Lecture 08: User Defined Functions

Swakkhar Shatabda

B.Sc. in Data Science
Department of Computer Science and Engineering
United International University

January 17, 2024

What is a function?

- Named blocks of code designed to do one specific job.
- Examples:
  - int, float, print, input, type, len
- More:
  - sum, max, min

```python
listA = [1,2,3,4,5,6,7,6,5,4,3,2,1]
print("Sum:",sum(listA))
print("Min:",min(listA))
print("Max:",max(listA))
```

Write your own function!

- When you want to perform a particular task that you've defined in a function, you call the function responsible for it.
- If you need to perform that task multiple times throughout your program, you don't need to type all the code for the same task again and again; you just call the function dedicated to handling that task.
- Two types of functions.
  - Returns something - similar to mathematical functions,

  $$f(n) = \frac{n(n+1)}{2}$$

  - Does not return anything - void.

First Function

```
def myFunc():
    print("The weather is nice")
```

To call this function, we need to name it, similar to print function.

```
myFunc()
```

We have called the function by writing its name.

```
The weather is nice
```

- A function definition begins with the def keyword, followed by the function name, a set of parentheses and a colon (:).
- Like variable identifiers, by convention function names should begin with a lowercase letter and in multiword names underscores should separate each word.
- The required parentheses contain the function's parameter list.

Passing Values

```python
def myFunc2(season):
    print("The weather is nice in",season)

myFunc2("Spring")
myFunc2("Autumn")
```

- An argument is a piece of information that's passed from a function call to a function.

```
The weather is nice in Spring
The weather is nice in Autumn
```

Returning Values

```
def calcSquare(n):
    return n*n

sq = calcSquare (5)
print(sq)
```

- The value the function returns is called a return value.
- The return statement takes a value from inside a function and sends it back to the line that called the function.

Find Output - I

```python
def myFunc(x):
    sum = 0
    for i in range(x):
        sum += i
    return sum
n = myFunc(6)
print(n)
```

Multiple parameters

```python
def infoCountry(country, capital):
    print("The capital of",country,"is",capital)

infoCountry("Bangladesh","Dhaka")
```

```
The capital of Bangladesh is Dhaka
```

Find Output - II

```python
def multStrings(a,b):
    if len(a)==len(b):
        print("Equal Length Strings")
    else:
        print("Not Equal Length Strings")

multStrings("Hello","Python")
```

Positional parameters

```python
def infoCountry(country, capital):
    print("The capital of",country,"is",capital)

infoCountry("Dhaka","Bangladesh")
```

```
The capital of Dhaka is Bangladesh
```

- The order of the arguments in your function call must match the order of the parameters in the function's definition.

Keyword Arguments

- A keyword argument is a name-value pair that you pass to a function.
- You directly associate the name and the value within the argument, so when you pass the argument to the function, there's no confusion.
- The order of keyword arguments doesn't matter.

```python
def infoCountry(country, capital):
    print("The capital of",country,"is",capital)


infoCountry(capital="Dhaka",country="Bangladesh")
```

```
The capital of Bangladesh is Dhaka
```

Default Values

- When writing a function, you can define a default value for each parameter.
- If an argument for a parameter is provided in the function call, Python uses the argument value.
- If not, it uses the parameter's default value.

```python
def infoCountry(country, capital='unknown'):
    print("The capital of",country,"is",capital)

infoCountry("India")
```

```
The capital of India is unknown
```

List as parameter

```python
def maximum(l):
    maxi = l[0]
    for i in range(1,len(l)):
        if l[i] > maxi:
            maxi = l[i]
    return maxi

print("Maximum:",maximum([1,2,3,4,5,6,7,1,2,3,4,5]))
```

```
Maximum: 7
```