Validation OTP for CodeTantra | CodeTantra Edu | Convert JPG to PDF. Images JP

csmssengg.codetantra.com/secure/topic-details1.jsp?cid=65969c70422c5e79de0c160e&tid=65969c71422c5e79de0c1695&bd=AMTc3M19jdF9jaA==

Q1

100%

L8/Input a... /Reading i...

In Python, to read input from the user, we have an in-built function called `input()`.

The syntax for input() function is :

```
input([prompt])
Here, the optional prompt string will be printed to the console for the user to input a value. The prompt string is used to indicate the type of value to be entered by the user.
```

Reading string inputs

Let us consider the below example:

Input1.py

Submit

```
1  place = input("Enter your favourite place: ")
2  #take your favourite place using input statement
3
4  # Print your favourite place
5  print("My favourite place is:",place)
```

Close Reset Submit

Type here to search

39°C Sunny

ENG US 5:03 PM 07/05/2024

Validation OTP for CodeTantra | CodeTantra Edu | Convert JPG to PDF. Images JP

csmssengg.codetantra.com/secure/topic-details1.jsp?cid=65969c70422c5e79de0c160e&tid=65969c71422c5e79de0c1695&bd=AMTc3M19jdF9jaA==

Q2

100%

L8/Input a... /Understan...

We already discussed this function to print output on Screen.

It is `print()` function, which is a built-in function in Python.

We can pass zero or more number of expressions separated by **commas(,)** to `print()` function.

The `print()` function converts those expressions into strings and writes the result to standard output which then displays the result on screen.

Let us consider an example of `print()` with multiple values with comma(,) as separator.

Print1.py

Submit

```
1 lang = input("Enter Language: ")
2 print("My Favourite Language is",lang)
```

Close Reset Submit

Type here to search

39°C Sunny

ENG US

5:03 PM 07/05/2024

Validation OTP for CodeTantra | CodeTantra Edu | Convert JPG to PDF. Images JP

csmssengg.codetantra.com/secure/topic-details1.jsp?cid=65969c70422c5e79de0c160e&tid=65969c71422c5e79de0c1695&bd=AMTc3M19jdF9jaA==

Q3

100%

L8/Input a... /% - forma...

The `print()` function is used to print the values to the standard output.

The general syntax of `print()` function is:

```
print(value1, value2..., sep = ' ', end = '\n', file = sys.stdout, flush = False)
```

where,

- `value1,value2...,valuen` These are the actual data values to be printed.
- `sep` : It is the separator used between each value. If there is no separator, then by default 'whitespace' is taken.

Formatspecifier.py

Submit

```
1  a=int(input("a: "))
2  b=int(input("b: "))
3
4  print(f'The value of a = {a}, b = {b} ')
5  a,b=b,a
6  print(f'The value of a = {a}, b = {b} ')
7
```

Close Reset Submit

Type here to search

39°C Sunny

ENG US 5:03 PM 07/05/2024

L../Input and Output Stat... /% - formatting and str.format()...

Follow the instructions given and write a program to print the user-given number in different formats using the format specifier.

If we want to limit the decimal places of a certain value, we use the format **{.nf}** where **n** is the **count** of the decimal places.

Example: If the **variable** is assigned with number **6** and the format is taken as **"{0:.4f}".format(variable)** then it results in **6.0000**.

Also, If we want to print a number **X** after **n** spaces, we use the format **{:(number of digits of X + n)d}.**

Let's say, we want to print the number **12** after **4** spaces, we use the format **{:(2 + 4)d}** which can be justified as the number of digits of 12 is 2 and we require 4 spaces.

More Examples:

- **print("{0:10d}".format(5))** will print **5** after **9** spaces/blanks.
- **print("{0:3d}".format(123))** will print **123** after **0** spaces/blanks.

Formatspecifier.py

Submit

```
1  # take float number from the user
2  a=float(input("a: "))
3  # print up to 2 decimal points
4  print("{0:.2f}".format(a))
5  # print up to 6 decimal points
6  print("{0:.6f}".format(a))
7  # take int number from the user
8  b=int(input("Enter b value: "))
9  # print the number with one space
10 print("{0:1d}".format(b))
11 # print the number with two spaces
12 print("{0:2d}".format(b))
13 # print the number with three spaces
14 print("{0:3d}".format(b))
15 # print the given number b in octal form
16 print(f'octal: {a}')
17 # print the given input b in hexadecimal form
18 print(f'hex: {b}')
19
```

Close Reset Submit