Lecture #14
CSE-2040 Programming IV

Transforming Education. Enriching Lives

Sin Thi Yar Myint

Professor

Faculty of Computer Science

MIIT

What will be cover today

- Defining a function •
- Arguments to a function •
- Function Parameters vs Arguments •
- Types of parameters and arguments

Types of arguments

Types of arguments

Type	Description	Example
Positional	• Arguments without a name • Each is matched to the positional-or-keyword or positional-only parameter with the same index or to the var-positional parameter if there is no matching index	`fn(10,20)`
Keyword	• Arguments with a name • Each is matched to the positional-or-keyword or keyword-only parameter with the same name or to the var-keyword parameter if there is no matching name	`fn(a=10,b=20)`

```
def compute(a, b, c = 1):
    return a ** 2 + b ** 2 - c
```

Function Call	a	b	c	returns
compute(1,2,3) Only positional	1	2	3	2
compute(1,2) Positional and default	1	2	D	4
compute(c = 5, b = 2, a = 3) Only named	3	2	5	8
compute(b = 2, a = 2) Named and default	2	2	D	7
compute(5, c = 2, b = 1) Positional and named	5	1	2	24
compute(8, b = 0) Positional, named and default	8	0	D	63

Example

Game of Casting a Die

A die has six faces with each face having a dot ranging from 1 to 6.

Two players cast a die a certain number of times. The number of times the die is cast is decided by the players. We will call this number `n`. Value of `n` should be even. Player 1 casts the die first, followed by player 2. The game ends when both the players have cast the die `n` times. players have won the game or the game is a draw. A draw is when both players gain equal points at the end of the game.

Since this is a simulation of a game with dice, you have the liberty to change the number of faces of the die to be greater than 6.

Var Positional & VarKeyword

- Passing variable number of arguments: *args and ** kwargs.

```
def sum(*args):
    total = 0
    for num in args:
        total = total + num
    return total
print(sum(1,2,3,4,5,6,7,8))
```

```
def printStudentProfile(**kwargs):
    for key, value in kwargs.items():
        print("{} is {}".format(key, value))
printStudentProfile(**{'Name': 'Doraemon', 'Age': 8})
printStudentProfile(**{'Name': 'Nobita', 'Age': 8, 'Gender':
'Male'})
printStudentProfile(**{'Name': 'Aye Aye', 'Age': 18, 'Gender':
'Female','Rollno':12,'Academadic Year':2021})
```

- Name resolution of identifiers
- • Identifier •
  - Identifies something in a program
  - • Variable
  - • Function
  - • Module •
  - Scope of variables assigned inside a function • Local to that function
  - • Scope of variables assigned inside an enclosing function •
  - Nonlocal to nested function •
  - Scope of variables assigned outside of functions •
  - Global in the enclosing module

Functions are first class objects

- Possible to assign a function to a variable
- Possible to define one function inside another function
- Possible to pass a function as parameter to another function
- Possible that a function can return another function.

REFERENCES

- https://docs.python.org/3/tutorial/inputoutput.html#reading-and-writing-files
- Core Python Programming- Chapter-9 Dr.R.Nageswaro Rao, second edition

Thank you

Successful and unsuccessful people do not vary greatly in their abilities. They vary in their desires to reach their potential. – *John Maxwell*