

Exercício – converter cadeias de caracteres em números e usar valores absolutos

8 minutos

Área restrita ativada! Tempo restante: 37 min

Você usou todas as 3 de 10 áreas restritas de hoje. Mais áreas restritas estarão disponíveis amanhã.

✓ Tempo de execução

Arquivo

Editar

Exibir

Comentários

▶ Executar tudo

py38_default

↺

↻

Create an application to work with numbers and user input

You'll frequently need to convert string values into numbers to properly perform different operations, or determine the absolute value of a number. In this exercise, you will create a project to calculate the distance between two planets based on user input.

This exercise is broken into a series of steps. For each step you will be presented with the goal for the step, followed by an empty cell. Enter your Python into the cell and run it. The solution for each step will follow each cell.

Read the values from the user

To create our application, we want to read the distance from the sun for two planets, and then display the distance between the planets. We'll do this by using `input` to read the values, `int` to convert to integer, and then `abs` to convert the result into its absolute value.

Start by adding the code to prompt the user for the distance between the sun and the first planet, and then the second. Store each result in variables named `first_planet_input` and `second_planet_input`.

```
# Enter code below
first_planet_input = input("Enter the distance from the sun form the first planet in km: ")
second_planet_input = input("Now, enter the distance from the sun form the second planet in km")
```

[2] ✓ 13 s

Now, enter the distance from the sun form the second planet in km:

Your code should resemble the following

```
first_planet_input = input('Enter the distance from the sun for the first planet in km')
second_planet_input = input('Enter the distance from the sun for the second planet in km')
```

Convert to number

Because `input` returns string values, we need to convert them to numbers. Add the code to convert each input into an integer using `int`. Store the values in `first_planet` and `second_planet` respectively.

```
# Enter code below
first_planet = int(first_planet_input)
```

```
second_planet = int(second_planet_input)
```

[3] ✓ <1 s

Your code should resemble the following:

```
first_planet = int(first_planet_input)
second_planet = int(second_planet_input)
```

Perform the calculation and convert to absolute value

With your values stored as numbers, you can now add the code to perform the calculation, subtracting the first planet from the second. Because the second planet might be a greater number, you'll use `abs` to convert it to an absolute value.

Subtract `first_planet` from `second_planet` and convert the result to its absolute value by using `abs`. Store the result in a variable named `distance_km`. Then display the result on the screen.

M ↴ ↵ 🔒 ⚙️ 🗑️



```
# Enter code below
distance_km = first_planet - second_planet
print(abs(distance_km))
```

[5] ✓ <1 s

... 558747509

Your code should resemble the following:

```
distance_km = second_planet - first_planet
print(abs(distance_km))
```

Test your application

To test your project, run the notebook. You'll be prompted in a dialog to provide the distances. You can use the ones from the following table:

Planet	Distance from sun
Mercury	57900000
Venus	108200000
Earth	149600000
Mars	227900000
Jupiter	778600000
Saturn	1433500000
Uranus	2872500000
Neptune	4495100000

Unidade seguinte: Verificação de conhecimentos

Continuar >
