

How can you take multiple space-separated values as input?

```
In [1]: k = print(input("Please enter a sentence: ").split())
```

```
Please enter a sentence: raja debnath.  
['raja', 'debnath.']}
```

5. How do you check if a number entered by the user is positive, negative, or zero?

```
In [6]: k = input("please enter a number: ")  
if int(k) > 0:  
    print(f"{k} is a positive number.")  
elif int(k) < 0:  
    print(f"{k} is a negative number.")  
else:  
    print(f"{k} is zero")
```

```
please enter a number: 5  
5 is a positive number.
```

6. How do you convert user input to a list of integers?

```
In [8]: k = print(input("Please enter a sentence: ").split())      #split on the basis of space  
Please enter a sentence: 5 3 6 9  
['5', '3', '6', '9']
```

```
In [16]: num = int(k) for k in input("Please enter a sentence: ").split()  
print(num)
```

```
Cell In[16], line 1  
    num = int(k) for k in input("Please enter a sentence: ").split()  
          ^  
SyntaxError: invalid syntax
```

```
In [19]: num = [k for k in input("Please enter a sentence: ").split()]  
print(num)
```

```
Please enter a sentence: num = int(k) for k in input("Please enter a sentence: ").split()  
print(num)  
['num', '=', 'int(k)', 'for', 'k', 'in', 'input("Please', 'enter', 'a', 'sentence:',  
'").split()', 'print(num)']
```

```
In [20]: k = input("Please enter: ").split()  
print(k)
```

```
Please enter: num = int(k) for k in input("Please enter a sentence: ").split() print(num)
['num', '=', 'int(k)', 'for', 'k', 'in', 'input("Please', 'enter', 'a', 'sentence:', '')'].split()', 'print(num)']
```

7. How do you accept a string input and print it in uppercase?

```
In [21]: k = input("Please enter: ").upper()
print(k)
```

```
Please enter: raja
RAJA
```

8. Write a Python program that accepts a string and prints the number of vowels in it.

```
In [23]: user_input = input("Please enter your sentence: ").lower()
vowels = 'aeiou'
j = 0
for i in user_input:
    if i in vowels:
        j += 1
print(j)
```

```
Please enter your sentence: Write a Python program that accepts a string and prints t
he number of vowels in it
21
```

```
In [25]: user_input = input("Please enter your sentence: ").lower()
vowels = 'aeiou'
k = sum(1 for char in user_input if char in vowels)
print(k)
```

```
Please enter your sentence: Write a Python program that accepts a string and prints t
he number of vowels in it
21
```

```
numbers = [1, 2, 3, 4, 5] total = sum(numbers) print(total) # Output: 15 Using with a Tuple
```

```
numbers_tuple = (10, 20, 30) total = sum(numbers_tuple) print(total) # Output: 60 Using with a
Start Value You can also provide a start value that will be added to the sum of the iterable.
```

```
numbers = [1, 2, 3] total = sum(numbers, 10) print(total) # Output: 16 Summing Elements of a
List of Lists If you have a list of lists and you want to sum the elements of each sublist, you can
use a list comprehension:
```

```
list_of_lists = [[1, 2, 3], [4, 5], [6, 7, 8]] total = sum([sum(sublist) for sublist in list_of_lists])
print(total) # Output: 36 Summing with a Condition You can also sum elements that meet a
certain condition using a generator expression:
```

```
numbers = [1, 2, 3, 4, 5] total = sum(x for x in numbers if x % 2 == 0) print(total) # Output: 6 (2 + 4)
```

9. Write a program that takes a number as input and checks if it is even or odd.

```
In [29]: num = int(input("Please enter a number: "))
if num%2 == 0 and num != 0:
    print("The number is even")
elif num%2 != 0:
    print("The number is odd")
else:
    print("The number is zero")
```

Please enter a number: 5

The number is odd

10. How would you check if a string is a palindrome using input()?

```
In [31]: text = input("please enter a string: ")
if text == text[::-1]:
    print("The string is a palindrome")
else:
    print("The string is not a palindrome")
```

please enter a string: abba

The string is a palindrome

14. How do you accept a list of comma-separated values as input?

```
In [34]: k = input("Please enter a sentence: ").split(",")
print(k)
```

Please enter a sentence: raja, is, a, good boy
['raja', ' is', ' a', ' good boy']

15. Write a Python program that takes two numbers as input and prints their product.

```
In [36]: num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))
print("Product:", num1 * num2)
```

Enter first number: 5

Enter second number: 6

Product: 30

16. Write a program that checks if the input number is a prime number.

```
In [42]: num = int(input("Enter first number: "))
num1 = int(num**(0.5))
if num > 1:
    for i in range (2,num1+1):
        if num%i == 0:
            print("is not a prime")
            break
    else:
        print("is a prime number")
else:
    print("please enter a correct number")
```

```
Enter first number: 1001
is not a prime
```

17. How can you accept a boolean value (True/False) from the user?

```
In [44]: user_input = input("Enter True or False: ").lower() == "true"
Enter True or False: 2
```

18. Write a program that accepts a string and prints the reverse of that string.

```
In [45]: text = input("please enter a string: ")
print(text[::-1])
```

```
please enter a string: raja
ajar
```

19. Write a program that asks for a user's name and age and prints a message.

```
In [53]: name = input("Please enter your name: ")
age = input("Please enter your age: ")
k = {
    "name" : name,
    "age" : age,
}
for i in k:
    print(f"{i}: {k[i]}")
```

```
Please enter your name: Raja Debnath
Please enter your age: 49
name: Raja Debnath
age: 49
```

```
In [3]: k={}
name = input("Please enter your name: ")
age = input("Please enter your age: ")
while True:

    k[name] = age
    user_input = input("would you like to continue, \"y\" for yes and \"n\" for no: ")
    if user_input == "y":
        name = input("Please enter your name: ")
        age = input("Please enter your age: ")
    else:
        break
for i in k:
    print(f"{i} : {k[i]}")
```

```
Please enter your name: Raja Debnath
Please enter your age: 56
would you like to continue, "y" for yes and "n" for no: y
Please enter your name: Avra Dutta
Please enter your age: 69
would you like to continue, "y" for yes and "n" for no: y
Please enter your name: Arjun Debnath
Please enter your age: 456
would you like to continue, "y" for yes and "n" for no: n
Raja Debnath : 56
Avra Dutta : 69
Arjun Debnath : 456
```

```
In [54]: name = input("Enter your name: ")
age = int(input("Enter your age: "))
print(f"Hello, {name}. You are {age} years old.")
```

```
Enter your name: Raja Debnath
Enter your age: 49
Hello, Raja Debnath. You are 49 years old.
```

20. Write a program to calculate the factorial of a number using input().

```
In [63]: num = int(input("enter a numer for factorial: "))
fact = 1
for i in range(1, num + 1):
    fact *= i
print(fact)
```

```
enter a numer for factorial: 4
24
```

21. How do you prevent a user from entering an empty string?

```
In [66]: user_input = input("Enter something: ").strip()
if not user_input:
    print("Input cannot be empty.")
else:
    print(f"You entered: {user_input}")
```

```
Enter something: 26
You entered: 26
```

22. Write a program to check if the entered number is a perfect square.

```
In [73]: user_input = int(input("Enter a number: "))
k = user_input**(0.5)
if (int(k))**2 == user_input:
    print(f"{user_input} is a perfect square & the sqrt is {k}")
else:
    print(f"{user_input} is not a perfect square")
```

```
Enter a number: 2531
2531 is not a perfect square
```

23. Write a program that asks for a year and determines if it's a leap year.

```
In [75]: user_input = int(input("Enter a year: "))
if (user_input%4 == 0 and user_input % 100 != 0) or user_input % 400 == 0:    # how to
    print(f"{user_input} is a leap year")
else:
    print(f"{user_input} is not a leap year")
```

```
Enter a year: 1989
1989 is not a leap year
```

24. How can you remove leading and trailing spaces from a string input?

```
In [76]: user_input = input("Enter something: ").strip()
print(f"You entered: {user_input}")
```

```
Enter something: rja debnath
You entered: rja debnath
```

25. How do you handle incorrect inputs when you expect an integer using input()?

```
In [86]: user_input = input("Enter something: ")
k = '0123456789'
for i in user_input:
```

```
if i not in k:  
    print("not a integr")  
    break  
else:  
    print("integer")
```

Enter something: f
not a integr

```
In [92]: while True:  
    try:  
        num = int(input("Enter a number: "))  
        break  
    except ValueError:  
        print("Invalid input! Please enter a valid integer.")
```

Enter a number: 1
Invalid input! Please enter a valid integer.
Enter a number: 1
Invalid input! Please enter a valid integer.
Enter a number: 2

26. Write a program that accepts a string and counts the occurrence of a particular character.

```
In [3]: text1 = input("Enter a string: ")  
char = input("Enter a character to count: ")  
print(f"Occurrence of {char}: {text1.count(char)}")
```

Enter a string: raja debnath
Enter a character to count: a
Occurrence of a: 3

```
In [93]: user_input = input("Enter something: ")  
for char in user_input:  
    print(f"the occurrence of {char} is {user_input.count(char)}")
```

Enter something: raja
the occurrence of r is 1
the occurrence of a is 2
the occurrence of j is 1
the occurrence of a is 2

```
In [2]: #how to count occurance of each character in a sentence  
user_input = input("Enter something: ")  
i=0  
for char in user_input:  
    i += 1  
    if char not in user_input[0:i-1]:  
        print(f"the occurrence of {char} is {user_input.count(char)}")
```

```
Enter something: raja debnath
the occurance of r is 1
the occurance of a is 3
the occurrence of j is 1
the occurrence of   is 1
the occurrence of d is 1
the occurrence of e is 1
the occurrence of b is 1
the occurrence of n is 1
the occurrence of t is 1
the occurrence of h is 1
```

In [103...]

```
#how to count occurance of each character in a sentence
user_input = input("Enter something: ")
k = set(user_input)
print(k)
for char in k:
    print(f"the occurrence of {char} is {user_input.count(char)}")
```

```
Enter something: raja debnath
{' ', 'r', 'j', 'a', 'b', 't', 'h', 'd', 'n', 'e'}
the occurrence of   is 1
the occurrence of r is 1
the occurrence of j is 1
the occurrence of a is 3
the occurrence of b is 1
the occurrence of t is 1
the occurrence of h is 1
the occurrence of d is 1
the occurrence of n is 1
the occurrence of e is 1
```

28. Write a program that accepts a number and prints whether it is a multiple of 10.

```
In [5]: user_input = int(input("Enter something: "))
if user_input % 10 == 0:
    print("Divisible by 10")
else:
    print("Not divisible by 10")
```

```
Enter something: 56
Not divisible by 10
```

29. How would you check if a string entered by the user contains only alphabets using input()?

```
In [10]: user_input1 = input("Enter something: ").lower()
k = 'qwertyuioplkjhgfdaszcxbnm'
for i in user_input1:
    if i not in k:
        print("Is not a word")
        break
```

```
else:  
    print("Is a word")
```

Enter something: 63fghd

Is not a word

```
In [12]: user_input = input("Enter a string: ")  
if user_input.isalpha():  
    print("Only alphabets")  
else:  
    print("Contains non-alphabet characters")
```

Enter a string: fg3

Contains non-alphabet characters

```
In [16]: user_input = input("Enter a number: ")  
if user_input.isnumeric():  
    print("Only number")  
else:  
    print("Contains non-numeric characters")
```

Enter a number: 786

Only number

30. Write a program to count the number of words in a sentence entered by the user.

```
In [18]: word_count = input("please enter a sentence: ").split()  
k = len(word_count)  
print(k)
```

please enter a sentence: raja debnath

2

31. How would you accept a date input from the user in Python?

```
In [19]: from datetime import datetime  
date_str = input("Enter a date (YYYY-MM-DD): ")  
date = datetime.strptime(date_str, "%Y-%m-%d")  
print("Entered date:", date)
```

Enter a date (YYYY-MM-DD): 2024-5-6

Entered date: 2024-05-06 00:00:00

37. Question: How would you accept and store multiple names from the user?

```
In [22]: word_count = input("please enter a sentence: ").split(",")  
print("Names:", word_count)
```

please enter a sentence: rraja, rana, akash, rima, siba

Names: ['rraja', ' rana', ' akash', ' rima', ' siba']

38. How would you extract numbers from a string entered by the user?

```
In [39]: import re
text = input("Enter a string: ")
numbers = re.findall(r'\d+', text)
print("Extracted numbers:", numbers)
```

```
Enter a string: jkl3 hi hello 143 j56
Extracted numbers: ['3', '143', '56']
```

re.findall The re.findall function in Python is part of the re module, which handles regular expressions. This function searches the string for all occurrences that match the given pattern and returns them as a list.

r"\d+": The r before the string indicates a raw string, which tells Python not to interpret backslashes as escape characters. \d: This is a regular expression pattern that matches any digit (0-9). +: This quantifier means "one or more" of the preceding element. So, \d+ matches one or more digits in a row. Example Here's a simple example to illustrate how re.findall(r'\d+', text) works:

```
import re

text = "There are 123 apples and 45 oranges."
```

Find all sequences of digits in the text

```
numbers = re.findall(r'\d+', text)
```

```
print(numbers) # Output: ['123', '45'] In this example, re.findall(r'\d+', text) finds all sequences of digits in the text and returns them as a list of strings.
```

I hope this helps! If you have any more questions or need further clarification, feel free to ask.

39. How do you find the maximum number from a list of integers entered by the user?

```
In [25]: no_list = input("Please enter numbers with spaces: ").split()
max_no = 0
for i in no_list:
    if int(i) > max_no:
        max_no = int(i)
print(max_no)
```

```
Please enter numbers with spaces: 20 30 44 52 65 95 85 45 75 65 35 84 89
95
```

40. How would you prompt the user for input until they enter a valid number?

```
In [28]:  
while True:  
    valid_number = input("please enter a number: ")  
    if valid_number.isnumeric():  
        break  
    else:  
        print("Please enter correct input")
```

```
please enter a number: 5d  
Please enter correct input  
please enter a number: 5f3  
Please enter correct input  
please enter a number: 56
```

```
In [30]:  
while True:  
    try:  
        num = int(input("Enter a valid number: "))  
        break  
    except ValueError:  
        print("Invalid input, please enter a number.")
```

```
Enter a valid number: 41  
Invalid input, please enter a number.  
Enter a valid number: 54
```

41. Write a program to check if the entered string has digits.

```
In [35]:  
number = '0123456789'  
valid_number = input("please enter a number: ")  
for i in valid_number:  
    if i in number:  
        print("string has at least one digit")  
        break  
    else:  
        print("string has no digit")  
        break
```

```
please enter a number: !!!  
string has no digit
```

```
In [41]:  
valid_number = input("please enter a number: ")  
for i in valid_number:  
    if i.isdigit():  
        print("string has at least one digit")  
        break  
    else:  
        print("string has no digit")  
        break
```

```
please enter a number: fgh  
string has no digit
```

```
In [38]: user_input = input("Enter a string: ")  
if any(char.isdigit() for char in user_input):  
    print("Contains digits")  
else:  
    print("No digits")
```

```
Enter a string: jkl  
No digits
```

```
In [79]: import string  
  
def is_digit_only(s):  
    return all(char in string.digits for char in s)  
  
print("12345 is only digit?", is_digit_only("12345"))  
print("123a5 is only digit?", is_digit_only("123a5"))
```

```
12345 is only digit? True  
123a5 is only digit? False
```

42. Write a program to check if the entered string has only whitespace characters.

```
In [44]: #if only space are entered. one or many spaces but only spaces.  
  
user_input = input("Enter a string: ")  
if user_input.isspace():  
    print("Only whitespace")  
else:  
    print("Contains non-whitespace characters")
```

```
Enter a string:  
Only whitespace
```

```
In [91]: import string  
def is_only_whitespace(s):  
    return all(char in string.whitespace for char in s)  
  
print(is_only_whitespace(" \t\n"))
```

```
True
```

43. Write a program to find the sum of all digits in a string entered by the user.

```
In [49]: user_input1 = input("Enter a number: ")  
nnumber_list = list(user_input1)  
sum_list = sum(int(i) for i in nnumber_list)  
print(sum_list)
```

```
Enter a number: 12546
```

```
18
```

```
In [50]: text = input("Enter a string: ")
digit_sum = sum(int(digit) for digit in text if digit.isdigit())
print("Sum of digits:", digit_sum)

Enter a string: k1456
Sum of digits: 15
```

44. Write a program that accepts a number and prints its absolute value.

```
In [55]: num = int(input("Enter a number: "))
print("Absolute value:", abs(num))

Enter a number: -10
Absolute value: 10
```

45. How would you check if a string entered by the user contains any uppercase letters?

```
In [57]: #check if all characters are upper case
text = input("Enter a string: ")
if text.isupper():
    print("upper case present")
else:
    print("no upper case")

Enter a string: ASD
upper case present
```

```
In [58]: #check if any letter is in upper case
text = input("Enter a string: ")
if any(char.isupper() for char in text):
    print("upper case present")
else:
    print("no upper case")

Enter a string: Asdf
upper case present
```

46. Write a program that converts Celsius to Fahrenheit.

```
In [64]: celcius = int(input("please enter celcius degree: "))
f = ((9*celcius)/5)+32
print(f"The fahrenheit is {f} of the {celcius} celcius")

please enter celcius degree: 36
The fahrenheit is 96.8 of the 36 celcius
```

47. Write a program to find the average of a list of numbers entered by the user.

```
In [65]: user_number = input("Please enter number with coma: ").split(",")
avg= (sum(int(no) for no in user_number))/(len(user_number))
print(avg)
```

```
Please enter number with coma: 9,6,3
6.0
```

```
In [73]: numbers = list(map(int, input("Enter numbers separated by space: ").split()))
print("Average:", sum(numbers) / len(numbers))
```

```
Enter numbers separated by space: 1 2 3
Average: 2.0
```

```
In [72]: k = ["1", "2", "3", "4", "5"]
k_1 = map(int, k)
print(list(k_1))
```

```
[1, 2, 3, 4, 5]
```

In Python, the `map()` function is a built-in function that allows you to apply a given function to all items in an iterable (like a list or tuple) and return a map object (which is an iterator). This is particularly useful for transforming data in a concise and readable way.

Here's a simple example to illustrate how `map()` works:

Example 1: Squaring Numbers

Define a function that squares a number

```
def square(num): return num ** 2
```

List of numbers

```
numbers = [1, 2, 3, 4, 5]
```

Use map to apply the square function to each item in the list

```
squared_numbers = map(square, numbers)
```

Convert the map object to a list and print it

```
print(list(squared_numbers)) Output:
```

[1, 4, 9, 16, 25] Example 2: Using Lambda Functions You can also use lambda functions with map() for more concise code:

List of numbers

```
numbers = [1, 2, 3, 4, 5]
```

Use map with a lambda function to square each number

```
squared_numbers = map(lambda x: x ** 2, numbers)
```

Convert the map object to a list and print it

```
print(list(squared_numbers)) Output:
```

[1, 4, 9, 16, 25] Example 3: Converting Strings to Uppercase

List of strings

```
words = ['hello', 'world', 'python']
```

Use map to convert each string to uppercase

```
uppercase_words = map(str.upper, words)
```

Convert the map object to a list and print it

```
print(list(uppercase_words)) Output:
```

['HELLO', 'WORLD', 'PYTHON'] The map() function is a powerful tool for applying transformations to data in a clean and efficient manner.

48 . Write a program to count the number of consonants in a string entered by the user.

```
In [78]: vowels="aeiou "
user_input2 = input("Please enter string: ")
consonants = sum(1 for char in user_input2 if char not in vowels)
print(consonants)
```

```
Please enter string: raja debnath
7
```

49. How do you check if a string entered by the user contains any punctuation?

```
In [97]: #with def
import string
def check_if_punctuation(s):
    return any(char in string.punctuation for char in s)

user_input3 = input("Please enter your string: ")
if check_if_punctuation(user_input3):
    print("Your string has punctuation")
else:
    print("No punctuation")
```

```
Please enter your string: 123!
Your string has punctuation
```

```
In [99]: #without def
user_input3 = input("Please enter your string: ")
if any(char in string.punctuation for char in user_input3):
    print("Your string has punctuation")
else:
    print("No punctuation")
```

```
Please enter your string: 123
No punctuation
```

```
In [80]: import string
print(string.digits)
```

```
0123456789
```

```
In [81]: import string
print(string.ascii_letters)
```

```
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
```

```
In [82]: import string
print(string.ascii_lowercase)
```

```
abcdefghijklmnopqrstuvwxyz
```

```
In [83]: import string
print(string.ascii_uppercase)
```

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

```
In [88]: import string
print(string.punctuation)
```

```
!"#$%&'()*+, -./:;=>?@[\\]^_`{|}~
```

```
In [89]: import string  
print(string.hexdigits)
```

0123456789abcdefABCDEF

```
In [90]: import string  
print(string.octdigits)
```

01234567

```
In [85]: import string  
print(string.capwords("raja debnath is a good boy"))
```

Raja Debnath Is A Good Boy

```
In [92]: import string  
print(string.whitespace)      #Formatter() takes no arguments
```

¶

```
In [94]: import string  
print(string.Formatter)      #Formatter() takes no arguments & it is place holder. use f  
  
from string import Formatter  
  
formatter = Formatter()  
formatted_string = formatter.format("Hello, {}!", "world")  
print(formatted_string)
```

<class 'string.Formatter'>
Hello, world!

50. Write a program that accepts a sentence and prints the longest word.

```
In [103...]: user_input4 = input("Please enter your string: ").split()  
longest = 0  
for word in user_input4:  
    if len(word) > longest:  
        k=""  
        longest = len(word)  
        k += word  
  
print(k)
```

Please enter your string: raja debnath is the best
debnath

```
In [104...]: text = input("Enter a sentence: ")  
words = text.split()  
longest_word = max(words, key=len)  
print("Longest word:", longest_word)
```

Enter a sentence: raja debnath is the best
Longest word: debnath

In [105...]

```
import keyword
print(keyword.kwlist)

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

The max() function in Python returns the largest item in an iterable or the largest of two or more arguments. It can be used with numbers, strings, and other objects.

Example

Finding the maximum value in a list

```
numbers = [3, 2, 8, 5, 10, 6] largest_number = max(numbers) print("The largest number is:", largest_number)
```

Output: The largest number is: 10

Usage with Multiple Arguments

The max() function can also take multiple arguments and return the largest one.

Finding the maximum value among multiple arguments

```
result = max(4, -5, 23, 5) print("The maximum number is:", result)
```

Output: The maximum number is: 23

Usage with Strings

When used with strings, max() returns the lexicographically largest string.

Finding the lexicographically largest string

```
languages = ["Python", "C Programming", "Java", "JavaScript"] largest_string = max(languages)
print("The largest string is:", largest_string)
```

Output: The largest string is: Python

Using Key Parameter

You can use the key parameter to specify a function to be called on each element before making comparisons.

Finding the longest string using key parameter

```
string_list = ["Geeks", "for", "Geek"] max_val = max(string_list, key=len) print(max_val)
```

Output: Geeks

Handling Empty Iterables

If the iterable is empty, you can provide a default value to avoid a ValueError.

Providing a default value for an empty iterable

```
dictionary = {} max_val = max(dictionary, default={1: "Geek"}) print(max_val)
```

Output: {1: 'Geek'}

Note: The max() function raises a TypeError if you try to compare items of different data types.