Assignment No. 1

Q1. Why do we call Python as a general purpose and high-level programming language?

Using python we can access different type of python libraries which help us to reduce work of coding and having predefined function help in coding.

Python is an object-oriented, high-level programming language. Object-oriented means this language is based around objects (such as data) rather than functions, and high-level means it's easy for humans to understand.

Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems. This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today.

Q2. Why is Python called a dynamically typed language?

Python variable assignment is different from some of the popular languages like c, c++ and java. There is no declaration of a variable, just an assignment statement.

Let us see why?
When we declare a variable in C or alike languages, this sets aside an area of memory for holding values allowed by the data type of the variable. The memory allocated will be interpreted as the data type suggests. If it's an integer variable the memory allocated will be read as an integer and so on. When we assign or initialize it with some value, that value will get stored at that memory location. At compile time, initial value or assigned value will be checked. So we cannot mix types. Example: initializing a string value to an int variable is not allowed and the program will not compile.

But Python is a dynamically typed language. It doesn't know about the type of the variable until the code is run. So declaration is of no use. What it does is, It stores that value at some memory location and then binds that variable name to that memory container. And makes the contents of the container accessible through that variable name. So the data type does not matter. As it will get to know the type of the value at run-time.

Q3. List some pros and cons of Python programming language?

Pros for python are:

Python is beginner-friendly, large community, flexible-extensible, extensive libraries, embeddable, highly scalable, IOT opportunities, portable

Cons for Python are:

Mutithreading in Python, python memory consumption and garbage collection, python is dynamically typed, slower than other languages like c++, java, etc

Q4. In what all domains can we use Python?

Domain under python are: all domain related to Data science, machine learning, application development, desktop GUI.

Q5. What are variable and how can we declare them?

- Declaration of a variable in a computer programming language is a statement used to specify the variable name and its data type. Declaration tells the compiler about the existence of an entity in the program and its location. When you declare a variable, you should also initialize it.
- Initialization is the process of assigning a value to the Variable. Every programming language has its own method of initializing the variable. If the value is not assigned to the Variable, then the process is only called a Declaration.

Basic Syntax

The basic form of declaring a variable is:

type identifier [= value] [, identifier [= value]]…];

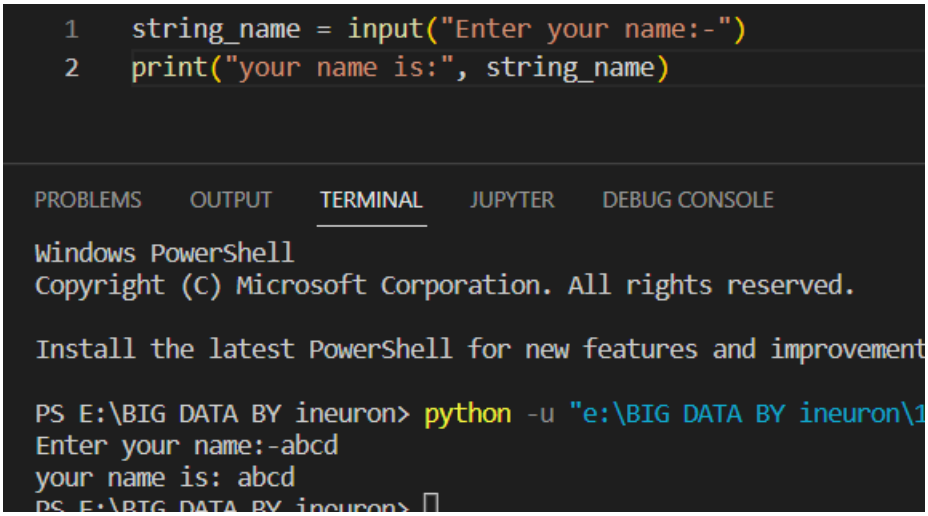
OR

data_type variable_name = value;

where,
type = Data type of the variable
identifier = Variable name
value = Data to be stored in the variable (Optional field)
Note 1: The Data type and the Value used to store in the Variable must match.
Note 2: All declaration statements must end with a semi-colon (;)

Q6. How can we take an input from the user in Python?

In Python, Using the input() function, we take input from a user, and using the print() function, we display output on the screen. Using the input() function, users can give any information to the application in the strings or numbers format.



Q7. What is the default datatype of the value that has been taken as an input using input() function?

Python input() function is used to take user input. By default, it returns the user input in form of a string.By default input() function helps in taking user input as stirng.If any user wants to take input as int or float, we just need to typecast it.

```
# Taking input as string
color = input("What color is rose?: ")
print(color)
# Taking input as int
# Typecasting to int
n = int(input("How many roses?: "))
print(n)
# Taking input as float
# Typecasting to float
price = float(input("Price of each rose?: "))
print(price)
```

Output:

What color is rose?: red

red

How many roses?: 10

10

Price of each rose?: 15.50

15.5

Q8. What is type casting?

Type Casting is the method to convert the variable data type into a certain data type in order to the operation required to be performed by users. In this article, we will see the various technique for typecasting.

There can be two types of Type Casting in Python –

- Implicit Type Casting
- Explicit Type Casting

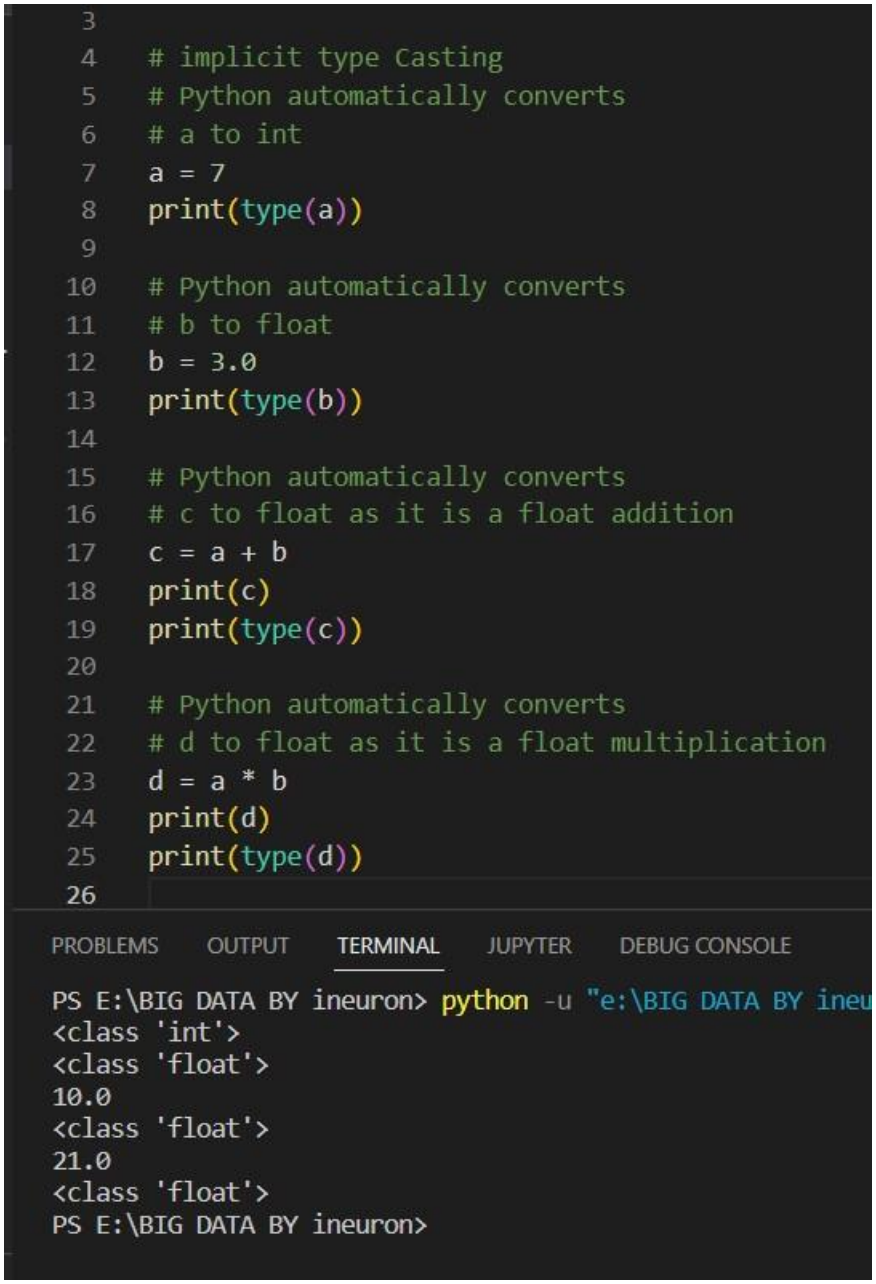
Implicit Type Conversion

In this, methods, Python converts data type into another data type automatically. In this process, users don't have to involve in this process.

Explicit Type Casting

In this method, Python need user involvement to convert the variable data type into certain data type in order to the operation required.

Mainly in type casting can be done with these data type function:



```
3
4   # implicit type Casting
5   # Python automatically converts
6   # a to int
7   a = 7
8   print(type(a))
9
10  # Python automatically converts
11  # b to float
12  b = 3.0
13  print(type(b))
14
15  # Python automatically converts
16  # c to float as it is a float addition
17  c = a + b
18  print(c)
19  print(type(c))
20
21  # Python automatically converts
22  # d to float as it is a float multiplication
23  d = a * b
24  print(d)
25  print(type(d))
26

PROBLEMS  OUTPUT  TERMINAL  JUPYTER  DEBUG CONSOLE

PS E:\BIG DATA BY ineuron> python -u "e:\BIG DATA BY ineu
<class 'int'>
<class 'float'>
10.0
<class 'float'>
21.0
<class 'float'>
PS E:\BIG DATA BY ineuron>
```

Int() : Int() function take float or string as an argument and return int type object.
float() : float() function take int or string as an argument and return float type object.
str() : str() function take float or int as an argument and return string type object.

Let's see some example of type casting:

Type Casting int to float:
Here, we are casting integer object to float object with **float()** function.

```
29   # int variable
30   a = 5
31
32   # typecast to float
33   n = float(a)
34
35   print(n)
36   print(type(n))
37
```

```
PROBLEMS   OUTPUT   TERMINAL   JUPYTER

Windows PowerShell
Copyright (C) Microsoft Corporation.

Install the latest PowerShell for new

PS E:\BIG DATA BY ineuron> python -u
5.0
<class 'float'>
PS E:\BIG DATA BY ineuron>
```

Q9. Can we take more than one input from the user using single input() function? If yes, how? If no, why?

Yes, we can definetly take more than one input from user using single input function:

```
48   x = input("Enter any number: ")
49   print("These are total number of student:", x)
50   print("these are total number of employee", x)
51   print("these are total number products:", x)
52
```

```
PROBLEMS   OUTPUT   TERMINAL   JUPYTER   DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improv

PS E:\BIG DATA BY ineuron> python -u "e:\BIG DATA BY ineu
Enter any number: 85
These are total number of student: 85
these are total number of employee 85
these are total number products: 85
PS E:\BIG DATA BY ineuron>
```

```
41  # taking three inputs at a time
42  x, y, z = input("Enter three values: ").split()
43  print("Total number of employee: ", x)
44  print("Number of male employeeis : ", y)
45  print("Number of female employeeis : ", z)
46
47
```

```
PROBLEMS  OUTPUT  TERMINAL  JUPYTER  DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improveme

PS E:\BIG DATA BY ineuron> python -u "e:\BIG DATA BY ineuron
Enter three values: 44 55 68
Total number of employee:  44
Number of male employeeis :  55
Number of female employeeis :  68
PS E:\BIG DATA BY ineuron>
```

Q10. What are keywords?

List of keyword in python:

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

Q11. Can we use keywords as a variable? Support your answer with reason.

No, we cannot use reserved keyword to declare an variable
The compiler has to be able to distinguish between keywords and identifiers (variable names, function names, etc.), so that it can determine:

* Whether or not your code is following the syntax and semantic rules of the language.
* What code to generate, assuming you are following the syntax and semantic rules of the

```
Q12. What is indentation? What's the use of indentaion in Python?
```

Indentation refers to the spaces at the beginning of a code line.

Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important.

Python uses indentation to indicate a block of code.

```
Q13. How can we throw some output in Python?
```

The basic way to do output is the **print statement**. To end the printed line with a newline, add a print statement without any objects. This will print to any object that implements write(), which includes file objects.

```
Q14. What are operators in Python?
```

Types of Python Operators

Here's a list of different types of Python operators that we will learn in this tutorial.

1. Arithmetic operators (+, - , *, etc)
2. Assignment Operators (*=. *=, /=, etc)
3. Comparison Operators (<, >, <=, >=, etc)
4. Logical Operators (and {&&}, or {||}, not {!=})
5. Bitwise Operators (&, ^, >>, etc)

```
Q15. What is difference between / and // operators?
```

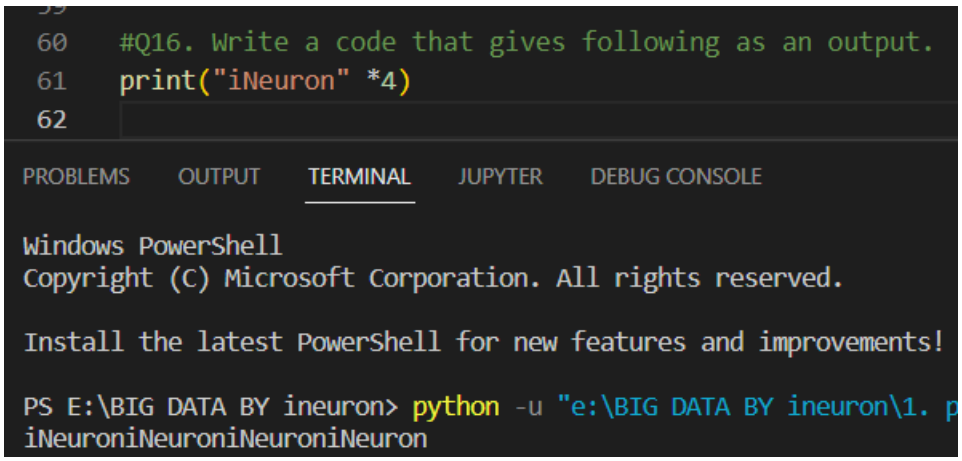
Python supports different types of operators:
They are arithmetic operators, logical operators, assignment operators, etc.
'/' and '//' belong to the arithmetic operators.
'/' is used for the normal division of two numbers.
'//' is used to obtain the smallest integer nearest to the quotient obtained by dividing two numbers.
Let us see an example to understand this.
x = 15

```
y = 3
print(x / y)   #This prints output as 5
print(x // y)  #This prints output as 5
a = 19
b = 4
print(a // b)  #This prints output as 4
print(a / b)  #This prints output as 4.75
```

So, if the quotient obtained by dividing two numbers is not an integer, then operators '/' and '//' will return different answers.

'/' is the division operator. '//' is the floor division operator.

```
Q16. Write a code that gives following as an output.
```



Q17. Write a code to take a number as an input from the user and check if the number is odd or even.

```
65  #Q17. Write a code to take a number as an i
66  a = int(input("Enter any number:"))
67  if (a % 2) ==0:
68      print("Your number is even")
69  else :
70      print("Your number is odd")
```

```
PROBLEMS  OUTPUT  TERMINAL  JUPYTER  DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights rese

Install the latest PowerShell for new features and i

PS E:\BIG DATA BY ineuron> python -u "e:\BIG DATA BY
Enter any number:355
Your number is odd
PS E:\BIG DATA BY ineuron>
```

```
PS E:\BIG DATA BY ineuron> pyth
Enter any number:122
Your number is even
PS E:\BIG DATA BY ineuron>
```

Q18. What are boolean operator?

Boolean is **type of value that can be either True or False** . In Python, the Boolean type is bool , which is a subtype of int . Boolean values are the values True or False (with a capital T and F) in Python. A Boolean variable is a variable that can be either True or False.

Q19. What will the output of the following?
```
1 or 0

0 and 0

True and False and True

1 or 0 or 0
```

```
134   a = 1
135   b = 0
136   if(a or b):
137
138       print("True")
139
140   else:
141
142       print("False")
```

```
PROBLEMS   OUTPUT   TERMINAL   JUPYTER   DEB

Windows PowerShell
Copyright (C) Microsoft Corporation. All r

Install the latest PowerShell for new feat

PS E:\BIG DATA BY ineuron> python -u "e:\B
True
```

```
134   a = 0
135   b = 0
136   if(a and b):
137
138       print("True")
139
140   else:
141
142       print("False")
```

```
PROBLEMS   OUTPUT   TERMINAL   JUPYTER

Windows PowerShell
Copyright (C) Microsoft Corporation. Al

Install the latest PowerShell for new f
://aka.ms/PSWindows

PS E:\BIG DATA BY ineuron> python -u "e
 basics\BigDataQuestions-main\basic.py"
False
```

```
134   a = True
135   b = False
136   if(a and b and a):
137
138       print("True")
139
140   else:
141
142       print("False")
```

```
PROBLEMS   OUTPUT   TERMINAL   JUPYTER

Windows PowerShell
Copyright (C) Microsoft Corporation. Al

Install the latest PowerShell for new f
://aka.ms/PSWindows

PS E:\BIG DATA BY ineuron> python -u "e
 basics\BigDataQuestions-main\basic.py"
False
```

```
134   a = 1
135   b = 0
136   if(a or b or a):
137
138       print("True")
139
140   else:
141
142       print("False")
```

```
PROBLEMS   OUTPUT   TERMINAL   JUPYTER

Windows PowerShell
Copyright (C) Microsoft Corporation. Al

Install the latest PowerShell for new f
://aka.ms/PSWindows

PS E:\BIG DATA BY ineuron> python -u "e
 basics\BigDataQuestions-main\basic.py"
True
```

Q20. What are conditional statements in Python?

The if statement is a conditional statement in python, that is used to determine whether a block of code will be executed or not. Meaning if the program finds the condition defined in the if statement to be true, it will go ahead and execute the code block inside the if statement.

```
Q21. What is use of 'if', 'elif' and 'else' keywords?
```

Decision making is required when we want to execute a code only if a certain condition is satisfied.

The `if…elif…else` statement is used in Python for decision making.

Python if Statement Syntax

Here, the program evaluates the `test expression` and will execute statement(s) only if the test expression is `True`.

If the test expression is `False`, the statement(s) is not executed.

In Python, the body of the `if` statement is indicated by the indentation. The body starts with an indentation and the first unindented line marks the end.

Python interprets non-zero values as `True`. `None` and `0` are interpreted as `False`.

Python if Statement Flowchart

Flowchart of if statement in Python programming

In the above example, `num > 0` is the test expression.

The body of `if` is executed only if this evaluates to `True`.

When the variable `num` is equal to 3, test expression is true and statements inside the body of `if` are executed.

If the variable `num` is equal to -1, test expression is false and statements inside the body of `if` are skipped.

The `print()` statement falls outside of the `if` block (unindented). Hence, it is executed regardless of the test expression.

Python if...else Statement

The `if..else` statement evaluates `test expression` and will execute the body of `if` only when the test condition is `True`.

If the condition is `False`, the body of `else` is executed. Indentation is used to separate the blocks.

Python if..else Flowchart

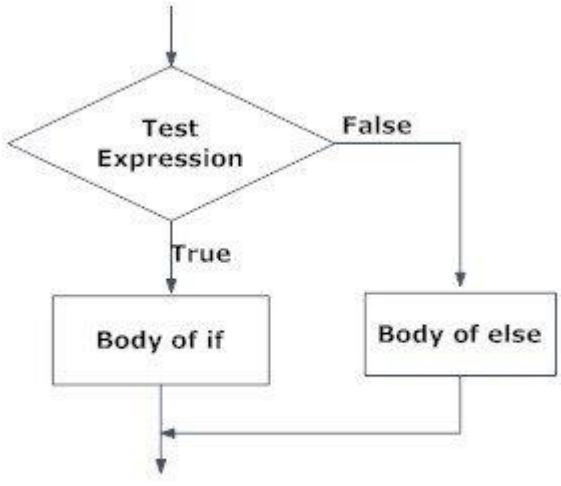


Fig: Operation of if...else statement

Flowchart of if...else statement in Python

In the above example, when num is equal to 3, the test expression is true and the body of if is executed and the body of else is skipped.

If num is equal to -5, the test expression is false and the body of else is executed and the body of if is skipped.

If num is equal to 0, the test expression is true and body of if is executed and body of else is skipped.

Python if...elif...else Statement

Syntax of if...elif...else

The elif is short for else if. It allows us to check for multiple expressions.

If the condition for if is False, it checks the condition of the next elif block and so on.

If all the conditions are False, the body of else is executed.

Only one block among the several if...elif...else blocks is executed according to the condition.

The if block can have only one else block. But it can have multiple elif blocks.

Flowchart of if...elif...else

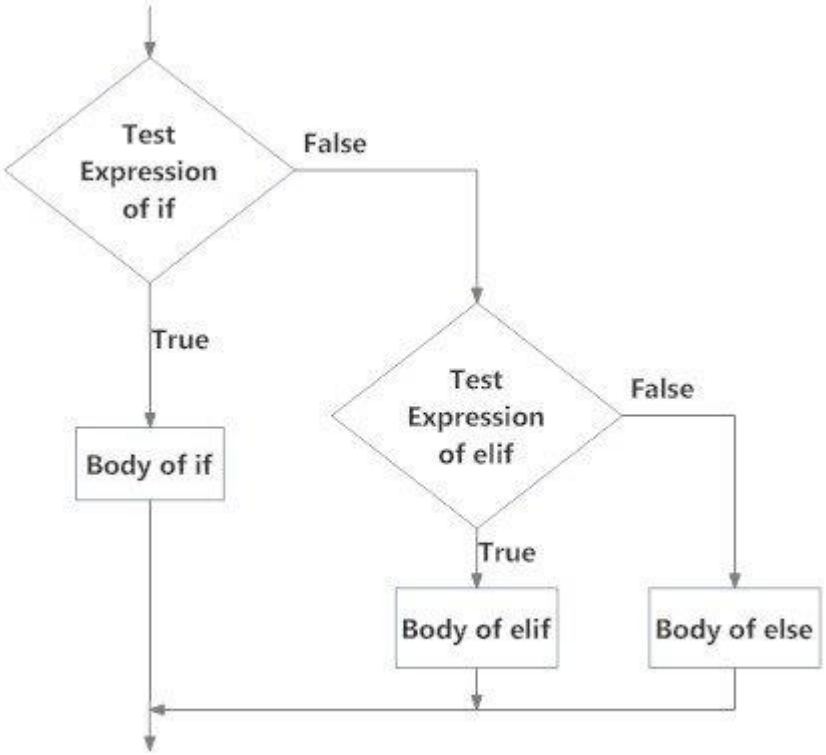


Fig: Operation of if...elif...else statement

Flowchart of if...elif....else statement in Python

Example of if...elif...else

```
'''In this program,
we check if the number is positive or
negative or zero and
display an appropriate message'''

num = 3.4

# Try these two variations as well:
# num = 0
# num = -4.5

if num > 0:
    print("Positive number")
elif num == 0:
    print("Zero")
else:
    print("Negative number")
```

Run Code

When variable `num` is positive, `Positive number` is printed.

If `num` is equal to 0, `Zero` is printed.

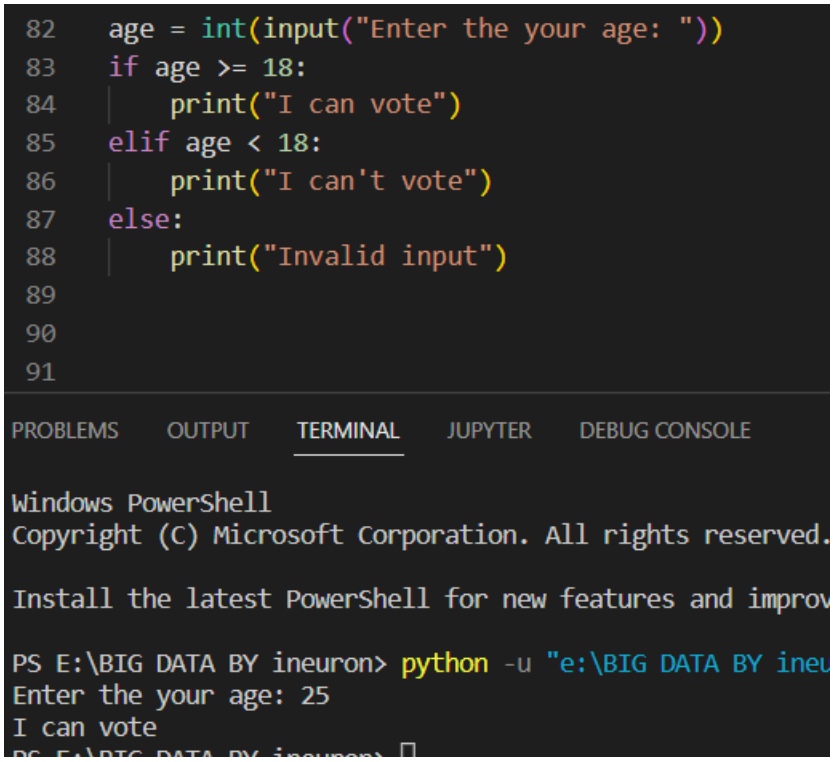
If `num` is negative, `Negative number` is printed.

Python Nested if statements

We can have a `if...elif...else` statement inside another `if...elif...else` statement. This is called nesting in computer programming.

Any number of these statements can be nested inside one another. Indentation is the only way to figure out the level of nesting. They can get confusing, so they must be avoided unless necessary.

```
Q22. Write a code to take the age of person as an input and if age >= 18
display "I can vote". If age is < 18 display "I can't vote".
```



```
PS E:\BIG DATA BY ineuron> py
Enter the your age: 14
I can't vote
PS E:\BIG DATA BY ineuron>
```

Q23. Write a code that displays the sum of all the even numbers from the given list.

```
numbers = [12, 75, 150, 180, 145, 525, 50]
```

```
90  numbers = [12, 75, 150, 180, 145, 525, 50]
91  sum = 0;
92
93  for i in range(0, len(numbers)):
94      sum = sum + numbers[i];
95
96  print("Sum of all the elements of an array:
97
98
```

PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reser

Install the latest PowerShell for new features and im

PS E:\BIG DATA BY ineuron> python -u "e:\BIG DATA BY
Sum of all the elements of an array: 1137
PS E:\BIG DATA BY ineuron>
```

Q24. Write a code to take 3 numbers as an input from the user and display the greatest no as output.

```
 99  a = int(input('Enter first number  : '))
100  b = int(input('Enter second number : '))
101  c = int(input('Enter third number  : '))
102
103  largest = 0
104
105  if a > b and a > c:
106      largest = a
107  if b > a and b > c:
108      largest = b
109  if c > a and c > b:
110      largest = c
111
112  print(largest, "is the largest of three numbers.")
113
```

```
PROBLEMS  OUTPUT  TERMINAL  JUPYTER  DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvement

PS E:\BIG DATA BY ineuron> python -u "e:\BIG DATA BY ineuron\1
Enter first number  : 35
Enter second number : 88
Enter third number  : 76
88 is the largest of three numbers.
PS E:\BIG DATA BY ineuron>
```

Q25. Write a program to display only those numbers from a list that satisfy the following conditions

- The number must be divisible by five

- If the number is greater than 150, then skip it and move to the next number

- If the number is greater than 500, then stop the loop

```
numbers = [12, 75, 150, 180, 145, 525, 50]
```