Min () function in Python

The built-in min () function in Python returns the smallest element in an iterable or smallest of passed parameters. So, how does the syntax of min () function work, what parameters and how does it work?

The built-in min () function in Python returns the smallest element in an iterable or smallest of passed parameters.

If the values ??are strings, they will compare alphabetically.

So, how does the syntax of min () function work, what parameters and how does it work? Invites you to read the track.

The syntax of min () function in Python

The min () function in Python has two forms:

```
min(iterable, *iterables[,key, default])
```

Or:

```
min(item1, item2, *item[, key])
```

Parameters of min function ()

The min () function works with two types of parameters corresponding to the two syntaxes mentioned above:

1. `min(iterable, *iterables[, key, default])`

1. `iterable` : Required. The tuples, string, set, dictionary or iterator objects that you need to find the smallest element in.
2. `*iterables` : Optional. The smallest Iterable will be returned.
3. `key` : Optional. Key function, where the iterable goes through. Comparisons are made based on the results returned after passing the key function.
4. `default` : Optional. The default value when iterable is empty.

2. `min(item1, item2, *item[, key])`

1. `item1` , `item2` : Required. Object to compare, can be number, string .
2. `*item` : Optional. Other objects for comparison.

Picture 2 of Min () function in Python

At least two objects are needed to make comparisons with the min () function.

3. `key` : Optional. Key function, where items pass. The comparison is performed on the result returned after passing the key function.

Return value from min ()

The min () function returns different results corresponding to the two types as above:

1. `min(iterable, *iterables[, key, default])`

Case
Key
Default
Return value
Iterable empty Yes or No No Exception Exception **ValueError** Iterable blank Yes Yes Returns Default value An iterable (not empty) No or No Returns the smallest number in iterable One iterable (blank) Yes or No Transmission each element in iterable for the key function, the result returned is the smallest element based on the return value from the key function Multiple iterable (not empty) No Yes or No Returns iterable smallest Multiple iterable (not empty) Yes Yes or Don't pass each iterable to the key function. The result returned is the smallest iterable based on the value returned from the key function

2. `min(item1, item2, *item[, key])`

Case
Key
Return value
2 item No Return parameter smaller than 2 item Yes Pass each parameter for the key function, the result returned is smaller element based on the return value from the key function Many items No Returns the smallest parameter Many items Yes Pass each parameter to the key function, the result returned is the smallest element based on the value returned from the key function

Example 1: Find the smallest element in the number passed

```
# su dung min(item1, item2, *item) print('So nho nhat la:', min(1, 3, 2, 5, 4)) #
```

Run the program, the result is:

```
So nho nhat la: 1 So nho nhat la: 2
```

Example 2: Find the number that has the smallest number of digits using the key function

```
def sumDigit(num): sum = 0 while(num): sum += num % 10 num = int(num / 10) retur
```

The return output is:

```
Ket qua nho nhat la: 100 Ket qua nho nhat la: 10
```

In this example, the parameters or each element in the iterable parameter are passed in turn to `sumDigit()` to get the result that the number has the smallest sum of digits.

Example 3: Find the list with the smallest length using the key function

```
num = [15, 300, 2700, 821] num1 = [12, 2] num2 = [34, 567, 78] # su dung min(ite
```

Results returned:

```
List ngan nhat la: [12, 2]
```

In the above program, iterable num, num1 and num2 passed into the key function are built-in len () functions in Python. The result is iterable, and the min () function will iterable with the smallest length.

See also: Built-in Python functions

You finished reading the article "**Min () function in Python**" edited by the **TipsMake** team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.

© 2019 TipsMake.com