

Python Range Float

Python Range Float

In Python range() tutorial, we have seen that we can create an iterable for a sequence of integers. That is because range() function accepts only integers for start, stop and step arguments.

So, to generate Range of floating point numbers, we have to either workaround range() function to generate sequence of floating point numbers. We shall look into an example for this workaround.

Also, there is a function in numpy() library, arange() function, that can generate a sequence of floating point numbers. We will also look into this with some examples.

Python Range of Floating Numbers using numpy.arange()

In this example, we will use numpy.arange() function to generate an iterator to the sequence of floating point numbers.

Python Program

```
import numpy

start = 0
stop = 0.9
step = 0.1

for i in numpy.arange(start, stop, step) :
    print(i)
```

Output

```
0.0
0.1
0.2
0.30000000000000004
0.4
0.5
0.6000000000000001
0.7000000000000001
0.8
```

Python Range of Floating Numbers using range()

In this example, we will use range() function to create a sequence of floating point numbers. The idea is that, we shall multiply with some integer, say scaling factor, that would make these floating point values to integers, and then we shall generate a range() object. During iteration, if we divide each element with the same scaling factor, we get sequence of floating point numbers.

Python Program

```
start = 0
stop = 0.9
step = 0.1

scale = 10

for i in range(int(scale*start), int(scale*stop), int(scale*step)) :
    print(i/scale)
```

Output

```
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
```

You may have to change the scale based on the start, stop, and step values. Or you may write a function to calculate an optimized scale based on these start, stop and step values. Or you can always take a very high scale value like 1000000000000.

Conclusion

In this [Python Tutorial](#), we learned how to create a Python Range with floating point values.

Python Programming

‡ [Python Tutorial](#)

‡ [Install Python](#)

‡ [Install Anaconda Python](#)

‡ [Python HelloWorld Program](#)

‡ [Python Variables](#)

‡ [Python Variable Data Type Conversion](#)

‡ [Python Comments](#)

Control Statements

‡ [Python If](#)

‡ [Python If Else](#)

‡ [Python While Loop](#)

‡ [Python For Loop](#)

Python String

‡ [Python String Methods](#)

‡ [Python String Length](#)

‡ [Python String Replace](#)

‡ [Python Split String](#)

‡ [Python Count Occurrences of Sub-String](#)

‡ [Python Sort List of Strings](#)

Functions

‡ [Python Functions](#)

Python Collections

‡ [Python List](#)

‡ [Python Dictionary](#)

Advanced

‡ [Python Multithreading](#)

Useful Resources

‡ [Python Interview Questions](#)