Creating Functions

28 November 2024

18:52

We can use the def function to create (and define) our own function.

For example, Python doesn't have a built-in average function, so we can create our own:

A computer screen shot of a program code

Description automatically generated

We have now created the 'mean' function by defining what the function should do (in this case sum() / len())

Result:

A computer screen with white text

Description automatically generated

This can be seen by doing type(mean):

A computer screen shot of a program code

Description automatically generated

Result:

A black screen with white text

Description automatically generated

A screenshot of a computer program

Description automatically generated

Print or Return?

28 November 2024

19:19

The value of a function is always what you return

However, if you print this is not the case, as it will go line-by-line

Conditionals Intro

28 November 2024

19:24

A screen shot of a computer program

Description automatically generated

This function currently only returns an error as the "mean" defined function will not work with the dictionary. However, we can make this work with conditionals

A computer screen shot of a code

Description automatically generated

This code has added the "**if**" function, we are then using **== dict:** to say "if the variable is a dict, use this"

We are then using sum(**dict**.values()) / len(**dict**)

This will work out the average (**dict** is defined in your def function)

Once we are done with this part, we use **else:** to say "if this is not a dict, do this instead"

The above conditional example happens to be in a function, but conditionals can also be outside of functions, example:

A screen shot of a computer code

Description automatically generated

This will check if true

e.g. the same as:

>>>

if True:

print("Greater")

else:

print("Not Greater")

\*\*\* Greater

Also better to use this intead of if type():

if isinstance(3, int)

Boolean Operators

28 November 2024

19:46

So far, you learned how to check for one single condition:

* 1. x = 1
  3. if x == 1:
  4. print("Yes")
  5. else:
  6. print("No")

You can also check if two conditions are met at the same time using an and operator:

* 1. x = 1
  2. y = 1
  4. if x == 1 and y==1:
  5. print("Yes")
  6. else:
  7. print("No")

That will return Yes since x == 1 and y ==1 are both True.

You can also check if one of two conditions are met using an or operator:

* 1. x = 1
  2. y = 1
  4. if x == 1 or y==2:
  5. print("Yes")
  6. else:
  7. print("No")

That will return Yes since at least one of the conditions is True. In this case x == 1 is True.

Elif Conditionals

28 November 2024

20:05

If you have additional filtering to be done, you can use "elif" to add conditions, below is an example:

A computer screen with text and numbers

Description automatically generated