

## Programming Functions $\neq$ Mathematical Functions

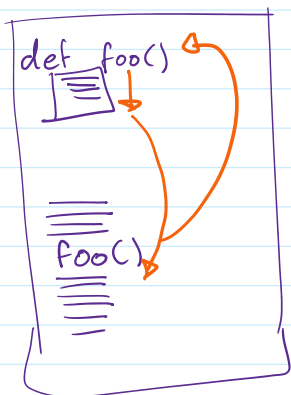
Definition:-

A named block of statements

A function call is an invocation of a function

Syntax:

```
def function_name():
    block
```



Use: encapsulate repeated tasks

- A function can receive data, called parameters

```
def function_name(p1, p2, p3, ..., pn)
    block.
```

parameters become variables inside of the block

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

e.g.

```
def foo(name):
    print('*****')
    print('*', name, '*')
    print('*****')
```

```
print('hello World')
foo('Bob')
foo('Alice')
print('Done!')
```

- a function can also return a value
  - so that you can use functions in expressions.

Syntax: return expr

- execution of the function stops

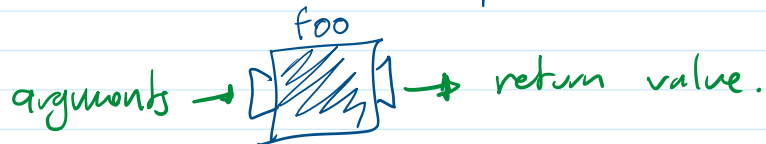
syntax. return expr

- execution of the function stops
- the returned value is used in the expression that contains the function call.

e.g.

$$x = 27 + \text{foo}() * 3$$

programming functions can be represent mathematical functions

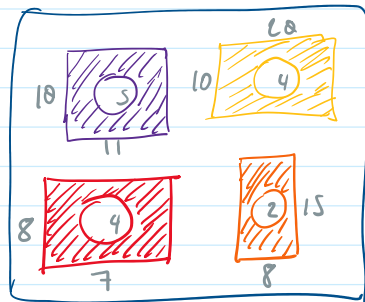


• why functions?

- 1.- code organization.
- 2.- code re-use.

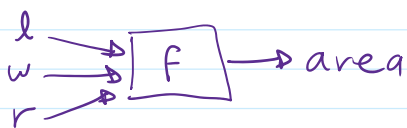
• Example

"floor mats"



Question:  
what is the area covered by the floor mats?

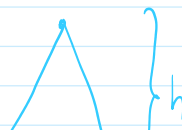
one floor mat  $(l * w) - (\pi * r * r)$



```
def area_of_floor_mat(l, w, r)
    a = (l * w) - (pi * r * r)
    return a.
```

• functions can call other functions.

E.g.

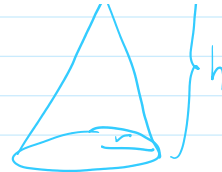




area\_circle(r)



vol\_cylinder(r, h)



vol\_cone(r, h)

pi = 3.141592

```
def area_circle( r ) :
    a = pi*r*r
    returns 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
```

### • The "scope" of a variable.

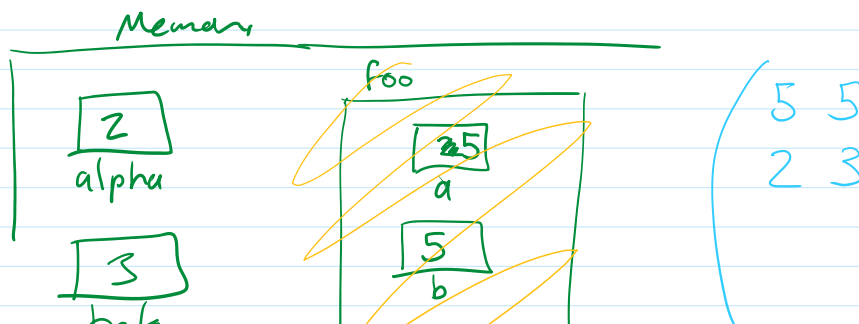
The scope is the range of statements over which a variable is visible

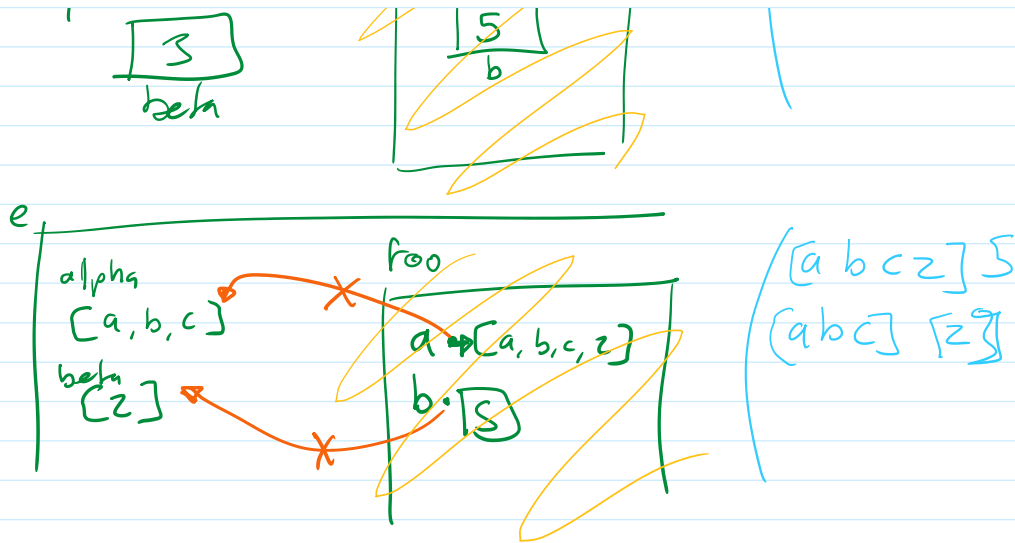
- "Global" :- are available throughout the program
- "Local" :- are available only in the function they are created.

- Local Scope "Masks" Global scope.
- global var use a global variable inside a function.

### • On modifying arguments / parameters.

- Some parameters are treated as local variables some are not.





- immutable variables are copied: "pass-by-value"  
int, float, strings, tuples
- mutable variables are linked: "pass-by-reference"  
lists, dictionaries, sets.



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

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

- Document your functions:  
Use comments to remind you what a function does.  
Use "doc strings"

```
"""
    your comments
"""
```

below function name.

what to put in a doc-string?

- Purpose of a function

- Purpose of a function
- Preconditions - what kinds of values the function expects
- Post-conditions - outcome of the function.

—●— EOF.