PYTHON

Lecture – 04 & 05

Recap

- Variable
- Data Type (Numeric, String basic, Boolean)
- Casting
- **Homework**
- **Class work**
    - **Write a python program that produce the result summation, subtraction, multiplication and division of two numbers and print the all 4 results by a single print function.**

Contents

- Python User Input
- Print Output with format
- Operators
- Python Conditions and If…Else statements

Print Output with format

- **Basic Usage**: print("Hello")
- **Multiple Items**: print("a =", a, "b =", b)
- **sep Parameter**: print("a", "b", "c", sep=", ")
- **end Parameter**: print("Hello", end=" World!")
- **Concatenation**: print("Hello, " + name + "!")
- **str.format()**: print("Name: {}, Age: {}".format(name, age))
- **F-Strings**: print(f"Name: {name}, Age: {age}")
- **Formatting Specifiers**: print(f"{pi:.2f}")
- **Number Systems**: print(f"{number:b}")
- **Escaping**: print("He said, \"Hello!\"")

Python User Input

- Python allows for user input. That means we are able to ask the user for input.
- Python (3.6+) uses the **input()** method.

```
yourname = input("Enter Your Name:")
print(”Your name is: " + yourname)
```

More examples:

```
name = input("Enter your name : ")
city = input("Enter your city : ")
print ("Hello My name is", name)
print ("I am from ", city)
```

Examples

- Calculate area of a rectangle from a given height and width. Take input from user/keyboard.

```
width = input("Enter width : ")
height = input("Enter height : ")
area = width*height
print ("Area of rectangle = ", area)
```

Find the problem and solve it.

Converting User Input to Other Data Types

- Input() function takes input (usually from the keyboard) and returns it as a string.
- You can then convert this input to other data types, such as integers or floats, if needed.

```
width = int (input("Enter width : "))
height =  int(input("Enter height : "))
area = width*height
print ("Area of rectangle = ", area)
```

Find the problem and solve it.

Multiple input in a single line

- **input()**: This function reads a line of input from the user.
- **split()**: This method **splits** the input string into a list of substrings based on spaces.
- **map(float/int, ...)**: This converts each substring into a float/int.

```
# Taking two integers as input on the same line
x, y = map(int, input("Enter two numbers: ").split())
print("The sum is:", x + y)

# Taking three float inputs from the user on a single line
a, b, c = map(float, input("Enter three float numbers: ").split())
print("The numbers you entered are:", a, b, c)
```

Python Operators

- Operators are used to perform operations on variables and values.
- Python divides the operators in the following groups:
  - *Arithmetic operators*
  - *Assignment operators*
  - *Comparison operators*
  - *Logical operators*
  - Identity operators
  - Membership operators
  - Bitwise operators

Arithmetic Operators in Python

(Let a=10 and b=20)

Operator	Name	Example
+	Addition	a + b = 30
-	Subtraction	a – b = -10
*	Multiplication	a * b = 200
/	Division	b / a = 2
%	Modulus	b % a = 0
**	Exponent	a**b =10**20
//	Floor Division	9//2 = 4

Arithmetic Operators in Python

```
a = 5
b = 3
c = a + b
print ("a: {} b: {} a+b: {}".format(a,b,c))
c = a / b
print ("a: {} b: {} a/b: {}".format(a,b,c))
c = a//b
print ("a: {} b: {} a//b: {}".format(a,b,c))
c = a % b
print ("a: {} b: {} a%b: {}".format(a,b,c))
c = a**b
print ("a: {} b: {} a**b: {}".format(a,b,c))
```

OUTPUT

a: 5 b: 3 a+b: 8

a: 5 b: 3 a/b: 1.6666667

a: 5 b: 3 a//b: 1

a: 5 b: 3 a%b: 2

a: 5 b: 3 a**b: 125

Assignment Operators in Python

Operator	Example	Same As
=	a = 10	a = 10
+=	a += 30	a = a + 30
-=	a -= 15	a = a - 15
*=	a *= 10	a = a * 10
/=	a /= 5	a = a / 5
%=	a %= 5	a = a % 5
**=	a **= 4	a = a ** 4
//=	a //= 5	a = a // 5
&=	a &= 5	a = a & 5
|=	a |= 5	a = a | 5
^=	a ^= 5	a = a ^ 5
>>=	a >>= 5	a = a >> 5
<<=	a <<= 5	a = a << 5

Assignment Operators in Python

```
a = 21
b = 10
c = 0
c += a
print ("a: {} c += a: {}".format(a,c))
c *= a
print ("a: {} c *= a: {}".format(a,c))
c /= a
print ("a: {} c /= a : {}".format(a,c))
c  = 2
print ("a: {} b: {} c : {}".format(a,b,c))
c %= a
print ("a: {} c %= a: {}".format(a,c))
```

OUTPUT

a: 21 c += a: 21
a: 21 c *= a: 441
a: 21 c /= a : 21.0
a: 21 b: 10 c : 2
a: 21 c %= a: 2

Comparison Operators in Python

Operator	Name	Example
==	Equal	(a == b) is not true.
!=	Not equal	(a != b) is true.
>	Greater than	(a > b) is not true.
<	Less than	(a < b) is true.
>=	Greater than or equal to	(a >= b) is not true.
<=	Less than or equal to	(a <= b) is true.

Comparison Operators in Python

```
a = 21
b = 10
if ( a == b ):
  print ("a is equal to b")
else:
  print ("a is not equal to b")

a,b = b,a #values of a and b swapped.

if ( a <= b ):
  print ("a is either less than or equal to  b")
else:
  print ("a is neither less than nor equal to  b")
```

OUTPUT

a is not equal to b

a is either less than or equal to b

Logical Operators in Python

Operator	Name	Example
and	AND	a and b
or	OR	a or b
not	NOT	not(a)

Logical Operators in Python

```
var = 5


print(var > 3 and var < 10)
print(var > 3 or var < 4)
print(not (var > 3 and var < 10))
```

OUTPUT

```
True
True
False
```

Bitwise Operators in Python

Operator	Name	Example
&	AND	a & b
|	OR	a | b
^	XOR	a ^ b
~	NOT	~a
<<	Zero fill left shift	a << 3
>>	Signed right shift	a >> 3

Bitwise Operators (example)

```python
#Bitwise AND (&)
a = 5  # (in binary: 0101)
b = 3  # (in binary: 0011)
result = a & b  # result is 1 (in binary: 0001)
print(result)  # Output: 1

#Bitwise OR (|)
a = 5  # (in binary: 0101)
b = 3  # (in binary: 0011)
result = a | b  # result is 7 (in binary: 0111)
print(result)  # Output: 7

#Bitwise XOR (^)
a = 5  # (in binary: 0101)
b = 3  # (in binary: 0011)
result = a ^ b  # result is 6 (in binary: 0110)
print(result)  # Output: 6
```

```python
#Bitwise XOR (~)
a = 5  # (in binary: 0101)
result = ~a  # result is -6 (in binary: 1010 in two's complement form)
print(result)  # Output: -6

#Bitwise Left Shift (<<)
a = 5  # (in binary: 0101)
result = a << 1  # result is 10 (in binary: 1010)
print(result)  # Output: 10

#Bitwise Right Shift (>>)
a = 5  # (in binary: 0101)
result = a >> 1  # result is 2 (in binary: 0010)
print(result)  # Output: 2
```

Membership Operators in Python

Operator	Description	Example
in	Returns True if it finds a variable in the specified sequence, false otherwise.	a in b
not in	returns True if it does not finds a variable in the specified sequence and false otherwise.	a not in b

Membership Operators in Python

```
a = 10
b = 5
list = [1, 2, 3, 4, 5 ]

if ( a in list ):
  print ("a is present in the given list")
else:
  print ("a is not present in the given list")

if ( b not in list ):
  print ("b is not present in the given list")
else:
  print ("b is present in the given list")
```

OUTPUT

a is not present in the given list

b is present in the given list

Identity Operators in Python

Operator	Description	Example
is	Returns True if both variables are the same object and false otherwise.	a is b
is not	Returns True if both variables are not the same object and false otherwise.	a is not b

Identity Operators in Python

```
a = [1, 2, 3, 4, 5]
b = [1, 2, 3, 4, 5]
c = a

print(a is c)
print(a is b)

print(a is not c)
print(a is not b)
```

OUTPUT

True
False
False
True

Operators Precedence

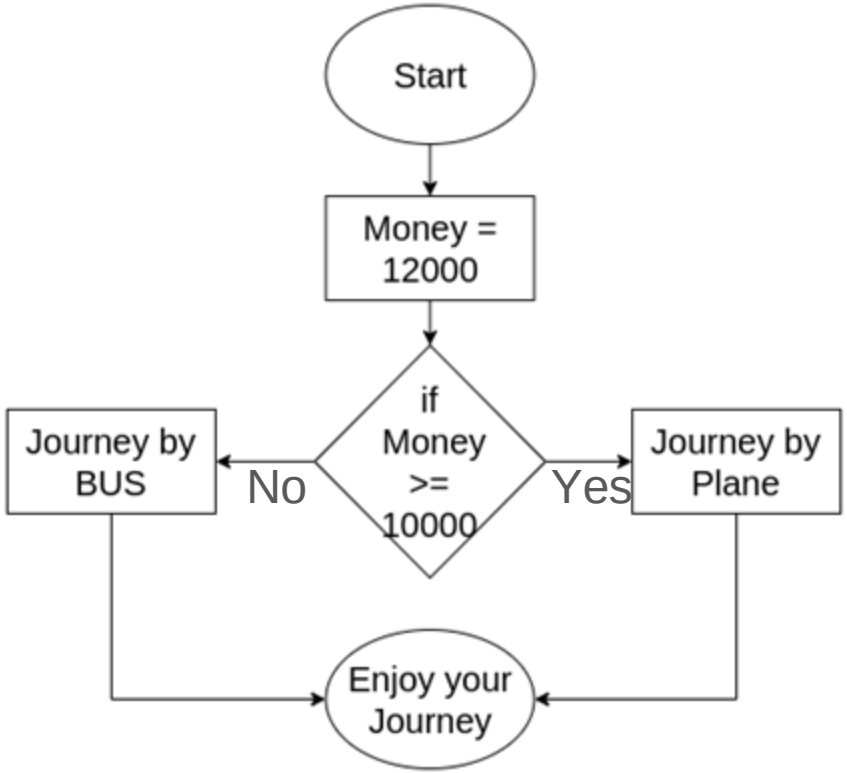
The following table lists all operators from highest precedence to lowest.

Sr.No.	Operator & Description
1	** Exponentiation (raise to the power)
2	~ + - Complement, unary plus and minus (method names for the last two are +@ and -@)
3	* / % // Multiply, divide, modulo and floor division
4	+ - Addition and subtraction
5	>> << Right and left bitwise shift
6	& Bitwise 'AND'
7	^ | Bitwise exclusive `OR' and regular `OR'
8	<= < > >= Comparison operators
9	<> == != Equality operators
10	= %= /= //= -= += *= **= Assignment operators
11	is is not Identity operators
12	in not in Membership operators
13	not or and Logical operators

What is Decision Making?

Decision making means what should we do in a circumstances.

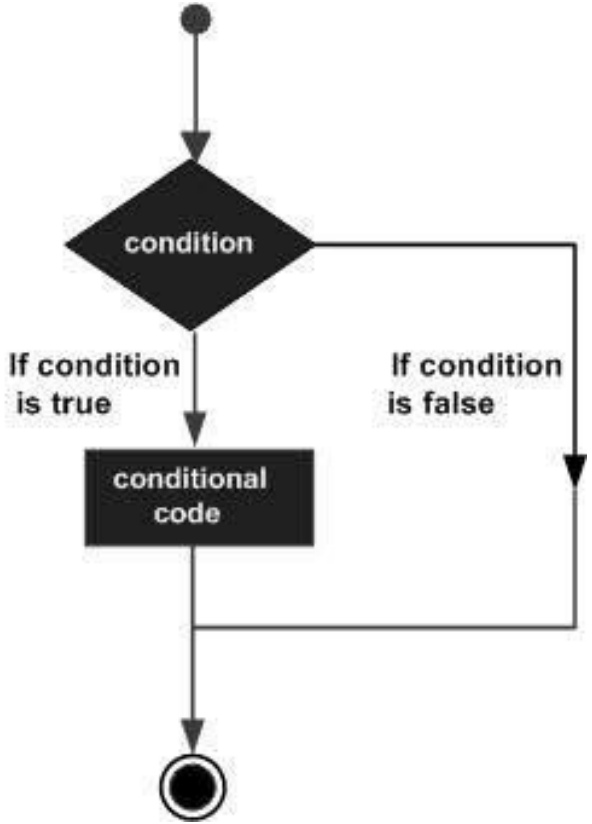
The flowchart helps you to decide what should you choose in the time of travel.



Python Decision Making (if Statement)

if Statement Syntax

```
if expression:
    statement(s)
```



Python Conditions

- Python supports the usual logical conditions from mathematics:
  - Equals: `a == b`
  - Not Equals: `a != b`
  - Less than: `a < b`
  - Less than or equal to: `a <= b`
  - Greater than: `a > b`
  - Greater than or equal to: `a >= b`
- An "if statement" is written by using the **if** keyword.
- For two conditions we use **if** and **else**
- For more than two conditions we use **if… elif…. else**

Python Conditions (Examples)

- See the following code

```
a = 33
b = 200
if b > a:
  print("b is greater than a")
```

- Indentation:
  - Python relies on indentation (whitespace at the beginning of a line) to define scope in the code.
  - Other programming languages often use curly-brackets for this purpose.

Examples: If-elif-else

- **If-elif**

```
a = 33
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
```

- **If-elif-else**

```
a = 200
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
else:
  print("a is greater than b")
```

Exercises:

Calculate Grade based on a given score.
[*Note: score>=80→ A+, score>=75 → A, score>=70→ A- and score >=60 →B unless Failed*]

```
score = 83

if score >= 80:
    print("Grade: A+")
elif score >= 75:
    print("Grade: A")
elif score >= 70:
    print("Grade: A-")
elif score >= 60:
    print("Grade: B")
else:
    print("Grade: F")
```

Exercises:

Let's consider an example of a customer entitled to 10% discount if his purchase amount is > 1000. if not, then no discount is applicable. Given his purchase amount, find out the payable amount

```
purchaseAmount = 1250
payableAmount = purchaseAmount
if(purchaseAmount>1000):
   payableAmount -= purchaseAmount*0.1

print(payableAmount)
```

Exercises:

You can vote if your age is more or equal to 18. Now, given an age, find if s/he can vote or not?

```
age = 20
if age >= 18:
   print("s/he can vote")
else:
   print("s/he can't vote")
```

Exercises:

You are given 3 numbers. Find out the largest number.

```
a,b,c = map(int,input().split(" "))

if(a>=b and a>=c):
   print(a)
elif(b>=c ):
   print(b)
else:
   print(c)
```

Ternary Operators

- A ternary operator in Python, also known as a **conditional expression**, is a way to write a simple if-else statement in a **single line**.
- It's a concise way to perform conditional assignments or return values based on a condition.

value_if_true if **condition** else **value_if_false**

- **condition**: The condition that is evaluated.
- **value_if_true**: The value that is returned if the condition is True.
- **value_if_false**: The value that is returned if the condition is False.

```
age = 18
status = "Adult" if age >= 18 else "Minor"
print(status)  # Output: Adult
```

EXERCISE

Exercise – 4.1

You are given 3 numbers. Find out the largest number.

Sample Input: 4 2 7
Sample Output: 7

Bonus-1: What if - you are given 5 numbers?
Bonus-2: What if - you are given 10 numbers?

Exercise – 4.1 (ans)

```
# a,b,c = map(int,input().split(" "))
a = int(input())
b = int(input())
c = int(input())

if(a>=b and a>=c):
   print(a)
elif(b>=c):
   print(b)
else:
   print(c)
```

Exercise – 4.2

Given a year, Find it is Leap-Year or not.

Sample Input: 2000
Sample Output: “Leap Year”

Sample Input: 2023
Sample Output: “Not Leap Year”

Exercise – 4.2 (ans)

```
year = int(input())
flag = 0 # 1 means leap year, otherwise 0

if year % 400 == 0:
   flag = 1
elif year%100 != 0 and year%4 == 0:
   flag = 1

if(flag):
   print("Leap Year")
else:
   print("Not Leap Year")
```

Resources

- https://www.tutorialspoint.com/python/index.htm
- https://www.w3resource.com/python/python-tutorial.php
- https://www.w3resource.com/python-exercises/string/
- https://www.w3schools.com/python/
- https://www.geeksforgeeks.org/python-programming-language/
- https://youtu.be/t2_Q2BRzeEE?si=OO6J_YNCZykedqsT
- https://realpython.com/
- Head First Python, 3rd Edition by Paul Barry
- Automate the Boring Stuff with Python By Al Sweigart.

Thank You