Ch6 Functions

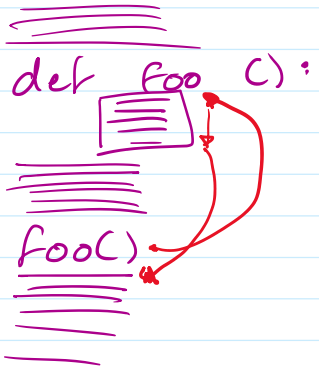
Monday, October 16, 2023 8:18 AM

Programming Functions ≠ math functions.

Definition:
A "named" sequence of statements.
A "function call" is an invocation of the function
It causes the statements of the function to be executed.

Syntax:
form 1: def function_name ( ):
block.

Intuitively:



- A function can receive data, called parameters.

form 2: def function_name (p1, p2, ..., pn)
parameters

- A function that expects parameters has a function call with arguments.

```
def say_hello ( x )
    print ('Hello', x)
```

```
say_hello( 'Eric')
```

X is the parameter.

'Eric' is the argument

- A function can return a value.
with the return statement
return expression.
the returned value is used in the expression

with the return statement

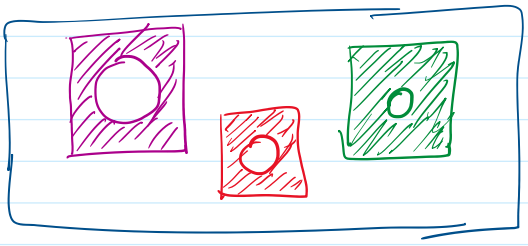
return expression.

the returned value is used in the expression that contains the function call.

- Why function:
  1. Code organization.
  2. Avoid repeating code.

E.g.

problem:



Given the width and radius of each fabric. Compute the total area.

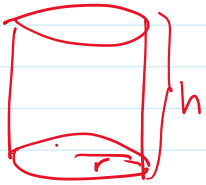
width w radius r

the area of one fabric is (w * w) - (Pi * r * r)

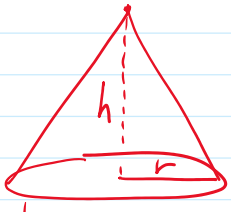
```
def area_of_fabric(w, r):
    return (w*w) - (pi * r * r)
```

- functions can call functions.

area_circle(r)



volume_cylinder(r, h)



volume_cone(r, h)

- function bodies or blocks are allowed to have any python statement, including Branches & loops.

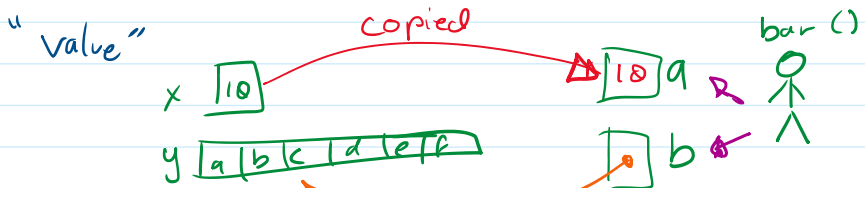
- Python does not checks for types:
  warning

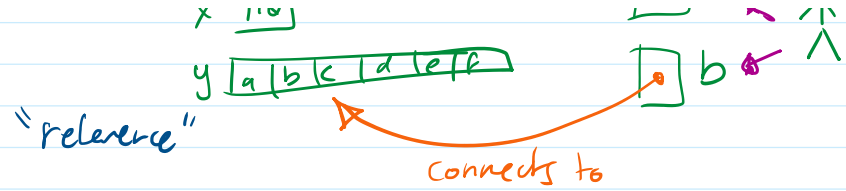
- Python does not checks for types:
  warning
  Python is a "weakly typed" language
  Python crashes when operators are used in the wrong types.

- Document your functions;
  doc strings.
  """
  comment here.
  """
  - Purpose:- what the function does.
  - Preconditions:- what does the function expects.
  - Post conditions:- any outcomes of the function besides "return" value.

- The "Scope" of variables:
  Scope:- the range of statements over which a variable is visible/available.
  Global Scope:- are available throughout the program*
  Local Scope:- are available only on the function they are created.
  global var :- use a global variable inside a function.

- On modifying arguments/parameters.
  - pass-by-value
  - pass-by-reference





- immutable values (int, float, strings, tuples) are passed by value
- mutable values (lists, dictionaries, sets...) are passed by reference

- Beyond positional arguments.
  - Default arguments.
    arg = val in parameter list
  - Named arguments
    arg = val in function call
  - Variadic arguments.
    * args
  - Variadic keyword arguments.
    ** kwargs
- Returning multible values. *
  return a, b, c