

Q. Python Arithmetic operators

A screenshot of the Visual Studio Code interface. The left sidebar shows two files: 'one.py' and 'two.py'. The 'one.py' file is open and contains the following Python code:

```
a=46
b=4
print("For a=",a,"and b=",b,"Calculate the following")
#Printing different results
print('1. Addition of two numbers:a+b',a+b)
print('2. Subtraction of two numbers:a-b',a-b)
print('3. Multiplication of two numbers:a*b',a*b)
print('4. Division of two numbers:a/b',a/b)
print('5. Floor Division of two numbers:a//b',a//b)
print('6. Remainder of two numbers:a mod b',a%b)
print('7. Exponent of two numbers:a^b',a**b)
```

The 'OUTPUT' tab is selected, showing the output of running the script:

```
[Running] python -u "c:\Users\Administrator\Desktop\python\one.py"
For a= 46 and b= 4
Calculate the following
1. Addition of two numbers:a+b 50
2. Subtraction of two numbers:a-b 42
3. Multiplication of two numbers:a*b 184
4. Division of two numbers:a/b 11.5
5. Floor Division of two numbers:a//b 11
6. Remainder of two numbers:a mod b 2
7. Exponent of two numbers:a^b 4477456
```

The status bar at the bottom right shows the date and time: 05-02-2025 15:28.

Q. Python Comparator Operators

A screenshot of the Visual Studio Code interface. The left sidebar shows two files: 'one.py' and 'two.py'. The 'two.py' file is open and contains the following Python code:

```
a=46
b=4
print("For a=",a,"and b=",b,"Check the following:")
#Printing different results
print('1. Two numbers are equal or not:',a==b)
print('2. Two numbers are not equal or not:',a!=b)
print('3. a is less than or equal to b:',a<=b)
print('4. a is greater than or equal to b:',a>=b)
print('5. a is greater than b:',a>b)
print('6. a is less than b:',a<b)
```

The 'OUTPUT' tab is selected, showing the output of running the script:

```
[Running] python -u "c:\Users\Administrator\Desktop\python\two.py"
For a= 46 and b= 4
Check the following:
1. Two numbers are equal or not: False
2. Two numbers are not equal or not: True
3. a is less than or equal to b: False
4. a is greater than or equal to b: True
5. a is greater than b: True
6. a is less than b: False
```

The status bar at the bottom right shows the date and time: 05-02-2025 15:31.

Q. Python Assignment Operators

A screenshot of a Python code editor interface. The top bar shows the title "python [Administrator]". The left sidebar has icons for file operations like Open, Save, Find, and Run. The main area shows three files: one.py, two.py, and three.py. The three.py file contains the following code:

```
a=34  
b=6  
  
#Printing the different results  
print('a+=b:',a+b)  
print('a-=b:',a-b)  
print('a*=b:',a*b)  
print('a/=b:',a/b)  
print('a**=b:',a**b)  
print('a//=b:',a//b)
```

The bottom status bar shows "Ln 11, Col 20" and other settings like "Spaces: 4", "UTF-8", "CRLF", "Python", and "Go Live".

Q. Python Bitwise Operators

A screenshot of a Python code editor interface. The top bar shows the title "python [Administrator]". The left sidebar has icons for file operations like Open, Save, Find, and Run. The four.py file contains the following code:

```
a = 7  
b = 8  
  
# printing different results  
print('a & b :', a & b)  
print('a | b :', a | b)  
print('a ^ b :', a ^ b)  
print('~a :', ~a)  
print('a << b :', a << b)  
print('a >> b :', a >> b)
```

The bottom status bar shows "Ln 11, Col 16" and other settings like "Spaces: 4", "UTF-8", "CRLF", "Python", and "Go Live".

Q. Python Logical Operators

A screenshot of a Windows desktop environment. In the center is a Microsoft Visual Studio Code window titled "python [Administrator]". The code editor displays a file named "five.py" with the following content:

```
a = 7

# printing different results
print("For a = 7, checking whether the following conditions are True or False:")
print('"\a > 5 and a < 7"\ =>', a > 5 and a < 7)
print('"\a > 5 or a < 7"\ =>', a > 5 or a < 7)
print('"\not (a > 5 and a < 7)"\ =>', not(a > 5 and a < 7))

[Done] exited with code=0 in 0.105 seconds
```

The output pane shows the execution results:

```
[Running] python -u "c:\Users\Administrator\Desktop\python\five.py"
For a = 7, checking whether the following conditions are True or False:
"\a > 5 and a < 7" => False
"\a > 5 or a < 7" => True
"\not (a > 5 and a < 7)" => True

[Done] exited with code=0 in 0.101 seconds
```

The status bar at the bottom right indicates the date and time: "05-02-2025 15:35".

Q. Python Membership Operators

A screenshot of a Windows desktop environment. In the center is a Microsoft Visual Studio Code window titled "python [Administrator]". The code editor displays a file named "six.py" with the following content:

```
myList = [12, 22, 28, 35, 42, 49, 54, 65, 92, 103, 245, 874]
x = 31
y = 28

# printing the given list
print("Given List:", myList)

# checking if x is present in the list or not
if (x not in myList):
    print("x =", x, "is NOT present in the given list.")
else:
    print("x =", x, "is present in the given list.")

# checking if y is present in the list or not
if (y in myList):
    print("y =", y, "is present in the given list.")
else:
    print("y =", y, "is NOT present in the given list.")

[Done] exited with code=0 in 0.098 seconds
```

The output pane shows the execution results:

```
[Running] python -u "c:\Users\Administrator\Desktop\python\six.py"
Given List: [12, 22, 28, 35, 42, 49, 54, 65, 92, 103, 245, 874]
x = 31 is NOT present in the given list.
y = 28 is present in the given list.

[Done] exited with code=0 in 0.098 seconds
```

The status bar at the bottom right indicates the date and time: "05-02-2025 15:36".

Q. Python Identity Operators

The screenshot shows a Windows desktop environment with the Visual Studio Code application open. The code editor displays a file named 'seven.py' containing the following Python code:

```
a = ["Rose", "Lotus"]
b = ["Rose", "Lotus"]
c = a

# printing the different results
print("a is c => ", a is c)
print("a is not c => ", a is not c)
print("a is b => ", a is b)
print("a is not b => ", a is not b)
print("a == b => ", a == b)
print("a != b => ", a != b)
```

The 'OUTPUT' tab shows the execution results:

```
[Done] exited with code=0 in 0.098 seconds
[Running] python -u "c:\Users\Administrator\Desktop\python\seven.py"
a is c => True
a is not c => False
a is b => False
a is not b => True
a == b => True
a != b => False
[Done] exited with code=0 in 0.092 seconds
```

The status bar at the bottom right indicates the date and time as 05-02-2025 and 15:37.

Q. Reading a csv file in python

The screenshot shows a Windows desktop environment with the Visual Studio Code application open. The code editor displays a file named 'eight.py' containing the following Python code:

```
import csv
with open("./trial.csv") as csv_file:
    csv_read=csv.reader(csv_file,delimiter=',')
    count_line=0
    for row in csv_read:
        if count_line==0:
            print(f"column names are {', '.join(row)}")
            count_line+=1
        else:
            print(f"\t{row[0]} roll number is: {row[1]} and department is: {row[2]}")
            count_line+=1
    print(f"Processed {count_line} lines.")
```

The 'OUTPUT' tab shows the execution results:

```
Column names are Name,Roll Number,Department
Abdul Samim roll number is: 1 and department is: IPE
Bhargav Kulla roll number is: 2 and department is: DSA
Prabhakar roll number is: 3 and department is: ISE
Parakram roll number is: 4 and department is: CSE
Processed 5 lines.
[Done] exited with code=0 in 0.397 seconds
```

The status bar at the bottom right indicates the date and time as 05-02-2025 and 17:56.

Q. Reversing a string in python using for loop

The screenshot shows a dark-themed instance of Visual Studio Code. In the top navigation bar, the title bar says "python [Administrator]". The left sidebar has a "py" icon and lists several files: two.py, three.py, four.py, five.py, six.py, seven.py, eight.py, 1.py, and trial.csv. The main editor area contains the following Python code:

```
reverse_string > 1.py
1 def reverse_string(str):
2     stri =""
3     for i in str:
4         stri=i+stri
5     return stri
6
7 str="JavaPoint"
8 print("The original string is:",str)
9 print("The reverse string is:",reverse_string(str))
```

Below the editor, the "OUTPUT" tab is selected, showing the terminal output:

```
[Done] exited with code=0 in 0.109 seconds
[Running] python -u "c:\Users\Administrator\Desktop\python\reverse_string\1.py"
The original string is: JavaPoint
The reverse string is: tniopTavaJ
[Done] exited with code=0 in 0.113 seconds
```

The status bar at the bottom right shows "Ln 4, Col 13" and "Python".

Q. Reversing a string in python using while loop

The screenshot shows a dark-themed instance of Visual Studio Code. In the top navigation bar, the title bar says "python [Administrator]". The left sidebar has a "py" icon and lists several files: three.py, four.py, five.py, six.py, seven.py, eight.py, 1.py, and trial.csv. The main editor area contains the following Python code:

```
reverse_string > 2.py
1 # Reversing a string using while loop
2 str="JavaPoint"
3 print("The original string is:",str)
4 reverse_string=""
5 count=len(str)
6 while count>0:
7     reverse_string+=str[count-1]
8     count-=1
9 print("The reversed string using a while loop is:",reverse_string)
```

Below the editor, the "OUTPUT" tab is selected, showing the terminal output:

```
[Done] exited with code=0 in 0.113 seconds
[Running] python -u "c:\Users\Administrator\Desktop\python\reverse_string\2.py"
The original string is: JavaPoint
The reversed string using a while loop is: tniopTavaJ
[Done] exited with code=0 in 0.113 seconds
```

The status bar at the bottom right shows "Ln 9, Col 66" and "Python".

Q. Reversing a string in python using slice operator

The screenshot shows the Visual Studio Code interface with a dark theme. The code editor has several tabs open, but the active tab is '3.py'. The code in '3.py' demonstrates reversing a string using the slice operator:

```
reverse_string > 3.py
1 # Reversing a string using slice operator
2
3 def reverse(str):
4     str=str[::-1]
5     return str
6 s="Javaatpoint"
7 print("The original string is:",s)
8 print("The reversed string using slice is :",reverse(s))
```

The output panel at the bottom shows the execution results:

```
[Done] exited with code=0 in 0.113 seconds
[Running] python -u "c:\Users\Administrator\Desktop\python\reverse_string\3.py"
The original string is: Javaatpoint
The reversed string using slice is : tniopTavaJ
[Done] exited with code=0 in 0.146 seconds
```

The status bar at the bottom right indicates the file is saved, the language is Python, and the current date and time are 05-02-2025.

Q. Reversing a string in python using reverse function with join

The screenshot shows the Visual Studio Code interface with a dark theme. The code editor has several tabs open, but the active tab is '4.py'. The code in '4.py' demonstrates reversing a string using the reverse function with join:

```
reverse_string > 4.py
1 # Reversing a string using reverse function with join
2
3 def reverse(str):
4     string="".join(reversed(str))
5     return string
6 s="Javaatpoint"
7 print("The original string is:",s)
8 print("The reversed string using reversed() is:", reverse(s))
```

The output panel at the bottom shows the execution results:

```
[Done] exited with code=0 in 0.146 seconds
[Running] python -u "c:\Users\Administrator\Desktop\python\reverse_string\4.py"
The original string is: Javaatpoint
The reversed string using reversed() is: tniopTavaJ
[Done] exited with code=0 in 0.116 seconds
```

The status bar at the bottom right indicates the file is saved, the language is Python, and the current date and time is 05-02-2025.

Q. Reversing a string in python using recursion

A screenshot of the Visual Studio Code interface. The top menu bar includes File, Edit, Selection, View, Go, Run, etc. The title bar says "python [Administrator]". The left sidebar shows files: six.py, seven.py, eight.py, 1.py, 2.py, 3.py, 4.py, 5.py (the active file), and trial.csv. The main editor area contains the following Python code:

```
reverse_string > 5.py
1 # Reversing string using recursion
2
3 def reverse(str):
4     if len(str)==0:
5         return str
6     else:
7         return reverse(str[1:]) + str[0]
8 str="Javaatpoint"
9 print("The original string is:",str)
10 print("The reversed string using recursion is:",reverse(str))
```

The bottom status bar shows "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", "TERMINAL", and "PORTS". The terminal tab is active, displaying the output of running the script:

```
[Done] exited with code=0 in 0.116 seconds
[Running] python -u "c:\Users\Administrator\Desktop\python\reverse_string\5.py"
The original string is: Javaatpoint
The reversed string using recursion is: tniopTavaJ
[Done] exited with code=0 in 0.13 seconds
```

The bottom right corner shows "Ln 10, Col 62" and "Python".

Python IF conditions

Q. Basic if conditions

A screenshot of the Visual Studio Code interface. The top menu bar includes File, Edit, Selection, View, Go, Run, etc. The title bar says "python [Administrator]". The left sidebar shows files: 1.py (the active file). The main editor area contains the following Python code:

```
if_else > 1.py
1 num=int(input("Enter the number:"))
2 if num%2==0:
3     print("The given number is an even number")
```

The bottom status bar shows "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", "TERMINAL", and "PORTS". The terminal tab is active, displaying the output of running the script:

```
PS C:\Users\Administrator\Desktop\python\if_else> python 1.py
Enter the number:08634725
PS C:\Users\Administrator\Desktop\python\if_else> python 1.py
Enter the number:034928752
The given number is an even number
PS C:\Users\Administrator\Desktop\python\if_else>
```

The bottom right corner shows "Ln 3, Col 48" and "Python".

Q. Program to print largest of three numbers

The screenshot shows a Microsoft Visual Studio Code interface. The top bar includes 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', and a search bar with 'python [Administrator]'. Below the search bar are icons for file operations like Open, Save, and Close. The main workspace displays two files: '1.py' and '2.py'. The code in '2.py' is as follows:

```
if __name__ == "__main__":
    #Program to print largest of three numbers
    a=int(input("Enter a:"))
    b=int(input("Enter b:"))
    c=int(input("Enter c:"))

    if a>b and a>c:
        print("From the above three numbers given, a is largest")
    if b>a and b>c:
        print("From the above three numbers given, b is largest")
    else:
        print("From the above three numbers gives, c is largest")
```

The bottom navigation bar includes 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab is active, showing command-line output:

- PS C:\Users\Administrator\Desktop\python\if_else> python .\2.py
- Enter a:12
- Enter b:32
- Enter c:21
- From the above three numbers given, b is largest
- PS C:\Users\Administrator\Desktop\python\if_else>

The status bar at the bottom shows 'Ln 12, Col 60', 'Spaces: 4', 'UTF-8', 'CRLF', 'Python', and 'Go Live'. The bottom right corner shows the date and time: '06-02-2025 10:47'.

Q. Program to check eligibility of voters

The screenshot shows a Microsoft Visual Studio Code interface. The top bar includes File, Edit, Selection, View, Go, Run, and a search bar for "python [Administrator]". The left sidebar has icons for file operations like Open, Save, Find, and Copy/Paste. The main editor window displays a Python script named 3.py:

```
if_else > 3.py
1 # Program to check whether a person is eligible to vote or not
2
3 age=int(input("Enter your age:"))
4 if age>=18:
5     print("You are eligible to vote!")
6 else:
7     print("Sorry! You have to wait!")
```

The bottom navigation bar includes PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected), and PORTS. The terminal window shows the output of running the script:

- PS C:\Users\Administrator\Desktop\python\if_else> python .\3.py
- Enter your age:22
- You are eligible to vote!
- PS C:\Users\Administrator\Desktop\python\if_else> python .\3.py
- Enter your age:12
- Sorry! You have to wait!
- PS C:\Users\Administrator\Desktop\python\if_else>

The status bar at the bottom shows Ln 6, Col 6, Spaces: 4, UTF-8, CRLF, Python, and Go Live. The taskbar at the very bottom includes icons for File Explorer, Task View, Edge, File Explorer, Mail, Google Chrome, Mozilla Firefox, and VS Code.

Q. Program to check odd/even number

A screenshot of a Windows desktop environment. At the top, there's a taskbar with icons for File Explorer, Edge browser, File Explorer, Task View, and Start. Below the taskbar is a dark-themed code editor window titled "python [Administrator]". The code editor shows a file named "4.py" with the following content:

```
if _else > 4.py
1 # Simple Python Program to check whether a number is even or not
2
3 num=int(input("Enter the number:"))
4
5 if num%2==0:
6     print("The given number is an even number:")
7 else:
8     print("The given number is an odd number:")
```

Below the code editor is a terminal window titled "powershell - if_else" which displays the output of running the script:

```
PS C:\Users\Administrator\Desktop\python\if_else> python .\4.py
Enter the number:1234565
The given number is an odd number:
● PS C:\Users\Administrator\Desktop\python\if_else> python .\4.py
Enter the number:93782
The given number is an even number:
○ PS C:\Users\Administrator\Desktop\python\if_else>
```

The bottom status bar shows "Ln 8, Col 46" and "Python".

Q. Elif basic program

A screenshot of a Windows desktop environment. At the top, there's a taskbar with icons for File Explorer, Edge browser, File Explorer, Task View, and Start. Below the taskbar is a dark-themed code editor window titled "python [Administrator]". The code editor shows a file named "5.py" with the following content:

```
if _else > 5.py
1 # Simple Python program to understand elif statement
2
3 num=int(input("Enter the number:"))
4
5 if num==10:
6     print("The given number is equals to 10")
7 elif num==50:
8     print("The given number is equal to 50")
9 elif num==100:
10    print("The given number is equal to 100")
11 else:
12     print("The given number is not equal to 10,50 or 100")
```

Below the code editor is a terminal window titled "powershell - if_else" which displays the output of running the script:

```
PS C:\Users\Administrator\Desktop\python\if_else> python .\5.py
Enter the number:69
The given number is not equal to 10,50 or 100
○ PS C:\Users\Administrator\Desktop\python\if_else>
```

The bottom status bar shows "Ln 11, Col 59" and "Python".

Q.

A screenshot of a Windows desktop environment. At the top is a taskbar with icons for File Explorer, Edge browser, File Explorer, Mail, and Task View. Below the taskbar is a Start button and a search bar. The main area shows a code editor window titled "python [Administrator]" containing a script named "6.py". The script uses an if-else ladder to determine a grade based on marks. Below the editor is a terminal window titled "powershell - if_else" showing the execution of the script and its output. The system tray at the bottom right shows the date and time as 06-02-2025 and 10:58.

```
if_else > 6.py
1 # Simple Python program to understand elif statement
2
3 marks=int(input("enter the marks:"))
4
5 if marks>85 and marks <=100:
6     print("Congrats! You scored grade A...")
7 elif marks > 60 and marks <= 85:
8     print("You scored grade B+...")
9 elif marks>40 and marks<=60:
10    print("You scored grade B...")
11 elif marks>30 and marks<=40:
12    print("You scored grade C...")
13 else:
14     print("You failed!")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\Administrator\Desktop\python\if_else> python .\6.py
enter the marks:99
Congrats! You scored grade A...
○ PS C:\Users\Administrator\Desktop\python\if_else>

Ln 14, Col 23 Spaces: 4 UTF-8 CRLF Python Go Live

Type here to search

10:58 ENG US 06-02-2025

Python Loops

Q. Working of for loop

A screenshot of a Windows desktop environment. At the top is a taskbar with icons for File Explorer, Edge browser, File Explorer, Mail, and Task View. Below the taskbar is a Start button and a search bar. The main area shows a code editor window titled "python [Administrator]" containing a script named "1.py loops". The script uses a for loop to calculate squares of numbers from 1 to 10 and prints the result. Below the editor is a terminal window titled "powershell - loops" showing the execution of the script and its output. The system tray at the bottom right shows the date and time as 06-02-2025 and 11:04.

```
loops > 1.py
1 # Working of for loop
2
3 num=[4,2,6,7,3,5,8,10,6,1,9,2]
4 square=0
5 squares=[]
6
7 for value in num:
8     square=value**2
9     squares.append(square)
10 print("The list of squares is",squares)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\Administrator\Desktop\python\loops> python .\1.py
The list of squares is [16, 4, 36, 49, 9, 25, 64, 100, 36, 1, 81, 4]
○ PS C:\Users\Administrator\Desktop\python\loops>

Ln 10, Col 40 Spaces: 4 UTF-8 CRLF Python Go Live

Type here to search

11:04 ENG US 06-02-2025

Q. If-else in for loops

A screenshot of the Visual Studio Code interface. The top menu bar shows 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', etc. The title bar says 'python [Administrator]'. The left sidebar has icons for file operations like Open, Save, Find, and Refresh. The main editor area contains the following Python code:

```
loops > 2.py
1 # if-else statements in for loops
2
3 string = "Python Loop"
4
5 for s in string:
6     if s=="o":
7         print("if block")
8     else:
9         print(s)
```

The terminal below shows the output of running the script:

```
-a--- 05-02-2025 15:35 305 five.py
PS C:\Users\Administrator\Desktop\python\loops> python .\2.py
P
y
t
h
if block
n

L
if block
if block
p
o PS C:\Users\Administrator\Desktop\python\loops>
```

The status bar at the bottom indicates 'Ln 5, Col 9' and 'Python'.

Q. else in for loop

A screenshot of the Visual Studio Code interface. The top menu bar shows 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', etc. The title bar says 'python [Administrator]'. The left sidebar has icons for file operations like Open, Save, Find, and Refresh. The main editor area contains the following Python code:

```
loops > 3.py
1 # else statement in for loop
2
3 tuple_ = (3,4,6,8,9,2,3,8,9,7)
4 for value in tuple_:
5     if value%2!=0:
6         print(value)
7     else:
8         print("These are the odd numbers present in the tuple")
```

The terminal below shows the output of running the script:

```
PS C:\Users\Administrator\Desktop\python\loops> python .\3.py
3
These are the odd numbers present in the tuple
These are the odd numbers present in the tuple
These are the odd numbers present in the tuple
9
These are the odd numbers present in the tuple
3
These are the odd numbers present in the tuple
9
7
o PS C:\Users\Administrator\Desktop\python\loops>
```

The status bar at the bottom indicates 'Ln 8, Col 62' and 'Python'.

Q. working of range function

A screenshot of a Windows desktop environment. At the top is the taskbar with icons for File Explorer, Edge, File Explorer, Task View, Start, and a pinned Microsoft Edge tab for 'powershell-loops'. Below the taskbar is a dark-themed code editor window titled 'python [Administrator]'. The code in the editor is:

```
loops > 4.py
1 # Working of range function
2
3 print(range(15))
4 print(list(range(15)))
5 print(list(range(4,13)))
6 print(list(range(5,25,4)))
```

The terminal window below shows the output of running the script:

```
PS C:\Users\Administrator\Desktop\python\loops> python .\4.py
range(0, 15)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
[4, 5, 6, 7, 8, 9, 10, 11, 12]
[5, 9, 13, 17, 21]
```

The system tray at the bottom right shows the date as 06-02-2025.

Q. Indexing in for loop

A screenshot of a Windows desktop environment. At the top is the taskbar with icons for File Explorer, Edge, File Explorer, Task View, Start, and a pinned Microsoft Edge tab for 'powershell-loops'. Below the taskbar is a dark-themed code editor window titled 'python [Administrator]'. The code in the editor is:

```
loops > 5.py
1 # indexing in for loop
2
3 tuple_ = ("Python","Loops","Sequence","Condition","Range")
4 for iterator in range(len(tuple_)):
5     print(tuple_[iterator].upper())
```

The terminal window below shows the output of running the script:

```
PS C:\Users\Administrator\Desktop\python\loops> python .\5.py
PYTHON
LOOPS
SEQUENCE
CONDITION
RANGE
```

The system tray at the bottom right shows the date as 06-02-2025.

Q. While loop counter

File Edit Selection View Go Run ... ← → python [Administrator] ⬧ 4.py if_else 5.py if_else 6.py if_else 1.py loops 2.py loops 3.py loops 4.py loops 5.py loops 6.py loops

loops > 6.py

```
1 # While loop counter
2 counter=0
3 while counter<10:
4     counter+=3
5     print("Python Loops")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell - loops +

PS C:\Users\Administrator\Desktop\python\loops> python .\6.py

Python Loops
Python Loops

PS C:\Users\Administrator\Desktop\python\loops>

Q. Else statement in while loop

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The top navigation bar includes File, Edit, Selection, View, Go, Run, and other standard options. A tab bar at the top right shows multiple open files: 5.py_if_else, 6.py_if_else, 1.py_loops, 2.py_loops, 3.py_loops, 4.py_loops, 5.py_loops, 6.py_loops, and 7.py. The main editor area contains the following Python code:

```
loops > 7.py
1 # Else statement with the while loop
2
3 counter=0
4
5 while(counter<10):
6     counter+=3
7     print("Python Loops")
8 else:
9     print("Code block inside the else statement")
```

Below the editor, there are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, displaying command-line output from a PowerShell window titled "powershell - loops". The output shows the execution of the script and its results:

```
PS C:\Users\Administrator\Desktop\python\loops> python .\7.py
Python Loops
Python Loops
Python Loops
Python Loops
Python Loops
Code block inside the else statement
```

The bottom status bar shows the current file is Ln 9, Col 48, with 48 spaces, using UTF-8 encoding, and is a Python file. It also shows the date and time as 06-02-2025, 11:50. The bottom navigation bar includes icons for back, forward, search, and other system functions.

Q. Single statement while block

A screenshot of the Visual Studio Code interface. The top menu bar includes File, Edit, Selection, View, Go, Run, etc. The title bar says "python [Administrator]". The left sidebar shows a file tree with several Python files like 6.py_if_else, 1.py_loops, etc. The main editor window contains the following code:

```
loops > 8.py
1 # Single statement while block
2
3 counter=0
4 while (counter<3):print("Python Loops")
```

The terminal below shows the output of running the script:

```
Python Loops
```

The status bar at the bottom right indicates "Ln 4, Col 40 Spaces: 4 UTF-8 CRLF Python 11:51 06-02-2025".

Q. Working of Continue statement

A screenshot of the Visual Studio Code interface. The top menu bar includes File, Edit, Selection, View, Go, Run, etc. The title bar says "python [Administrator]". The left sidebar shows a file tree with several Python files like 1.py_loops, 2.py_loops, etc. The main editor window contains the following code:

```
loops > 9.py
1 # Working of continue statement
2
3 for string in "Python Loops":
4     if string=="o" or string == "p" or string=="t":
5         continue
6     print("Current letter:",string)
```

The terminal below shows the output of running the script:

```
PS C:\Users\Administrator\Desktop\python\loops> python .\9.py
Current letter: P
Current letter: y
Current letter: h
Current letter: n
Current letter:
Current letter:
Current letter: L
Current letter: s
```

The status bar at the bottom right indicates "Ln 6, Col 35 Spaces: 4 UTF-8 CRLF Python 11:54 06-02-2025".

Q. Working of break statement

A screenshot of the Visual Studio Code interface. The terminal window at the bottom shows the output of running the file 10.py. The code in 10.py is as follows:

```
loops > 10.py
1 # Working of the break statement
2
3 for string in "Python Loops":
4     if string=="L":
5         break
6     print("Current Letter:",string)
```

The terminal output is:

```
PS C:\Users\Administrator\Desktop\python\loops> python .\10.py
Current Letter: P
Current Letter: y
Current Letter: t
Current Letter: h
Current Letter: o
Current Letter: n
Current Letter:
PS C:\Users\Administrator\Desktop\python\loops>
```

The status bar at the bottom right indicates the file is a Python file.

Q. Working of Pass statement

A screenshot of the Visual Studio Code interface. The terminal window at the bottom shows the output of running the file 11.py. The code in 11.py is as follows:

```
loops > 11.py
1 # Working of Pass statement
2
3 for string in "Python Loops":
4     pass
5     print("Last Letter:",string)
```

The terminal output is:

```
PS C:\Users\Administrator\Desktop\python\loops> python .\11.py
Last Letter: s
PS C:\Users\Administrator\Desktop\python\loops>
```

The status bar at the bottom right indicates the file is a Python file.

Q. Sum of squares using for loop

The screenshot shows a dark-themed instance of Visual Studio Code. In the top navigation bar, the file tab is open, showing several Python files: 4.py loops, 5.py loops, 6.py loops, 7.py, 8.py, 9.py, 10.py, 11.py, and 12.py. The active editor tab is labeled "python [Administrator]" and contains the following Python code:

```
loops > 12.py
1 # Sum of squares using for loop
2
3 numbers=[3,5,23,6,5,1,2,9,8]
4 sum_=0
5 for num in numbers:
6     sum_=sum_+num**2
7 print("The sum of squares is:",sum_)
```

Below the editor, the terminal tab is selected, showing the command line output:

```
PS C:\Users\Administrator\Desktop\python\loops> python .\12.py
The sum of squares is: 774
○ PS C:\Users\Administrator\Desktop\python\loops>
```

The status bar at the bottom right indicates the current file is "12.py", the language is "Python", and the date and time are "06-02-2025".

Q. Creating a list using range()

The screenshot shows a dark-themed instance of Visual Studio Code. In the top navigation bar, the file tab is open, showing several Python files: 5.py loops, 6.py loops, 7.py, 8.py, 9.py, 10.py, 11.py, 12.py, and 13.py. The active editor tab is labeled "python [Administrator]" and contains the following Python code:

```
loops > 13.py
1 # Creating a list using range()
2
3 my_list=[3,5,6,8,4]
4 for iter_var in range(len(my_list)):
5     my_list.append(my_list[iter_var]+2)
6 print(my_list)
```

Below the editor, the terminal tab is selected, showing the command line output:

```
● PS C:\Users\Administrator\Desktop\python\loops> python .\13.py
[3, 5, 6, 8, 4, 5, 7, 8, 10, 6]
○ PS C:\Users\Administrator\Desktop\python\loops>
```

The status bar at the bottom right indicates the current file is "13.py", the language is "Python", and the date and time are "06-02-2025".

Q.

A screenshot of a Windows desktop environment. At the top is the taskbar with icons for File Explorer, Start, Task View, and several pinned applications. Below the taskbar is a dark-themed code editor window titled "python [Administrator]". The code in file "14.py" is:

```
loops > 14.py
3     student_name_1 = "Itika"
4     student_name_2 = "Parker"
5
6     records = {"Itika":90,"Arshia":92,"Peter":46}
7     def marks(student_name):
8         for a_student in records:
9             if a_student==student_name:
10                 return records[a_student]
11                 break
12             else:
13                 return f"There is no student of name {student_name} in the records"
14
15     print(f"Marks of {student_name_1} are:", marks(student_name_1))
16     print(f"Marks of {student_name_2} are:",marks(student_name_2))
```

The terminal window below shows the execution of the script:

```
PS C:\Users\Administrator\Desktop\python\loops> python .\14.py
Marks of Itika are: 90
Marks of Parker are: There is no student of name Parker in the records
○ PS C:\Users\Administrator\Desktop\python\loops>
```

Q. Nested loops

A screenshot of a Windows desktop environment. At the top is the taskbar with icons for File Explorer, Start, Task View, and several pinned applications. Below the taskbar is a dark-themed code editor window titled "python [Administrator]". The code in file "15.py" is:

```
loops > 15.py
1     # Nested loops
2     import random
3     numbers=[ ]
4     for val in range(0,11):
5         numbers.append(random.randint(0,11))
6     for num in range(0,11):
7         for i in numbers:
8             if num==i:
9                 print(num,end=" ")
```

The terminal window below shows the execution of the script:

```
PS C:\Users\Administrator\Desktop\python\loops> python 15.py
2 4 4 7 7 8 9 10 10
○ PS C:\Users\Administrator\Desktop\python\loops>
```

Q. While loop with conditions

The screenshot shows a Microsoft Visual Studio Code interface. The top bar has tabs for 'File', 'Edit', 'View', 'Go', 'Run', and '...'. The title bar says 'python [Administrator]'. The left sidebar has icons for file operations like Open, Save, Find, and others. The main editor area shows a Python script named '16.py' with the following code:

```
loops > 16.py
1 # While loop with conditions
2
3 i=1
4 while i<51:
5     if i%5==0 or i%7==0:
6         print(i,end=' ')
7     i+=1
```

Below the editor, there's a navigation bar with 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab is selected, showing a PowerShell window titled 'powershell - loops'. It displays the output of the script execution:

```
PS C:\Users\Administrator\Desktop\python\loops> python .\16.py
5 7 10 14 15 20 21 25 28 30 35 40 42 45 49 50
PS C:\Users\Administrator\Desktop\python\loops>
```

The bottom status bar shows 'Ln 7, Col 9' and other details like 'Spaces: 4', 'UTF-8', 'CRLF', 'Python', and 'Go Live'. The bottom right corner shows the date and time as '06-02-2025 12:13'.

Q. Prime number via while loop

The screenshot shows a Microsoft Visual Studio Code interface. The top bar has tabs for 'File', 'Edit', 'View', 'Go', 'Run', and '...'. The title bar says 'python [Administrator]'. The left sidebar has icons for file operations like Open, Save, Find, and others. The main editor area contains a Python script named 'loops > 17.py'. The code defines a function 'prime_number' that checks if a number is prime using a while loop. It prints out whether each number in a list is prime or not. The terminal below shows the script being run and its output. The bottom status bar shows file paths, line and column numbers, and other settings.

```
File Edit View Go Run ...
python [Administrator]
08 □ □ - □ ... loops > 17.py
1 # Prime number via while loop
2
3 num=[34,12,54,23,75,34,11]
4
5 def prime_number(number):
6     condition=0
7     iteration=2
8     while iteration<=number//2:
9         if number%iteration==0:
10             condition=1
11             break
12         iteration+=1
13     if condition==0:
14         print(f"{number} is a PRIME number")
15     else:
16         print(f"{number} is not a PRIME number")
17 for i in num:
18     prime_number(i)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell - loops + ⌂ ⌂ ... ^ x

● PS C:\Users\Administrator\Desktop\python\loops> python .\17.py
34 is not a PRIME number
12 is not a PRIME number
54 is not a PRIME number
23 is a PRIME number
75 is not a PRIME number
34 is not a PRIME number
11 is a PRIME number
○ PS C:\Users\Administrator\Desktop\python\loops> █

Ln 18, Col 19 Spaces: 4 UTF-8 CRLF Python ⌂ Go Live
ENG US 06-02-2025
Type here to search
```

Q. Armstrong number using while loop

The screenshot shows a Windows terminal window titled "python [Administrator]". The code in the editor is:

```
loops > 18.py
1 # Armstrong number using while loop
2
3 n=int(input())
4 n1=str(n)
5 l=len(n1)
6 temp=n
7 s=0
8 while n1!=0:
9     r=n%10
10    s+=r**l
11    n=n//10
12 if s==temp:
13     print("It is an Armstrong number")
14 else:
15     print("It is not an Armstrong number")
```

The terminal output shows the program running and printing "It is not an Armstrong number" for the input 994327.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Administrator\Desktop\python\loops> python .\18.py
994327
It is not an Armstrong number
PS C:\Users\Administrator\Desktop\python\loops>

Ln 7, Col 1 Spaces: 4 UTF-8 CRLF Python Go Live

Type here to search ENG 12:19 US 06-02-2025

Q. Multiplication tables using while loop

The screenshot shows a Windows terminal window titled "python [Administrator]". The code in the editor is:

```
loops > 19.py
1 # Multiplication tables using while loop
2
3 num=21
4 counter=1
5 print("The Multiplication Table of:",num)
6 while counter<=10:
7     ans=num*counter
8     print(num,'x',counter,'=',ans)
9     counter+=1
```

The terminal output shows the program running and printing the multiplication table for 21 from 1 to 10.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Traceback (most recent call last):
● PS C:\Users\Administrator\Desktop\python\loops> python .\19.py
The Multiplication Table of: 21
21 x 1 = 21
21 x 2 = 42
21 x 3 = 63
21 x 4 = 84
21 x 5 = 105
21 x 6 = 126
21 x 7 = 147
21 x 8 = 168
21 x 9 = 189
● 21 x 10 = 210
○ PS C:\Users\Administrator\Desktop\python\loops>

Ln 9, Col 15 Spaces: 4 UTF-8 CRLF Python Go Live

Type here to search ENG 12:21 US 06-02-2025

Q. Letter check

```
File Edit Selection View Go Run ... ← → python [Administrator] 08 - 12.py 13.py 14.py 15.py 16.py 17.py 18.py 19.py 20.py loops > 20.py
1 # Code to determine the number of letters of every word from the given list
2
3 trial = ['Priya','Neha','Cow','To']
4 index=0
5 while index<len(trial):
6     element=trial[index]
7     print(len(element))
8     index+=1
```

TERMINAL

```
PS C:\Users\Administrator\Desktop\python\loops> python .\20.py
5
4
3
2
```

Q. Multiple conditions using OR operator

The screenshot shows a Visual Studio Code (VS Code) interface. The top bar includes the title "python [Administrator]" and standard file navigation buttons. On the left, there's a vertical toolbar with icons for file operations like Open, Save, Find, and Run. The main workspace contains a code editor with the following Python script:

```
loops > 21.py
1 # Multiple conditions with OR operator
2 num1=17
3 num2=-12
4
5 while num1>5 or num2<-5:
6     num1-=2
7     num2+=3
8     print((num1,num2))
```

Below the code editor is a tab bar with "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", "TERMINAL", and "PORTS". The "TERMINAL" tab is selected, showing the output of running the script:

```
PS C:\Users\Administrator\Desktop\python\loops> python .\21.py
(15, -9)
(13, -6)
(11, -3)
(9, 0)
(7, 3)
• (5, 6)
○ PS C:\Users\Administrator\Desktop\python\loops>
```

The bottom of the screen features a taskbar with the Windows Start button, a search bar, and various system icons. The status bar at the bottom right displays "Ln 8, Col 22" and "Python".

Q. Multiple logical operators in while loop

A screenshot of the Visual Studio Code interface. The top menu bar shows 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', etc. The title bar says 'python [Administrator]'. The left sidebar has icons for file operations like open, save, and close. The main editor area contains a Python script named '22.py' with the following code:

```
loops > 22.py
1 # Multiple logical operators in while loop
2
3 num1=9
4 num=14
5 max_value=4
6 counter=0
7 while(counter<num1 or counter<num2) and not counter>max_value:
8     print(f"Number of iterations:{counter}")
9     counter+=1
```

The terminal below the editor shows the output of running the script:

```
PS C:\Users\Administrator\Desktop\python\loops> python .\22.py
Number of iterations:0
Number of iterations:1
Number of iterations:2
Number of iterations:3
Number of iterations:4
Number of iterations:5
Number of iterations:6
Number of iterations:7
Number of iterations:8
Number of iterations:9
Number of iterations:10
Number of iterations:11
Number of iterations:12
Number of iterations:13
Number of iterations:14
```

The bottom status bar shows 'Ln 9, Col 15 Spaces: 4 UTF-8 CRLF Python Go Live' and the date '06-02-2025'.

Python Strings

Q. Creating basic strings

A screenshot of the Visual Studio Code interface. The top menu bar shows 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', etc. The title bar says 'python [Administrator]'. The left sidebar has icons for file operations like open, save, and close. The main editor area contains a Python script named '1.py' with the following code:

```
string > 1.py
1 # Creating basic strings
2
3 str1='Hello Python'
4 print(str1)
5
6 str2="Hello Python2"
7 print(str2)
8
9 str3="""Triple quotes are generally used for
10 represent the multipline or
11 docstring"""
12 print(str3)
```

The terminal below the editor shows the output of running the script:

```
PS C:\Users\Administrator\Desktop\python\string> python .\1.py
Hello Python
Hello Python2
Triple quotes are generally used for
represent the multipline or
docstring
```

The bottom status bar shows 'Ln 12, Col 12 Spaces: 4 UTF-8 CRLF Python Go Live' and the date '06-02-2025'.

Q. String operations

A screenshot of a code editor interface, likely Visual Studio Code, running on Windows. The title bar says "python [Administrator]". The left sidebar shows files 1.py, 2.py, and 3.py. The main editor area contains the following code for 2.py:

```
string > 2.py
1 str="HELLO"
2 print(str[0])
3 print(str[1])
4 print(str[2])
5 print(str[3])
6 print(str[4])
7 print(str[6])
```

The terminal below shows the output of running 2.py:

```
PS C:\Users\Administrator\Desktop\python\string> python .\2.py
H
E
L
L
O
```

Traceback (most recent call last):
File "C:/Users/Administrator/Desktop/python/string/2.py", line 7, in <module>
print(str[6])
^
IndexError: string index out of range

The status bar at the bottom indicates "Ln 7, Col 12 Spaces: 4 UTF-8 CRLF Python Go Live".

Q. String range

A screenshot of a code editor interface, likely Visual Studio Code, running on Windows. The title bar says "python [Administrator]". The left sidebar shows files 1.py, 2.py, and 3.py. The main editor area contains the following code for 3.py:

```
string > 3.py
1 # String range
2 str="JAVATPOINT"
3 print(str[0:])
4 print(str[1:5])
5 print(str[2:4])
6 print(str[:3])
7 print(str[4:7])
```

The terminal below shows the output of running 3.py:

```
PS C:\Users\Administrator\Desktop\python\string> python .\3.py
JAVATPOINT
AVAT
VA
JAV
TPO
```

The status bar at the bottom indicates "Ln 7, Col 15 Spaces: 4 UTF-8 CRLF Python Go Live".

Q. String ranges

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows files 1.py, 2.py, 3.py, and 4.py.
- Code Editor:** Displays the contents of 4.py:

```
string > 4.py
1  # String ranges
2  str="JAVATPOINT"
3  print(str[-1])
4  print(str[-3])
5  print(str[-2:])
6  print(str[-4:-1])
7  print(str[-7:-2])
8  print(str[::-1])
9  print(str[-12])
```
- Terminal:** Shows the command `python .\4.py` being run, resulting in the output:

```
T
I
NT
OIN
ATPOI
TNIOPTTAVAJ
Traceback (most recent call last):
  File "C:\Users\Administrator\Desktop\python\string\4.py", line 9, in <module>
    print(str[-12])
               ^
IndexError: string index out of range
```
- Status Bar:** Shows the file is Python, has 9 lines, 14 columns, and was last modified on 06-02-2025 at 15:46.

Q. Reassigning strings

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows files 1.py, 2.py, 3.py, 4.py, and 5.py.
- Code Editor:** Displays the contents of 5.py:

```
string > 5.py
1  # Reassigning Strings
2
3  str="HELLO"
4  str[0]="h"
5  print(str)
```
- Terminal:** Shows the command `python .\5.py` being run, resulting in the output:

```
Traceback (most recent call last):
  File "C:\Users\Administrator\Desktop\python\string\5.py", line 4, in <module>
    str[0]="h"
               ^
TypeError: 'str' object does not support item assignment
```
- Status Bar:** Shows the file is Python, has 5 lines, 11 columns, and was last modified on 06-02-2025 at 15:46.

Q. String reassignment

A screenshot of the Visual Studio Code interface. The top bar shows the file path 'C:\Users\Administrator\Desktop\python\string' and the title 'python [Administrator]'. The left sidebar has icons for file, search, and other tools. The main editor window contains a Python script named '6.py' with the following code:string > 6.py
1 # String Reassignment
2
3 str="HELLO"
4 print(str)
5 str="hello"
6 print(str)The terminal below shows the output of running the script: 'Hello' followed by a new line. The status bar at the bottom indicates the file is 'Ln 6, Col 11' with 'Spaces: 4', 'UTF-8', 'CRLF', and 'Python' selected.

Q. String formatting

A screenshot of the Visual Studio Code interface. The top bar shows the file path 'C:\Users\Administrator\Desktop\python\string' and the title 'python [Administrator]'. The left sidebar has icons for file, search, and other tools. The main editor window contains a Python script named '7.py' with the following code:string > 7.py
1
2
3
4
5 print(str*3)
6 print(str+str)
7 print(str[4])
8 print(str[2:4])
9 print('w' in str)
10 print('wo' not in str)
11 print(r'C://python37')
12 print("The string str: %s"%str)
13The terminal below shows the output of running the script, demonstrating various string operations like concatenation, repetition, indexing, and membership testing. The status bar at the bottom indicates the file is 'Ln 13, Col 1' with 'Spaces: 4', 'UTF-8', 'CRLF', and 'Python' selected.

Q. Python string formatting

A screenshot of the Visual Studio Code interface. The left sidebar shows files 7.py, 8.py, and 9.py. The main editor window contains the following code for file 8.py:

```
string > 8.py
1 # Python String Formatting
2
3 print("""They said, "What's there?""")
4
5 print('They said, "What\'s going on?"')
6
7 print("They said, \'What's going on?\'"")
```

The terminal at the bottom shows the output of running the script:

```
PS C:\Users\Administrator\Desktop\python\string> python .\8.py
They said, "What's there?
They said, "What's going on?"
They said, "What's going on?"
```

Q. String formatting, format() method

A screenshot of the Visual Studio Code interface. The left sidebar shows files 7.py, 8.py, and 9.py. The main editor window contains the following code for file 9.py:

```
string > 9.py
1 # String Formatting, format() method
2
3 print("{} and {} both are the best friend".format("Devansh","Abhishek"))
4
5 print("{1} and {0} best players".format("Virat","Rohit"))
6
7 print("{(a),(b),(c)}".format(a="James",b="Peter",c="Ricky"))
```

The terminal at the bottom shows the output of running the script:

```
PS C:\Users\Administrator\Desktop\python\string> python .\9.py
Devansh and Abhishek both are the best friend
Rohit and Virat best players
James,Peter,Ricky
PS C:\Users\Administrator\Desktop\python\string>
```

Q. Lists and Tuples

A screenshot of a Windows desktop environment. At the top, there's a taskbar with icons for File Explorer, Edge browser, File Manager, Task View, and others. Below the taskbar is a dark-themed code editor window titled "python [Administrator]". The editor has tabs for 7.py, 8.py, 9.py, and 10.py. The 10.py tab is active, showing the following Python code:

```
string > 10.py
1 # List and Tuple
2
3 list_ = [4,5,7,1,7]
4 tuple_ = (4,1,8,3,9)
5
6 print("List is:",list_)
7 print("Tuple is:",tuple_)
```

The terminal window below the editor shows the output of running the script:

```
PS C:\Users\Administrator\Desktop\python\string> python .\10.py
List is: [4, 5, 7, 1, 7]
Tuple is: (4, 1, 8, 3, 9)
PS C:\Users\Administrator\Desktop\python\string>
```

At the bottom of the screen, there's a system tray with icons for battery, signal, volume, and date/time (06-02-2025).

Q. Mutable and immutable tuple

A screenshot of a Windows desktop environment, similar to the previous one. The taskbar at the top includes icons for File Explorer, Edge browser, File Manager, Task View, and others. Below the taskbar is a dark-themed code editor window titled "python [Administrator]". The editor has tabs for 7.py, 8.py, 9.py, 10.py, and 11.py. The 11.py tab is active, showing the following Python code:

```
string > 11.py
1 # Mutable and immutable tuple
2
3 list_ = ["Python","Lists","Tuples","Differences"]
4 tuple_ = ("Python","Lists","Tuples","Differences")
5
6 list_[3] = "Mutable"
7 print(list_)
8 try:
9     tuple_[3] = "Immutable"
10    print(tuple_)
11 except TypeError:
12     print("Tuples cannot be modified because they are immutable")
13
```

The terminal window below the editor shows the output of running the script:

```
PS C:\Users\Administrator\Desktop\python\string> python .\11.py
['Python', 'Lists', 'Tuples', 'Mutable']
Tuples cannot be modified because they are immutable
PS C:\Users\Administrator\Desktop\python\string>
```

At the bottom of the screen, there's a system tray with icons for battery, signal, volume, and date/time (06-02-2025).

Q. Size difference between tuples and lists

The screenshot shows a Windows desktop environment. At the top, there is a taskbar with several pinned icons: File Explorer, Microsoft Edge, File Explorer, Mail, Photos, and Task View. Below the taskbar is a Start button icon. The main area of the screen displays a code editor window titled "python [Administrator]". The code in the editor is:

```
string > 12.py
1 # Size difference between tuples and lists
2
3 list_ = ["Python", "Lists", "Tuples", "Differences"]
4 tuple_ = ("Python", "Lists", "Tuples", "Differences")
5
6 print("Size of tuple:", tuple_.__sizeof__())
7 print("Size of list:", list_.__sizeof__())
```

Below the code editor is a terminal window titled "powershell - string". The terminal output shows the execution of the script and its results:

```
PS C:\Users\Administrator\Desktop\python\string> python .\12.py
Size of tuple: 56
Size of list: 72
PS C:\Users\Administrator\Desktop\python\string>
```

The bottom right corner of the screen shows system status information: Ln 7, Col 42, Spaces: 4, UTF-8, CRLF, Python, Go Live, ENG 1621, US, 06-02-2025.