How to read the data from keyboard using python.

To read data from the keyboard (standard input or stdin) in Python, you can use the **input()** function. It reads a line of text from the standard input (stdin) and returns it as a string.

Example 1: Simple Input and Output

```
# Prompting the user to enter their name
name = input("Enter your name: ")

# Printing a greeting message using the input provided by the user
print("Hello, " + name + "! Welcome to Python programming.")
```

In this example, the **input()** function prompts the user to enter their name. The entered name is stored in the variable **name**, and then a greeting message is printed using that name.

Example 2: Simple Input and Output

```
data = input("Enter some data: ")
print("You entered:", data)
```

When you run this script, it will prompt you to enter some data. Once you've entered the data and pressed Enter, it will be stored in the variable **data**, and then it will print out what you entered.

Keep in mind that the **input()** function always returns a string. If you need to convert the input into a different type (like integer or float), you can use conversion functions like **int()** or **float()** accordingly.

For example:

```
number = int(input("Enter an integer: ")) # 20.50
```

This will read an integer from the user's input. If the input cannot be converted into an integer, it will raise a **ValueError** exception. So, it's a good practice to handle exceptions when converting input data to different types.

Example 2: Converting Input to Integer

```
# Prompting the user to enter an integer
num = int(input("Enter an integer: "))

# Doubling the entered integer and printing the result
result = num * 2
print("Double of", num, "is:", result)
```

In this example, the **input()** function is used to receive an integer input from the user. The **int()** function is then used to convert the input (which is a string) to an integer so that arithmetic operations can be performed.

Example 3: Handling Floating Point Input

```
# Prompting the user to enter a floating-point number
value = float(input("Enter a floating-point number: "))

# Squaring the entered number and printing the result
squared_value = value ** 2
print("Square of", value, "is:", squared_value)
```

In this example, the **input()** function is used to receive a floating-point number input from the user. The **float()** function is then used to convert the input (which is a string) to a floating-point number for further computation.

Example 4: Accepting Multiple Inputs

```
# Prompting the user to enter multiple values separated by commas
numbers = input("Enter multiple numbers separated by commas: ")

# Splitting the input string into a list of numbers
number_list = numbers.split(",")

# Converting each number in the list to an integer
integer_list = [int(num) for num in number_list]

# Finding the sum of the entered numbers and printing the result
total = sum(integer_list)
print("Sum of the entered numbers:", total)
```

In this example, the **input()** function is used to receive multiple numbers separated by commas from the user. The **split()** method is then used to split the input string into a list of numbers. Each number in the list is converted to an integer using list comprehension. Finally, the sum of the entered numbers is calculated and printed.

These examples demonstrate different use cases of the input() function in Python, allowing interaction with users through the command line interface.