# Chapter-3

# Functions in Math

- Given some number (*Input*)
- Calculates the result (*Output*)

**Ex:**

a = 3
**f(x)** = x*x
f(a) = 3*3 = 9

# Functions in Programming

- A named sequence of of statements that performs a computation
- Not only numbers, but different types (**Ex:** string, list)
- Not only calculations (**Ex:** print out, draw, change value)

**Ex:**

```
>>> type(63)
<class 'int'>
```

- *Function Call*

# Built-in Functions

**Ex:**

```python
print()

type(x)

pow(x,y)
```

- Name, *argument*, result (*return value*)

A function *takes* an argument, and *returns* a result.

# Type Conversion Functions

```
>>> int(3.14)

3

>>> float('2.79')

2.79

>>> str(777)

'777'
```

# Math Functions

- Module (library): a file consisting of related functions

**Ex:**

```
>>> import math
>>> print(math)
```

- Dot Notation

**Ex:**

```
>>> math.sin(radians)
>>> math.pi
```

# Composition

```
>>> x = math.sin(degrees / 360.0 * 2* math.pi)

>>> x = math.exp(math.log(x+1))



>>> minut = saghat * 60

>>> saghat * 60 = minut
```

# Adding New Function

```python
def function_name():
    # action1 (statement)
    # action2
    …
```

- We *define* a function, then we *call* it.

- *Function object*

```python
print(fun)
type(fun)
```

# Flow of Execution

```python
def fun1():
    fun2()
    print(1)

def fun2():
    print(2)

def fun3():
    print(3)
    fun1()

>>> fun3()
```

# Arguments and Parameters

```
type(x)
pow(b,d)

def square_area(side):
    print(side*side)
x=4
square_area(x)
```

- A function requires an argument(s)
- An argument is assigned to function parameter

# Scope: Variables & Parameters are Local

```python
def square_area(side):
    area = side*side
    print(area)


x=4
square_area(x)
print(side)
print(area)
```

- *Stack Diagram*: See textbook sections 3.8 and 3.9

# Fruitful & void Functions

- *Fruitful*: returns a result (**Ex:** `math.sqrt()`)
- *void*: performs an action, but doesn't return a value (**Ex:** `print()`)

- *None*: a special value with its own type

# Why use Functions?

- ...
- ...
- ...
- ...