Python Operators

1. Arithmetic Operators:

Run Debug Stop Share Save { } Beautify Language Python 3

main.py

```python
a = 46
b = 4

print("For a =", a, "and b =", b,"\nCalculate the following:")

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. Reminder of two numbers: a mod b =', a % b)
print('7. Exponent of two numbers: a ^ b =',a ** b)
```

input

```
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. Reminder of two numbers: a mod b = 2
7. Exponent of two numbers: a ^ b = 4477456


...Program finished with exit code 0
Press ENTER to exit console.
```

2. Comparison Operators:

Run Debug Stop Share Save { } Beautify Language Python 3

main.py

```python
a = 46
b = 4

print("For a =", a, "and b =", b,"\nCheck 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 b:', a > b)
print('6. a is less than b:', a < b)
```

input

```
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 b: True
6. a is less than b: False


...Program finished with exit code 0
Press ENTER to exit console.
```

3. Assignment Operators:

```python
a = 34
b = 6
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)
print('a //= b:', a // b)
```

```
a += b: 40
a -= b: 28
a *= b: 204
a /= b: 5.666666666666667
a %= b: 4
a **= b: 1544804416
a //= b: 5


...Program finished with exit code 0
Press ENTER to exit console.
```

4. Bitwise Operators:

```python
a = 7
b = 8
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)
```

```
a & b : 0
a | b : 15
a ^ b : 15
~a : -8
a << b : 1792
a >> b : 0


...Program finished with exit code 0
Press ENTER to exit console.
```

5. Logical Operators:

```python
a = 7
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))
```

```
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


...Program finished with exit code 0
Press ENTER to exit console.
```

6. Membership Operators:

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

print("Given List:", myList)
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.")
if (y in myList):
    print("y =", y,"is present in the given list.")
else:
    print("y =", y,"is NOT present in the given list.")
```

```
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.


...Program finished with exit code 0
Press ENTER to exit console.
```

7. Identity Operators:

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

c = a
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)
```

```
a is c =>  True
a is not c =>  False
a is b =>  False
a is not b =>  True
a == b =>  True
a != b =>  False


...Program finished with exit code 0
Press ENTER to exit console.
```

REVERSE A STRING IN PYTHON:

1. Using for loop:

```python
def reverse_string(str):
    str1 = ""
    for i in str:
        str1 = i + str1
    return str1

str = "Trivandrum "
print("The original string is: ",str)
print("The reverse string is :",reverse_string(str)) # Function call
```

```
The original string is:  Trivandrum
The reverse string is :  murdnavirT


...Program finished with exit code 0
Press ENTER to exit console.
```

2. Using while loop:

```python
# Reverse string
# Using a while loop

str = "Trivandrum"
print ("The original string  is : ",str)
reverse_String = ""
count = len(str)
while count > 0:
    reverse_String += str[ count - 1 ]
    count = count - 1
print ("The reversed string using a while loop is : ",reverse_String)# reversed string
```

```
The original string  is :  Trivandrum
The reversed string using a while loop is :  murdnavirT


...Program finished with exit code 0
Press ENTER to exit console.
```

3. Using the slice ([]) operator:

```python
def reverse(str):
    str = str[::-1]
    return str

s = "Trivandrum"
print ("The original string  is : ",s)
print ("The reversed string using extended slice operator  is : ",reverse(s))
```

```
The original string  is :  Trivandrum
The reversed string using extended slice operator  is :  murdnavirT


...Program finished with exit code 0
Press ENTER to exit console.
```

4. Using reverse function with join:

```python
def reverse(str):
    string = "".join(reversed(str)) # reversed() function inside the join() function
    return string

s = "Trivandrum"

print ("The original string is : ",s)
print ("The reversed string using reversed() is : ",reverse(s) )
```

```
The original string is :  Trivandrum
The reversed string using reversed() is :  murdnavirT


...Program finished with exit code 0
Press ENTER to exit console.
```

5. Using recursion():

```python
def reverse(str):
    if len(str) == 0: # Checking the lenght of string
        return str
    else:
        return reverse(str[1:]) + str[0]

str = "HArish Arjun"
print ("The original string  is : ", str)
print ("The reversed string(using recursion) is : ", reverse(str))
```

```
The original string  is :  HArish Arjun
The reversed string(using recursion) is :  nujrA hsirAH


...Program finished with exit code 0
Press ENTER to exit console.
```

PYTHON CSV FILE :

```python
# Importing the csv module
import csv
# Open file by passing the file path.
with open(r'C:\Users\Administrator\Documents\python\example.csv') as csv_file:
    csv_read = csv.reader(csv_file, delimiter=',')  # Delimiter is comma
    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.')
```

Output:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

 Files\Python313\python.exe' 'c:\Users\Administrator\.vscode\extensions\ms-python.debugpy-2024.14.0-wi
n32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '59728' '--' 'c:\Users\Administrator\reci
pewebsite\import_csv_module.py'
Column names are Name, Roll Number, Department
        Alice roll number is:  101 and department is: Computer Science.
        Bob roll number is:  102 and department is: Mechanical.
        Charlie roll number is:  103 and department is: Electrical.
        David roll number is:  104 and department is: Civil.
        Emma roll number is:  105 and department is: Electronics.
Processed 6 lines.
PS C:\Users\Administrator\recipewebsite>
```

Python IF-ELSE:

1. if statement:

```python
num = int(input("enter the number:"))
if num%2 == 0:
    print("The Given number is an even number")
```

```
enter the number:24
The Given number is an even number


...Program finished with exit code 0
Press ENTER to exit console.
```

2. print the largest of the three numbers

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

```
Enter a: 5
Enter b: 78
Enter c: 23
From the above three numbers given b is largest
```

If-Else statement:

1.

```python
age = int (input("Enter your age: "))
if age>=18:
    print("You are eligible to vote !!");
else:
    print("Sorry! you have to wait !!");
```

```
Enter your age: 76
You are eligible to vote !!
```

2. check whether a number is even or not:

```python
num = int(input("enter the number:"))
if num%2 == 0:
    print("The Given number is an even number")
else:
    print("The Given Number is an odd number")
```

```
enter the number:34
The Given number is an even number
```

Elif statement:

1.

```python
number = int(input("Enter the number:"))
if number==10:
    print("The given number is equals to 10")
elif number==50:
    print("The given number is equal to 50");
elif number==100:
    print("The given number is equal to 100");
else:
    print("The given number is not equal to 10, 50 or 100");
```

```
Enter the number:4
The given number is not equal to 10, 50 or 100


...Program finished with exit code 0
Press ENTER to exit console.
```

2.

```python
marks = int(input("Enter the marks? "))
if marks > 85 and marks <= 100:
    print("Congrats ! you scored grade A ...")
elif marks > 60 and marks <= 85:
    print("You scored grade B + ...")
elif marks > 40 and marks <= 60:
    print("You scored grade B ...")
elif (marks > 30 and marks <= 40):
    print("You scored grade C ...")
else:
    print("Sorry you are fail ?")
```

```
Enter the marks? 45
You scored grade B ...
```

WHILE LOOPS:

1. sum of squares:

```python
num = 21
summation = 0
c = 1

while c <= num:
    summation = c**2 + summation
    c = c + 1
print("The sum of squares is", summation)
```

```
The sum of squares is 3311


...Program finished with exit code 0
Press ENTER to exit console.
```

2. Prime Numbers:

```python
num = [34, 12, 54, 23, 75, 34, 11]
def prime_number(number):
    condition = 0
    iteration = 2
    while iteration <= number / 2:
        if number % iteration == 0:
            condition = 1
            break
        iteration = iteration + 1

    if condition == 0:
        print(f"{number} is a PRIME number")
    else:
        print(f"{number} is not a PRIME number")
for i in num:
    prime_number(i)
```

```
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
```

3. Armstrong:

```python
n = int(input())
n1=str(n)
l=len(n1)
temp=n
s=0
while n!=0:
    r=n%10
    s=s+(r**1)
    n=n//10
if s==temp:
    print("It is an Armstrong number")
else:
    print("It is not an Armstrong number ")
```

```
121
It is not an Armstrong number
```

4. Multiplication Table using While Loop:

```python
num = 21
counter = 1
print("The Multiplication Table of: ", num)
while counter <= 10:
    ans = num * counter
    print (num, 'x', counter, '=', ans)
    counter += 1
```

```
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


...Program finished with exit code 0
Press ENTER to exit console.
```

5. Break Statement & Continue Statement:

```python
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

for number in numbers:
    if number % 2 == 0:
        print("Skipping even number:", number)
        continue
    if number == 7:
        print("Encountered 7, breaking the loop.")
        break
    print("Processing odd number:", number)

print("Loop has completed.")
```

```
Processing odd number: 1
Skipping even number: 2
Processing odd number: 3
Skipping even number: 4
Processing odd number: 5
Skipping even number: 6
Encountered 7, breaking the loop.
Loop has completed.


...Program finished with exit code 0
Press ENTER to exit console.
```

FOR LOOPS:

1.

```python
numbers = [3, 5, 23, 6, 5, 1, 2, 9, 8]
sum_ = 0
for num in numbers:
    sum_ = sum_ + num ** 2
print("The sum of squares is: ", sum_)
```

```
The sum of squares is:  774


...Program finished with exit code 0
Press ENTER to exit console.
```

2.

```python
my_list = [3, 5, 6, 8, 4]
for iter_var in range( len( my_list ) ):
    my_list.append(my_list[iter_var] + 2)
print( my_list )
```

```
[3, 5, 6, 8, 4, 5, 7, 8, 10, 6]


...Program finished with exit code 0
Press ENTER to exit console.
```

3.

```python
student_name_1 = 'Itika'
student_name_2 = 'Parker'
records = {'Itika': 90, 'Arshia': 92, 'Peter': 46}
def marks(student_name):
    for a_student in records:
        if a_student == student_name:
            return records[a_student]
            break
    else:
        return f'There is no student of name {student_name} in the records'
print(f"Marks of {student_name_1} are: ", marks(student_name_1))
print(f"Marks of {student_name_2} are: ", marks(student_name_2))
```

```
Marks of Itika are:  90
Marks of Parker are:  There is no student of name Parker in the records


...Program finished with exit code 0
Press ENTER to exit console.
```

4.

```python
import random
numbers = [ ]
for val in range(0, 11):
    numbers.append( random.randint( 0, 11 ) )
for num in range( 0, 11 ):
    for i in numbers:
        if num == i:
            print( num, end = " " )
```

```
0 1 2 3 4 5 6 7 9 10

...Program finished with exit code 0
Press ENTER to exit console.
```

STRINGS

1.

```python
str1 = 'Hello Python'
print(str1)
str2 = "Hello Python"
print(str2)
str3 = '''''Triple quotes are generally used for
    represent the multiline or
    docstring'''
print(str3)
```

```
Hello Python
Hello Python
''Triple quotes are generally used for
    represent the multiline or
    docstring


...Program finished with exit code 0
Press ENTER to exit console.
```

2. Indexing and Splitting:

```python
str = "HELLO"
print(str[0])
print(str[1])
print(str[2])
print(str[3])
print(str[4])
print(str[6])
```

```
H
E
L
L
O
Traceback (most recent call last):
  File "/home/main.py", line 7, in <module>
    print(str[6])
          ~~~^^^
IndexError: string index out of range
```

3.

```python
str = "JAVATPOINT"
print(str[0:])
print(str[1:5])
print(str[2:4])
print(str[:3])
print(str[4:7])
```

```
JAVATPOINT
AVAT
VA
JAV
TPO
```

4.

```python
str = "Hello"
str1 = " world"
print(str*3)
print(str+str1)
print(str[4])
print(str[2:4]);
print('w' in str)
print('wo' not in str1)
print(r'C://python37')
print("The string str : %s"%(str))
```

```
HelloHelloHello
Hello world
o
ll
False
False
C://python37
The string str : Hello
```

Python List vs Tuple

1. List and Tuple Syntax Differences:

```python
# Python code to show the difference between creating a list and a tuple

list_ = [4, 5, 7, 1, 7]
tuple_ = (4, 1, 8, 3, 9)

print("List is: ", list_)
print("Tuple is: ", tuple_)
```

```
List is:  [4, 5, 7, 1, 7]
Tuple is:  (4, 1, 8, 3, 9)

=== Code Execution Successful ===
```

2. Updating the element of list and tuple at a particular index :

```python

# creating a list and a tuple
list_ = ["Python", "Lists", "Tuples", "Differences"]
tuple_ = ("Python", "Lists", "Tuples", "Differences")

list_[3] = "Mutable"
print( list_ )
try:
    tuple_[3] = "Immutable"
    print( tuple_ )
except TypeError:
    print( "Tuples cannot be modified because they are immutable"
    )
```

```
['Python', 'Lists', 'Tuples', 'Mutable']
Tuples cannot be modified because they are immutable

=== Code Execution Successful ===
```

3. Code to show the difference in the size of a list and a tuple:

main.py

```python
#creating a list and a tuple
list_ = ["Python", "Lists", "Tuples", "Differences"]
tuple_ = ("Python", "Lists", "Tuples", "Differences")
# printing sizes
print("Size of tuple: ", tuple_.__sizeof__())
print("Size of list: ", list_.__sizeof__())
```

Output

```
Size of tuple:  56
Size of list:  72

=== Code Execution Successful ===
```