20%3D%20i%20%25%202%0A%20%20%20%20return%20remainder%20%3D%
3D%200%0A%0Ais_even_with_return(3%29%20%0Aprint(is_even_with_return(3%29%20%29%0A%0Adef%20is_ev
en_without_return(%20i%20%29%3A%0A%20%20%20%20%22%22%22%20%0A%20%20%20%20Input%3A%20i,%20a%20po
sitive%20int%0A%20%20%20%20Does%20not%20return%20anything%0A%20%20%20%20%22%22%22%0A%20%20%20%2
0print('without%20return'%29%0A%0Ais_even_without_return(3%29%0Aprint(is_even_without_return(3%
29%20%29%0A&cumulative=false&curInstr=0&heapPrimitives=false&mode=display&o
rigin=opt-frontend.js&py=3&rawInputLstJSON=%5B%5D&textReferences=false
############
#############
def is_even_with_return( i ):
    """
    Input: i, a positive
int
    Returns True if i is even, otherwise False
    """
    print('with
return')
    remainder = i % 2
    return remainder == 0

is_even_with_return(3)

print(is_even_with_return(3) )

def is_even_without_return( i ):
    """

Input: i, a positive int
    Does not return anything
    """
    print('without
return')
    remainder = i % 2

is_even_without_return(3)
print(is_even_without_return(3) )

#
Simple is_even function definition
def is_even( i ):
    """
    Input: i, a
positive int
    Returns True if i is even, otherwise False
    """

remainder = i % 2
    return remainder == 0

# Use the is_even function later on in the
code
print("All numbers between 0 and 20: even or not")
for i in range(20):
    if
is_even(i):
        print(i, "even")
    else:
        print(i,
"odd")

#########################
## EXAMPLE: applying functions to repeat same task
many times
#########################
def bisection_cuberoot_approx(x, epsilon):

"""
    Input: x, an integer
    Uses bisection to approximate the cube root of
x to within epsilon
    Returns: a float approximating the cube root of x

"""
    low = 0.0
    high = x
    guess = (high + low)/2.0
    while
abs(guess**3 - x) >= epsilon:
        if guess**3 < x:
            low = guess

else:
            high = guess
        guess = (high + low)/2.0
    return guess

x = 1
while x
<= 10000:
    approx = bisection_cuberoot_approx(x, 0.01)
    print(approx, "is close
to cube root of", x)
    x *= 10


#########################
## EXAMPLE: functions as
arguments
## Python Tutor link:
http://www.pythontutor.com/visualize.html#code=def%20func_a(%29%3A%0A%20%20%20%20print('inside%
20func_a'%29%0A%0Adef%20func_b(y%29%3A%0A%20%20%20%20print('inside%20func_b'%29%0A%20%20%20%20r
eturn%20y%0A%0Adef%20func_c(z%29%3A%0A%20%20%20%20print('inside%20func_c'%29%0A%20%20%20%20retu
rn%20z(%29%0A%0Aprint(func_a(%29%29%0Aprint(5%2Bfunc_b(2%29%29%0Aprint(func_c(func_a%29%29%0A&a
mp;cumulative=false&curInstr=0&heapPrimitives=false&mode=display&origin=opt-fro
ntend.js&py=3&rawInputLstJSON=%5B%5D&textReferences=false
#########################

def func_a():
    print('inside func_a')

def func_b(y):
    print('inside func_b')
    return
y

def func_c(z):
    print('inside func_c')
    return
z()

print(func_a())
print(5+func_b(2))
print(func_c(func_a))


#########################
##
EXAMPLE: returning function objects
## Python Tutor link:
http://www.pythontutor.com/visualize.html#code=def%20f(%29%3A%0A%20%20%20%20def%20x(a,%20b%29%3
A%0A%20%20%20%20%20%20%20%20return%20a%2Bb%0A%20%20%20%20return%20x%0A%20%20%20%20%0Aval%20%3D%
20f(%29(3,4%29%0Aprint(val%29%0A&cumulative=false&curInstr=0&heapPrimitives=false&a
mp;mode=display&origin=opt-frontend.js&py=3&rawInputLstJSON=%5B%5D&textReferenc
es=false
#########################
def f():
    def x(a, b):
        return a+b
    return x


# the first part, f(), returns a function object
# then apply that function with parameters 3
```

```
and 4
val = f()(3,4)
print(val)



#########################
## EXAMPLE: shows accessing
variables outside scope
#########################
def f(y):
    x = 1
    x += 1
    print(x)
x
= 5
f(x)
print(x)

def g(y):
    print(x)
    print(x+1)
x = 5
g(x)
print(x)

def h(y):

pass
    #x += 1 #leads to an error without line `global x` inside h
x =
5
h(x)
print(x)


#########################
## EXAMPLE: hader scope example from slides
##
Python Tutor link:
http://www.pythontutor.com/visualize.html#code=def%20g(x%29%3A%0A%20%20%20%20def%20h(%29%3A%0A%
20%20%20%20%20%20%20%20x%20%3D%20'abc'%0A%20%20%20%20x%20%3D%20x%20%2B%201%0A%20%20%20%20print(
'in%20g(x%29%3A%20x%20%3D',%20x%29%0A%20%20%20%20h(%29%0A%20%20%20%20return%20x%0A%0Ax%20%3D%20
3%0Az%20%3D%20g(x%29&cumulative=false&curInstr=0&heapPrimitives=false&mode=disp
lay&origin=opt-frontend.js&py=3&rawInputLstJSON=%5B%5D&textReferences=false
###
######################
def g(x):
    def h():
        x = 'abc'
    x = x + 1
    print('in
g(x): x =', x)
    h()
    return x

x = 3
z = g(x)


#########################
## EXAMPLE:
complicated scope, test yourself!
## Python Tutor link:
http://www.pythontutor.com/visualize.html#code=def%20f(x%29%3A%0A%20%20%20x%20%3D%20x%20%2B%201
%0A%20%20%20print('in%20f(x%29%3A%20x%20%3D',%20x%29%0A%20%20%20return%20x%0A%0Ax%20%3D%203%0Az
%20%3D%20f(x%29%0Aprint('in%20main%20program%20scope%3A%20z%20%3D',%20z%29%0Aprint('in%20main%2
0program%20scope%3A%20x%20%3D',%20x%29%0A%0Adef%20g(x%29%3A%0A%20%20%20%20def%20h(x%29%3A%0A%20
%20%20%20%20%20%20%20x%20%3D%20x%2B1%0A%20%20%20%20%20%20%20%20print(%22in%20h(x%29%3A%20x%20%3
D%20%22,%20x%29%0A%20%20%20%20x%20%3D%20x%20%2B%201%0A%20%20%20%20print('in%20g(x%29%3A%20x%20%
3D%20',%20x%29%0A%20%20%20%20h(x%29%0A%20%20%20%20return%20x%0A%0Ax%20%3D%203%0Az%20%3D%20g(x%2
9%0Aprint('in%20main%20program%20scope%3A%20x%20%3D%20',%20x%29%0Aprint('in%20main%20program%20
scope%3A%20z%20%3D%20',%20z%29%0A&cumulative=false&curInstr=0&heapPrimitives=false&
amp;mode=display&origin=opt-frontend.js&py=3&rawInputLstJSON=%5B%5D&textReferen
ces=false
```

```
#########################
def f(x):
   x = x + 1
   print('in f(x): x =', x)

return x

x = 3
z = f(x)
print('in main program scope: z =', z)
print('in main program scope: x
=', x)

def g(x):
    def h(x):
        x = x+1
        print("in h(x): x = ", x)

x = x + 1
    print('in g(x): x = ', x)
    h(x)
    return x

x = 3
z = g(x)
print('in main
program scope: x = ', x)
print('in main program scope: z = ', z)
```