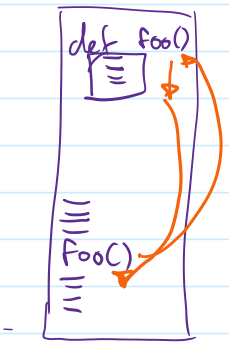
6 Functions

Monday, February 26, 2024 2:21 PM

Programming Functions ≠ Mathematical Functions.

Definition
A "named" sequence of statements.
A "function call" is an invocation of a function that causes the statements in the function to execute.

Syntax: def function_name ():
block



Use: encapsulate repeated tasks.

- A function can receive data, called parameters.

form 2
def function_name (p1, p2, p3, ..., pn)
block
parameters

Parameters become variables inside of block.

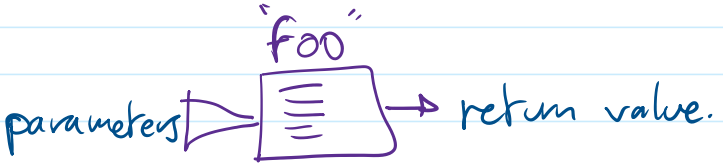
- A function that expects parameters has a function call with arguments

eg. def say_hello (x):
print ('Hello', x)
parameter.

say_hello ('kyle')
argument

- A function can also return a value using the return expr statement
  the returned value is then used in the expression that contains the function call.

- function blocks are allowed to have any valid statement even conditionals and loops.

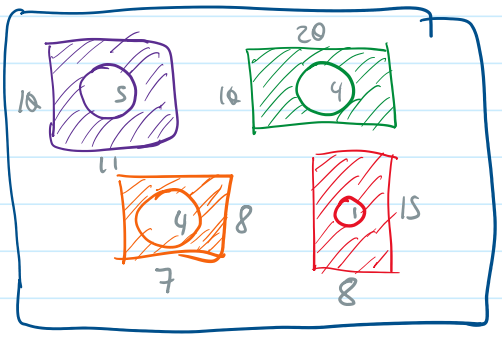


- Why functions:-
  1:- Code organization
  2:- Code re-use.

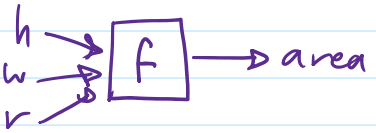
Example:-

story:

floormats

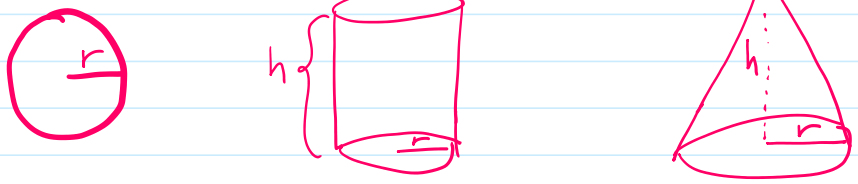


Question: what the area covered by the floormats?




```
def area_of_floormat(h, w, r):
    a = (h * w) - (pi * r * r)
    return a
```

- functions can call other functions.



area_circle(r) vol_cylinder(h,r) vol_cone(r,h)

```
pi = 3.15159

def area_circle( r ) :
    a = pi*r*r
    return a

def vol_cylinder( h, r ) :
    v = h*area_circle(r)
    return v

def vol_cone( h, r ) :
    v = vol_cylinder(h,r) / 3
    return v
```

Python does not checks whether arguments are compatible with a function.

- Python is called "weakly typed"
- Python crashes. 💥 when operators are used with incompatible operands.

- Document your functions:

  Doc strings.

  """
  your comments here.
  """

  what to put in doc-string.
  - Purpose of your function
  - Preconditions : what kinds of values the function expects.
  - Post-conditions:- outcomes of the function

- The "Scope" of variables

  Scope :- the range of statements over which a variable is visible or available.

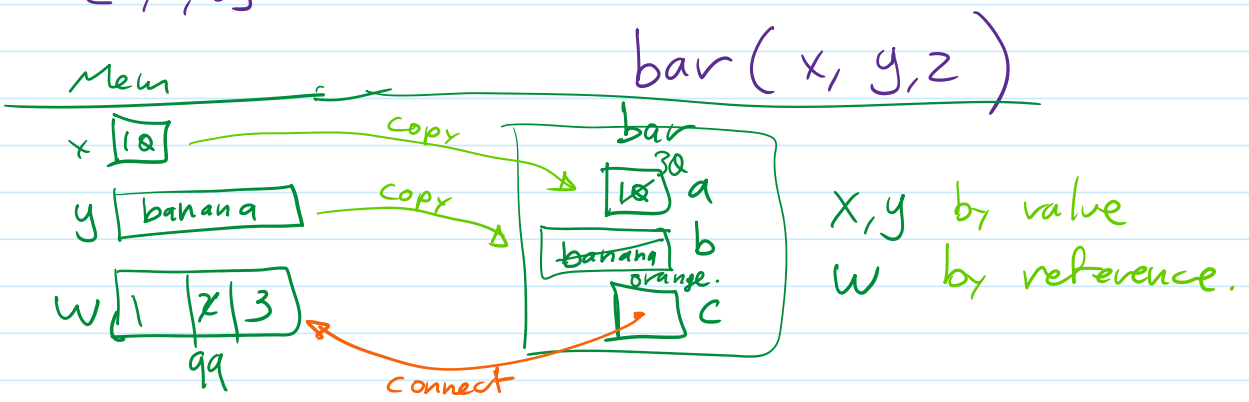
  "Global" :- are available throughout the program
  "Local" :- are available only on the function they are created.

  - Local scope "masks" Global scope.
  - global var — use a global variable inside of a function.

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

```
X = 10
y = 'banana'
W = [1, 2, 3]
```

```
def bar(a, b, c)
    ≡
```



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

- Beyond positional arguments
  - Default arguments
    arg = val in parameter list.
  - Named arguments.
    arg = val in the function call.
  - variadic arguments.
    *args in the argument list.
  - variadic keyword arguments
    **kargs in the argument list.

- Returning multiple values.
  return var_1, var_2, var_3

EOF