Continue

Nov 12, 2022 6 Minute Read Why Trust Us We uphold a strict editorial policy that emphasizes factual accuracy, relevance, and impartiality. Our content is crafted by top technical writers with deep knowledge in the fields of computer science and data science, ensuring each piece is meticulously reviewed by a team of seasoned editors to guarantee compliance with the highest standards in educational content creation and publishing. By Mradula Mittal Python is an easy-to-learn language. As a beginner, you must be well-equipped with loops within Python. In this article, we will learn how to create a multiplication table in python, to understand the basics of loops. You'll also learn to use lambda functions, as a replacement for loop here. So, let's get started! How to Make a Multiplication Table in Python? Before jumping into different ways, let's take a look at our problem statement. Problem Statement: Create a Multiplication Table for any number in Python Example: Let's say you need to generate and display a multiplication table for a number (say 4) in Python. The desired output must be like this: 5 x 1 = 5 5 x 2 = 10 5 x 3 = 15 5 x 4 = 20 5 x 5 = 25 5 x 6 = 30 5 x 7 = 35 5 x 8 = 40 5 x 9 = 45 5 x 10 = 50 or [4, 8, 12, 16, 20, 24, 28, 32, 36, 40] Let's move on to obtaining these solutions. 01) Using For loop The for loop is used to repeat a block of code a specified number of times. Hence, the for loop works within a range. The basic syntax of for loop is: for reference_variable_name in (iterable or range):{ code } Let's create a multiplication table for a number using 'for loop' in Python: # Program: Multiplication Table in Python # number num = 5 # let's take a syntax for our table - num x (1 - 10) = num*(1-10) # Since we're taking the table to 10, hence we'll iterate it 10 times print("The multiplication table of ", num) for i in range(1, 11): print(f" {num} x {i} = {num*i}") Output: The multiplication table of 5 5 x 1 = 5 5 x 2 = 10 5 x 3 = 15 5 x 4 = 20 5 x 5 = 25 5 x 6 = 30 5 x 7 = 35 5 x 8 = 40 5 x 9 = 45 5 x 10 = 50 The range() function, as the name suggests, returns a sequence of numbers ranging from one number to another. The basic syntax of the range() function is: range(initial, last, step) with the default values of the initial and step being 0 and 1, respectively. Also, note that I've used "11" in the range() function above, since range() in Python does not include the last number, i.e. it provides a sequence of one less than the last digit. 02) Using a while loop The while loop helps us execute a statement as long as a condition is true. You can also use a while loop to keep count of a sequence (basically a for loop with more lines of code). Let's try it out: # Program: Multiplication Table in Python # number num = 5 # let's take a syntax for our table - num x (1 - 10) = num*(1-10) # Since we're taking the table to 10, hence we'll iterate it 10 times print("The multiplication table of ", num) # initialize i for range count = 1 while count