For loop with range

For loop with range

In the previous lessons we dealt with sequential programs and conditions. Often the program needs to repeat some block several times. That's where the loops come in handy. There are `for` and `while` loop operators in Python, in this lesson we cover `for`.

`for` loop iterates over any sequence. For instance, any string in Python is a sequence of its characters, so we can iterate over them using `for`:

```
for character in 'hello':
    print(character)
```

for character in 'hello':

Another use case for a for-loop is to iterate some integer variable in increasing or decreasing order. Such a sequence of integer can be created using the function `range(min_value, max_value)`:

```
for i in range(5, 8):
    print(i, i ** 2)
print('end of loop')
# 5 25
# 6 36
# 7 49
# end of loop
```

Function `range(min_value, max_value)` generates a sequence with numbers `min_value`, `min_value + 1`, ..., `max_value - 1`. The last number is not included.

There's a reduced form of range() - `range(max_value)`, in which case min_value is implicitly set to zero:

```
for i in range(3):
```

```
    print(i)

# 0

# 1

# 2
```

This way we can repeat some action several times:

```
for i in range(2 ** 2):
    print('Hello, world!')
```

Same as with if-else, indentation is what specifies which instructions are controlled by `for` and which aren't.

Range() can define an empty sequence, like `range(-5)` or `range(7, 3)`. In this case the for-block won't be executed:

```
for i in range(-5):
    print('Hello, world!')
```

Let's have more complex example and sum the integers from 1 to n inclusively.

```
result = 0

n = 5

for i in range(1, n + 1):
    result += i
    # this ^^ is the shorthand for
    # result = result + i

print(result)
```

Pay attention that maximum value in range() is `n + 1` to make `i` equal to n on the last step.

To iterate over a decreasing sequence, we can use an extended form of range() with three arguments - `range(start_value, end_value, step)`. When omitted, the step is implicitly equal to 1. However, can be any non-zero value. The loop always includes start_value and excludes end_value during iteration:

```
for i in range(10, 0, -2):
```

```
    print(i)

# 10

# 8

# 6

# 4

# 2
```

setting the function print()

By default, the function `print()` prints all its arguments separating them by a space and the puts a newline symbol after it. This behavior can be changed using keyword arguments `sep` (separator) and `end`.

```
print(1, 2, 3)

print(4, 5, 6)

print(1, 2, 3, sep=', ', end='. ')

print(4, 5, 6, sep=', ', end='. ')

print()

print(1, 2, 3, sep='', end=' -- ')

print(4, 5, 6, sep=' * ', end='.')
```

Problem «Series - 1» (Easy)

Statement

Given two integers A and B (A ≤ B). Print all numbers from A to B inclusively.

Your solution

```
# Read an integer:

a = int(input())

b = int(input())
```

```
# Print a value:

for i in range (a, b+1 ):

    print(i)
```

Problem «Series - 2» (Easy)

Statement

Given two integers A and B. Print all numbers from A to B inclusively, in ascending order, if A < B, or in descending order, if A ≥ B.

Your solution

```
# Read an integer:

a = int(input())

b = int(input())

# Print a value:

if a < b:

    for i in range(a, b + 1):

        print(i)

else:

    for i in range(a, b - 1, -1):

        print(i)
```

Problem «Sum of ten numbers» (Easy)

Statement

10 numbers are given in the input. Read them and print their sum. Use as few variables as you can.

Your solution

```
# Read in 10 numbers:

sum = 0

for i in range(1, 11):

    vi = int(input())

    sum += vi

# Print a value:

print(sum)
```

Suggested solution

```
res = 0

for i in range(10):

    res += int(input())

print(res)
```

Problem «Sum of N numbers» (Medium)

Statement

N numbers are given in the input. Read them and print their sum.

The first line of input contains the integer N, which is the number of integers to follow. Each of the next N lines contains one integer. Print the sum of these N integers.

Your solution

```
# Read in 10 numbers:
```

```
N = int(input())

sum = 0

for i in range(1, N+1):

    vi = int(input())

    sum += vi

# Print a value:

print(sum)
```

Suggested solution

```
n = int(input())

res = 0

for i in range(n):

    res += int(input())

print(res)
```

Problem «Sum of cubes» (Easy)

Statement

For the given integer N calculate the following sum:

$$1^3+2^3+...+N^3$$

Your solution

```
n = int(input())

total = 0

for i in range(1, n+1):

    sqrt_num = i**3

    total += sqrt_num

print(total)
```

Suggested solution

```
res = 0

for i in range(1, int(input()) + 1):

    res += i ** 3

print(res)
```

Problem «Factorial» (Easy)

Statement

In mathematics, the factorial of an integer n, denoted by $n!$ is the following product:

$$n! = 1 \times 2 \times ... \times n$$

For the given integer n calculate the value $n!$. Don't use `math` module in this exercise.

Your solution

```
# Read in 10 numbers:

N = int(input())

sum = 1

for i in range(N, 1, -1):

    sum *= i

# Print a value:

print(sum)
```

Problem «The number of zeros» (Medium)

Statement

Given N numbers: the first number in the input is N, after that N integers are given. Count the number of zeros among the given integers and print it.

You need to count the number of numbers that are equal to zero, not the number of zero digits.

Your solution

```
# Read an integer:

N = int(input())

sum = 0

for i in range(1, N+1):

   vi= int(input())

   if vi ==0:

      sum += 1

# Print a value:

print(sum)

num_zeroes = 0

for i in range(int(input())):

   if int(input()) == 0:

      num_zeroes += 1

print(num_zeroes)
```

Problem «Adding factorials» (Medium)

Statement

Given an integer n, print the sum $1!+2!+3!+...+n!$.

This problem has a solution with only one loop, so try to discover it. And don't use the math library :)

Your solution

```
# Read an integer:
```

```
n = int(input())

fact = 1

sum = 0

for i in range (1, n+1):

      fact *= i

      sum  += fact

print(sum)
```

Suggested solution

```
n = int(input())

partial_factorial = 1

partial_sum = 0

for i in range(1, n + 1):

   partial_factorial *= i

   partial_sum += partial_factorial

print(partial_sum)
```

Problem «Ladder» (Medium

Statement

For given integer n ≤ 9 print a ladder of n steps. The k-th step consists of the integers from 1 to k without spaces between them.

To do that, you can use the `sep` and `end` arguments for the function print().

Your solution

```
n = int(input())

for i in range(1,n+2):
```

```
    print(sep= '',)
    for j in range(1,i):
        print(j, end='')
n = int(input())
for i in range(1, n + 1):
    for j in range(1, i + 1):
        print(j, sep='', end='')
    print()
```

Problem «Lost card» (Hard)

Statement

There was a set of cards with numbers from 1 to N. One of the card is now lost. Determine the number on that lost card given the numbers for the remaining cards.

Given a number N, followed by N − 1 integers - representing the numbers on the remaining cards (distinct integers in the range from 1 to N). Find and print the number on the lost card.

Your solution

```
N = int(input())
total = 0
sum = 0
for i in range(1, N+1):
    total += i
for i in range(1, N):
    vi = int(input())
    sum += vi
result = total - sum
```

```
print(result)
```

Suggested solution

```
n = int(input())

sum_cards = 0

for i in range(1, n + 1):

    sum_cards += i

# One can prove the following:

# sum_cards == n * (n + 1) // 2

# However, we'll calculate that using the loop.

for i in range(n - 1):

    sum_cards -= int(input())

print(sum_cards)
```