Example fruits["apple", "banana", "cherry"] for x in fruits: print(x) The for loop does not require an indexing variable to set beforehand. Looping Through a String Even strings are iterable objects, they contain a sequence of characters: for x in "banana": print(x) The range() Function To loop through a set of code a specified number of times, we can use the range() function, The range() function returns a sequence of numbers, starting from 0 by default, and increments by li=1 while i6: print(i) i+=1 Python Conditions and If statements Python supports the usual logical conditions from mathematics: Equals: ab Not Equals: ab Less than: a b Less than or equal to a b Greater than: ub Greater than or equal to: a b Example In this example we use two variables, a and b. which are used as part of the if statement to test whether b is greater than a. As a is 33, and b is 200, we know that 200 is greater than 33, and so we print to screen that "b is greater than a". b−200 ifba: print("h is greater than a")however it is possible to specify the increment value by adding a third parameter: range(2, 30, 3): The while Loop With the while loop we can execute a set of statements as long as a condition is true.The range() Function To loop through a set of code a specified number of times, we can use the range() function, The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.for x in range(6): print(x) The range() Function The range() function defaults to 0 as a starting value, however it is possible to specify the starting value by adding a parameter:range(2, 6)., which means values from 2 to 6 (but not including 6): for x in range(2, 6): print(x) The range() Function The range() function defaults to increment the sequence by 1.(by default), and ends at a specified number.