Python

Class 3

Introduction to Python

Azmain Adel

August 13, 2024

01.

Review of Previous Class

Review Topics

- Type casting
- Loops
- Loop controls
- Lists

Solution to Problem 1

Print all even numbers from 2 to 50

Solution:

```
for number in range(2, 51, 2):

    print(number)
```

Solution to Problem 2

Given the list *num_list* = [1, 4, 5, 23, 10, 12, 15, 19, 25]

Loop through the list and print the numbers that are divisible by 5

Solution:

```
num_list = [1, 4, 5, 23, 10, 12, 15, 19, 25]

for num in num_list:

    if num % 5 == 0:

        print(num)
```

02.

More on Python Lists

Looping Lists

```
for item in list:
   # Code block to execute
```

```
for index, item in enumerate(iterable):
```

```
lst = [25, 12, 10, -21, 10, 100]
indices = range(len(lst))
for i in indices:
   print ("lst[{}]: ".format(i), lst[i])
```

Sorting Lists

- `list_name.sort(key=None, reverse=False) # no new list`
- `sorted(list, key=None, reverse=False) # returns new list`

sort()

```
list2 = [10,16, 9, 24, 5]
print ("list before sort", list2)
list2.sort()
print ("list after sort : ", list2)
```

sorted()

```
numbers = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]
# Sorting in descending order
sorted_numbers_desc = sorted(numbers, reverse=True)
print(sorted_numbers_desc)
```

Joining Lists

- `‘+’ operator`
- `extend() method`

Using Concatenation Operator (+)

```
# Two lists to be joined
L1 = [10,20,30,40]
L2 = ['one', 'two', 'three', 'four']
# Joining the lists
joined_list = L1 + L2

# Printing the joined list
print("Joined List:", joined_list)
```

Using extend() method

```
# List to be extended
list1 = [10, 15, 20]
# List to be added
list2 = [25, 30, 35]
# Joining the lists using the extend() function
list1.extend(list2)
# Printing the extended list
print("Extended List:", list1)
```

Joining Lists: More ways!

- Iterate and append

More List Methods

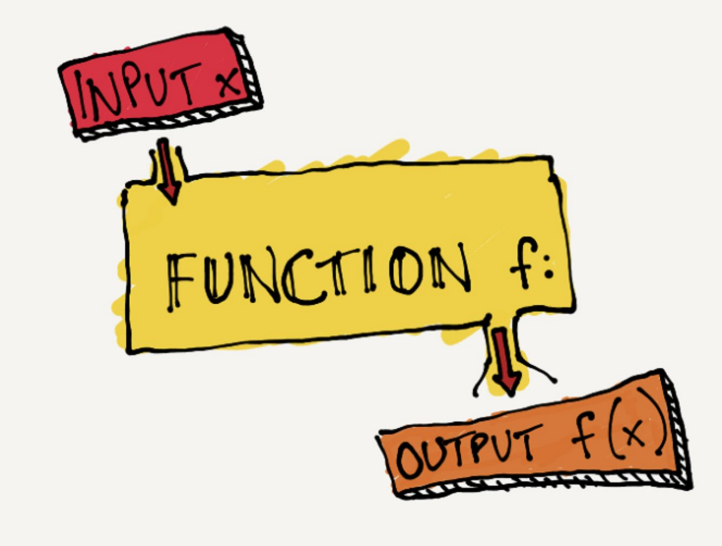
- `list.clear()`
- `list.index(object)`
- `list.count(object)`
- `len(list)`
- `list.copy()`
- `list.reverse()`

03.

Functions

What is a Function?

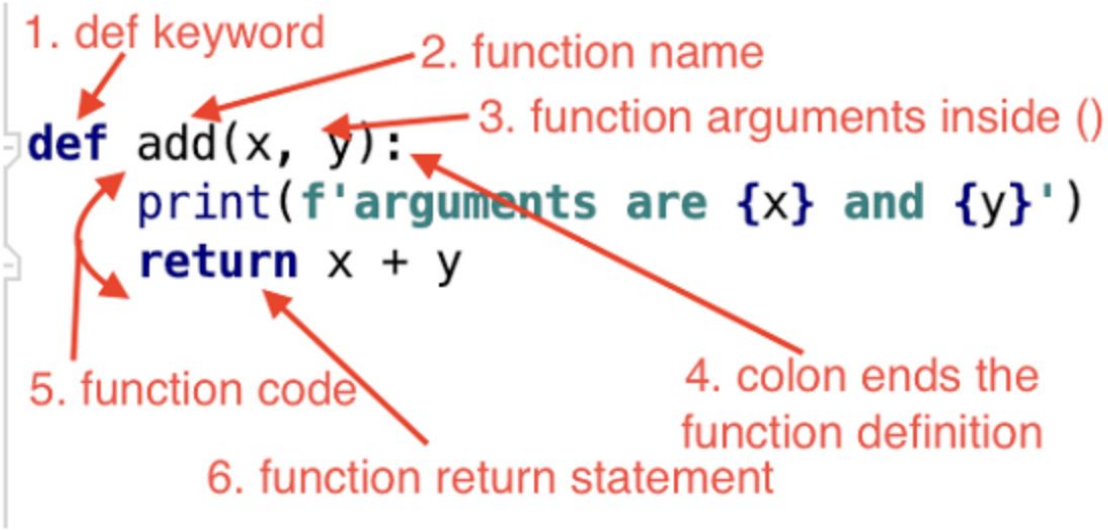
A Python function is a block of organized, reusable code that is used to perform a single, related action.



Function Syntax

```
def function_name( parameters ):
   "function_docstring"
   function_suite
   return [expression]
```

Function Explained



Types of Functions

1. Built-in functions: print(), len(), sum()
2. Module functions
3. User-defined functions

Function Arguments

1. Default
2. Keyword
3. Positional
4. Arbitrary

Default Arguments

```
# Function definition
def showinfo( name, city = "Hyderabad" ):
   "This prints a passed info into this function"
   print ("Name:", name)
   print ("City:", city)
   return

# Now call showinfo function
showinfo(name = "Ansh", city = "Delhi")
showinfo(name = "Shrey")
```

Keyword Arguments

```
# Function definition is here
def printinfo( name, age ):
   "This prints a passed info into this function"
   print ("Name: ", name)
   print ("Age ", age)
   return

# Now you can call printinfo function
# by positional arguments
printinfo ("Naveen", 29)

# by keyword arguments
printinfo(name="miki", age = 30)
```

Positional Arguments

```
def add(x,y):
    z = x+y
    print ("x={} y={} x+y={}".format(x,y,z))
a = 10
b = 20
add(a, b)
```

Arbitrary/Variable Arguments

```
# sum of numbers
def add(*args):
    s=0
    for x in args:
        s=s+x
    return s
result = add(10,20,30,40)
print (result)

result = add(1,2,3)
print (result)
```

04.

Recap and Q&A

Open floor for questions and clarifications

05.

To-do at Home

Problem to Solve 1

Write a program to find sum of all numbers in a list.

Problem to Solve 2

Write a Python function to print the maximum of three numbers.

Output should be: 15 for input of 5, 10, 15

Problem to Solve 3

Write a Python function to check whether a number falls within the range (1,100)

Print “Yes” or “No”

Output should be: “Yes” for input 15 and “No” for input 110

Script review in the next class

Thank you.

azmainadel47@gmail.com
linkedin.com/in/azmainadel
azmainadel.com