

List Comprehensions

 coursera.org/learn/python-crash-course/supplement/jRtIG/list-comprehensions

You can create lists from sequences using a for loop, but there's a more streamlined way to do this: **list comprehension**. List comprehensions allow you to create a new list from a sequence or a range in a single line.

For example, `[x*2 for x in range(1,11)]` is a simple list comprehension. This would iterate over the range 1 to 10, and multiply each element in the range by 2. This would result in a list of the multiples of 2, from 2 to 20.

You can also use conditionals with list comprehensions to build even more complex and powerful statements. You can do this by appending an if statement to the end of the comprehension. For example, `[x for x in range(1,101) if x % 10 == 0]` would generate a list containing all the integers divisible by 10 from 1 to 100. The if statement we added here evaluates each value in the range from 1 to 100 to check if it's evenly divisible by 10. If it is, it gets added to the list.

List comprehensions can be really powerful, but they can also be super complex, resulting in code that's hard to read. Be careful when using them, since it might make it more difficult for someone else looking at your code to easily understand what the code is doing.