Information Technology Institute

Python

Day02

Programming for everyone

Prepared by

Noha Shehab

05

Data structures

-Tuples-

Tuples

- Type of sequence but immutable datatype.

```
t = tuple()
```

- Immutable datatype means once created You cannot change it
- Tuple like list but immutable
- To define a tuple:

```
t = tuple([5, 6, 7])
t2 = ("iti", "3DFX", "python", "databases“)
```

- Lists can hold different values, with different datatypes.

```
t = tuple([5, 6, 7])

t3 = ("a", 5, "test", True, t)

print(t3)
```

- Tuple can hold different tuples also.

Tuple operations

- Get items at certain index:

```
z = ("abc", 55, 67)
print(z[2])
```

- Len

```
print(len(z))
```

- Concatenation

```
m = ("abc", True, 40 ,66)
n = ("python", "test", "iti", 88)
t = m + n
print(t)
```

- Membership

```
# membership
t4 = (“Python", “Maya", “c#“)
print(“Mohamed" in t4)
```

Tuple operations

- iterations

- Min, max

```
x = (5,66,77)
print(min(x))
```

```
x = (5,66,77)
print(max(x))
```

- Empty tuples are falsy values

```
# empty lists
t = ()
if t:
    print(“Non empty tuple")
else:
    print(“Empty tuple ")
```

- Tuple of one item

```
#create tuple of one item
unit = ("item",)
print(type(unit))
```

```
del unit
```

02

Data structures

-Dictionaries-

Dictionaries

- A dictionary: key: value comma separated elements data structure.

```
d = {} # empty dictionary
print(type(d))
```

```
info = {"name": "Noha", "email": "nshehab@iti.gov.eg"}
print(info)
```

- Keys: doesn't allow duplicates for keys

```
# access element using keys
print(info["name"])
# dictionary is mutable datatype
info["name"]= "Noha Shehab"  # check if key exists  --> update item
print(info)
# add new item
info["age"]= 29
print(info)
```

Dictionaries

- Len, keys, values, items.

```
# dictionary values from 3.7 ---> items of dictionary are stored ordered
print(len(info))
# get keys
print(list(info.keys()))
# get values
print(list(info.values()))
# get items
print(info.items())
```

- update

```
info = {"name": "Noha", "email": "nshehab@iti.gov.eg"}
d2 = {"work": "iti", "course": "python", "name": "noha"}
info.update(d2)
```

Dictionaries

- Check existence of keys.

```
info = {"name": "Noha", "email": "nshehab@iti.gov.eg", "course": "python"}
# check key exists in dictionary
if "name" in info:  # check the keys
    print("hi")
else:
    print("bye")
```

- Check existence of values.

```
# check values
if "python" in info.values():
    print("hi")
else:
    print("bye")
```

Dictionaries

- Loop over dictionary.

```
# loop over the dictionary
for item in info:
    print(f"{item} = {info[item]}")

for key,value in info.items():  # list of tuples
    print(f"{key} = {value}")
```

- Clear, del

```
# remove all items in dictionary
info.clear()
print(info)
del(d)
```

06

Ranges, Looping

Ranges

- Range function: returns a range object
  - `range([start,] end[, step])`
- Ranges

```
x = range(5)
y = range(0, 5, 1)
z = range(1, 10, 2)
m = range(100, 90, -2)
```

- Iterations

```
for i in range(10):
    print(i)
```

While, Break, Continue

- While loop:

```
dayCount = 0
while dayCount < 4:
    print("We are learning Python")
    dayCount += 1
```

- Break

```
for i in range(10):
    if i == 4:
        break
    print(i)
```

- Continue

```
for i in range(10):
    if i == 4:
        continue
    print(i)
```

For else, Pass statement

- For else.

```
for i in range(10):
    if i == 4:
        continue
    print(i)
else:
    print("loop ended")
```

- Pass

```
for i in range(10):
    pass
```

Input function

- Input function:

```
name = input("What’s your Name? ")
print(name)
```

- Input function: returns with string.

03

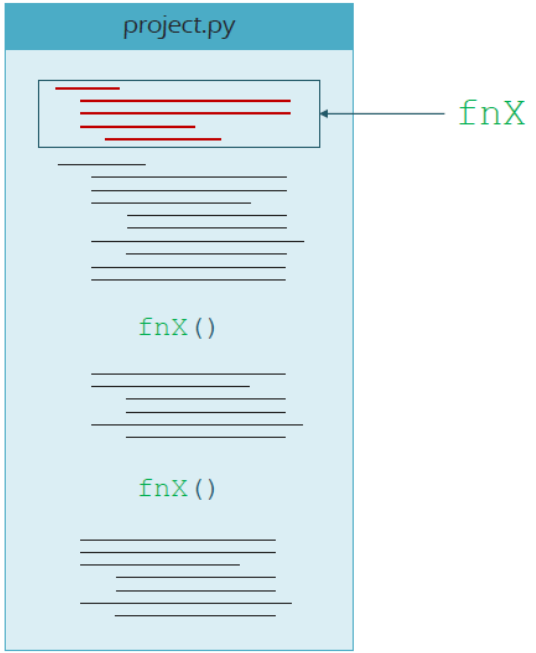
Functions

02

Functions

Functions

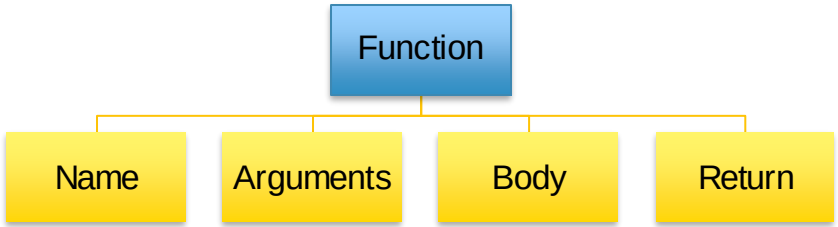
- Functions are "self contained" modules of code that accomplish a specific task.
- Functions usually "take in" data, process it, and "return" a result.
- Once a function is written, it can be used over and over and over again.
- Functions can be "called" from the inside of other functions.



Functions

- Define a function:

```
def myfunction():
    pass
    return
```



- Example

```
def getfullname(fname, lname):
    fullname = f"{fname} {lname}"
    return fullname
```

```
getfullname("Noha", "Shehab")
```

Functions with default arguments

- Python supports functions default arguments
- Example

```
def summ(num1=0, num2=0):
    res = num1+num2
    return res
```

- Call it using different ways:

```
summ() # 0
summ(10) # 10
summ(10,20) # 30
```

*args: variable number of arguments

- Python supports functions with variable number of arguments using the special syntax *args

```
def displayArgs(*args):
    print(type(args))
    print(len(args))
    for item in args:
        print(f"the argument is {item}")
    return
```

- Call it using different ways:

```
displayArgs()
displayArgs(10,20)
displayArgs("python","arguments",1000, True, [3,66])
```

args just a variable name, you can name it whatever you want ^^

**kwargs === keywords

- Python supports functions with variable number of arguments using the special syntax **Kwargs

```
def getValuePair(**kwargs):
    print(type(kwargs))
    for key, value in kwargs:
        print(f"{key}:{value}")
```

- Call it using different ways:

```
getValuePair(course="iti", track="fx")
```

```
<class 'dict'>
course:iti
track:fx
```

```
getValuePair(name="Noha", dept="opensource", title="TA")
```

```
<class 'dict'>
name:Noha
dept:opensource
title:TA
```

kwargs just a variable name, you can name it whatever you want ^^

05

Time for practice

Lab02

- Write a function that accepts two arguments (length, start) to generate an array of a specific length filled with integer numbers increased by one from start.
- write a function that takes a number as an argument and if the number divisible by 3 return "Fizz" and if it is divisible by 5 return "buzz" and if is is divisible by both return "FizzBuzz"
- Write a function which has an input of a string from user then it will return the same string reversed.
- Ask the user for his name then confirm that he has entered his name(not an empty string/integers). then proceed to ask him for his email and print all this data (Bonus) check if it is a valid email or not

Lab02

- Write a function that takes a string and prints the longest alphabetical ordered substring occurred For example, if the string is 'abdulrahman' then the output is: Longest substring in alphabetical order is: abdu

Lab02

- Write a program which repeatedly reads numbers until the user enters “done”.
  - Once “done” is entered, print out the total, count, and average of the numbers.
  - If the user enters anything other than a number, detect their mistake, print an error message and skip to the next number.

Lab02

- Word guessing game (hangman)
  - A list of words will be hardcoded in your program, out of which the interpreter will
  - choose 1 random word.
  - The user first must input their names
  - Ask the user to guess any alphabet. If the random word contains that alphabet, it
  - will be shown as the output(with correct placement)
  - Else the program will ask you to guess another alphabet.
  - Give 7 turns maximum to guess the complete word.

“It seems impossible until it is done”

Thank you

nshehab@iti.gov.eg