pr-4

October 5, 2023

1 Loops

Loops are indispensable constructs in programming that enable the automation of repetitive tasks and systematic processing of data. In this discussion, we'll delve into two fundamental types of loops:

1.1 The For Loop

The `for` loop is the go-to choice when you have a clear idea of how many times you want a specific block of code to run. It consists of three crucial components:

1. **Initialization**: At the start of the loop, you set up initial values and conditions. This step is executed only once.
2. **Condition**: The loop continues executing as long as a defined condition remains true. If the condition becomes false, the loop terminates.
3. **Iteration**: This part defines how the loop variable changes with each iteration. It is crucial for progress within the loop.

why is `for` loop incredibly useful:

- **Iterating Through Collections**: It excels at traversing collections like lists or arrays. You can easily access and manipulate each element within such structures.
- **Predictable Control Flow**: With a predefined number of iterations, the `for` loop provides a clear and structured approach to loop control.

1.2 The While Loop

In contrast, the `while` loop offers flexibility when you cannot determine the exact number of iterations in advance. It continues execution as long as a specified condition remains true. Key features include:

- **Dynamic Control**: Unlike the `for` loop, the termination of a `while` loop is entirely dependent on changing conditions. This means you have the freedom to adjust the loop's behavior during execution.
- **Versatility**: The `while` loop is ideal for scenarios where the number of iterations is uncertain or where you need to monitor an ongoing condition.

Both `for` and `while` loops play crucial roles in programming, offering distinct advantages depending on your specific requirements. They are fundamental tools that empower developers to create efficient, automated solutions for a wide range of tasks.

In summary, loops are the backbone of automation in programming, enabling the efficient execution of repetitive actions and the systematic processing of data. Choosing the right loop type, whether it's the structured `for` loop or the flexible `while` loop, is essential for achieving your programming goals.

[18]:

```
# Write a program to read the numbers until -1 is encountered also count the +,␣
 ↪- and 0 entered by the user.

inp = int(input("Please enter the fist number (enter -1 to exit): "))
plus = 0
minus = 0
zero = 0
arr = []
while inp != -1:
    if inp > 0:
        plus += 1
    elif inp < 0:
        minus += 1
    else:
        zero += 1
    arr.append(inp)
    inp = int(input("Please enter the next number (enter -1 to exit): "))
print("The inputs were: ", arr)
print("Plus: ", plus, "Minus: ", minus, "Zero: ", zero)
```

```
The inputs were:  [1, 54, 2, -34, -6, -4, -7, -5, -2, 0, 0, 0, 5, 3, 65, 2, -5,
-3]
Plus:  7 Minus:  8 Zero:  3
```

[19]:

```
# Wap to identify the number is armstrong number or not.

num = int(input("Please enter a number: "))
temp = num
sum = 0
while temp > 0:
    digit = temp % 10
    sum += digit ** len(str(num))
    temp //= 10
if num == sum:
    print(num, "is an Armstrong number")
else:
    print(num, "is not an Armstrong number")
```

```
928349 is not an Armstrong number
```

[20]:

```
# Wap to enter binary number and convert it into decimal number

num = int(input("Please enter a binary number: "))
dec = 0
base = 1
temp = num
while temp > 0:
    digit = temp % 10
    dec += digit * base
    base *= 2
    temp //= 10
print("The decimal number for the binary",num,"is: ", dec)
```

```
The decimal number for the binary 10101010101010101010 is:  699050
```

[21]:

```
# Wap to read the character until the asterisk sign is encountered. Also count␣
  ↪the number of uppercase,lowercase and numbers entered by user.

inp = input("Please enter the fist character (enter * to exit): ")
upper = 0
lower = 0
num = 0
arr = []
while inp != '*':
    if inp.isupper():
        upper += 1
    elif inp.islower():
        lower += 1
    elif inp.isdigit():
        num += 1
    arr.append(inp)
    inp = input("Please enter the next character (enter * to exit): ")
print("The inputs were: ", arr)
print("Upper: ", upper, "Lower: ", lower, "Number: ", num)
```

```
The inputs were:  ['a', 'gh', 'd', 'j', 'r', 'h', 'd', 'h', 'd', 'b', 'S', 'P',
'PD', 'F', 'F', 'GH', 'E', 'Y', 'J', 'V', 'N', 'D', '3', '6', '2', '6', '6',
'8', '5', '33', '6', '3']
Upper:  12 Lower:  10 Number:  10
```

[22]:

```
# Wap to print reverse of a number.

num = int(input("Please enter a number: "))
rev = 0
temp = num
while temp > 0:
    digit = temp % 10
    rev = rev * 10 + digit
    temp //= 10
print("The reverse of", num, "is: ", rev)
```

```
The reverse of 1298367987 is:  7897638921
```

[23]:

```
# Wap to print multiplication table of number. number entered by user.

num = int(input("Please enter a number: "))
for i in range(1, 11):
    print(num, "x", i, "=", num*i)
```

```
28 x 1 = 28
28 x 2 = 56
28 x 3 = 84
28 x 4 = 112
28 x 5 = 140
28 x 6 = 168
28 x 7 = 196
28 x 8 = 224
28 x 9 = 252
28 x 10 = 280
```

[35]:

```
# Wap to generate calendar of a month given the start day and number of days in␣
 ↪that month.

start = int(input("Please enter the start day of the month: "))
days = int(input("Please enter the number of days in the month: "))
print("SUN  MON  TUE  WED  THU  FRI  SAT")
for i in range(1, start):
    print("     ", end="")
for i in range(1, days+1):
    print(i if i>=10 else f"{i} ", end="   ")

    if (i+start-1) % 7 == 0:
        print()
print()
```

```
SUN  MON  TUE  WED  THU  FRI  SAT
               1    2    3    4
5    6    7    8    9    10   11
12   13   14   15   16   17   18
19   20   21   22   23   24   25
26   27   28   29   30   31
```

[25]:

```
# Wap to print the pattern
# *
# * *
# * * *
# * * * *
# * * * * *

for i in range(1, 6):
    for j in range(1, i+1):
        print("*", end=" ")
    print()
```

```
*
* *
* * *
* * * *
* * * * *
```

[26]:

```
# WAP to create a pattern of stars where the row is entered by the user
# *
# **
# *…

row = int(input("Please enter the number of rows: "))
for i in range(1, row+1):
    for j in range(1, i+1):
        print("*", end=" ")
    print()
```

```
*
* *
* * *
* * * *
* * * * *
* * * * * *
* * * * * * *
* * * * * * * *
* * * * * * * * *
* * * * * * * * * *
```

[27]:

```
# WAP to create pattern like
# 1
# 22
# 333
# 4444
# 55555…

row = int(input("Please enter the number of rows: "))
for i in range(1, row+1):
    for j in range(1, i+1):
        print(i, end=" ")
    print()
```

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
6 6 6 6 6 6
7 7 7 7 7 7 7
8 8 8 8 8 8 8 8
9 9 9 9 9 9 9 9 9
10 10 10 10 10 10 10 10 10 10
```

[28]:

```
# WAP to create pattern like
# 1
# 121
# 12321
# 1234321

row = int(input("Please enter the number of rows: "))
for i in range(1, row+1):
    for j in range(1, i+1):
        print(j, end=" ")
    for j in range(i-1, 0, -1):
        print(j, end=" ")
    print()
```

```
1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 5 6 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 9 10 9 8 7 6 5 4 3 2 1
```