Python

Class 2

Introduction to Python

Azmain Adel

August 11, 2024

01.

Review of Previous Class

Review Topics

- What is Python?
- Data types
- Basic input/output
- Conditionals

Solution to Problem 1

Take the heights of your 2 friends (in inches) as input and print their average height!

Solution:

```
height1 = float(input("Enter the height of the first friend in inches: "))

height2 = float(input("Enter the height of the second friend in inches: "))

average_height = (height1 + height2) / 2

print("The average height is", average_height)
```

Solution to Problem 1 (Bonus)

Take the heights of your 2 friends (in inches) as input and print their average height!

Solution:

```
height1 = float(input("Enter the height of the first friend in inches: "))

height2 = float(input("Enter the height of the second friend in inches: "))

average_height = (height1 + height2) / 2

feet = int(average_height // 12)

inches = int(average_height % 12)

print("The average height is", feet, "feet", inches, "inches")
```

Solution to Problem 2

Take two numbers as input, print *"Genjam"* if one is negative and one is positive. Else print "Thik Ase"

Solution:

```
num1 = input("Enter the first number: ")

num2 = input("Enter the second number: ")

if (num1 < 0 and num2 > 0) or (num1 > 0 and num2 < 0):

    print("Genjam")

else:

    print("Thik Ase")
```

02.

More Python Basics

Data Types

- Integers: Whole numbers (e.g., x = 5)
- Floats: Decimal numbers (e.g., y = 3.14)
- Strings: Text data (e.g., name = "Python")
- Booleans: True/False values (e.g., is_active = True)

Proper Inputs!

Input as proper data types

- int(input())
- float(input())

Type Casting

Type casting refers to converting an object of one type into another.

```
<<< a=10   # int object
<<< b=10.5 # float object
```

Comments

```
# This is a comment
print("Hello, World!")
```

```
"""
This function calculates the greatest common divisor (GCD)
of two numbers using the Euclidean algorithm. The GCD of
two numbers is the largest number that divides both of them
without leaving a remainder.
"""
def gcd(a, b):
   while b:
      a, b = b, a % b
   return a

result = gcd(48, 18)
print("The GCD of 48 and 18 is:", result)
```

03.

Control Statements

If-Else Conditionals

```
if expression1:
   statement(s)
elif expression2:
   statement(s)
elif expression3:
   statement(s)
else:
   statement(s)
```

Loops

1. **while** loop: Repeats a statement or group of statements while a given condition is TRUE. It tests the condition before executing the loop body.
2. **for** loop: Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.
3. **nested** loops: You can use one or more loop inside any another while, for or do..while loop.

For Loop

```
for iterating_var in sequence:
   statement(s)
```

For Loop

```
for iterating_var in sequence:
   statement(s)
```

```
range(start, stop, step)
```

While Loop

```
while expression:
   statement(s)
```

Nested Loop

```
for iterating_var in sequence:
   for iterating_var in sequence:
      statements(s)
   statements(s)
```

Loop Control Statements

1. **break statement:** Terminates the loop statement and transfers execution to the statement immediately following the loop.
2. **continue statement**: Causes the loop to skip the remainder of its body and immediately retest its condition prior to reiterating.
3. **pass statement**: The pass statement in Python is used when a statement is required syntactically but you do not want any command or code to execute.

Loop Control

```
looping statement:
   condition check:
      break
```

```
looping statement:
   condition check:
      continue
```

04.

Python Lists

Lists -> [ ]

A Python list is a sequence of comma separated items, enclosed in square brackets [ ]

- List is an ordered collection of items
- Each item in a list has a unique position index, starting from 0.
- Lists may have objects of different data types.

Example: *[1, 2, 'Alif', 34.6, 'c', True]*

Basic List Operations

my_list = [25, 35, 40]

- Access: my_list[2]
- Change/Update: my_list[2] = 45
- Add: my_list.append(55), my_list.insert(3, 55)
- Remove: my_list.remove(55), my_list.pop(3)
- Loop: for num in my_list

05.

Recap and Q&A

Open floor for questions and clarifications

06.

To-do at Home

Problem to Solve 1

Print all even numbers from 2 to 50

Problem to Solve 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

Output should be: 5 10 15 25

Problem to Solve 3

Recap everything and test in your own time

Script review in the next class

Thank you.

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