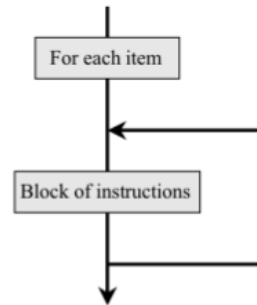


Model 1 for Statements

A **for** loop executes the same block of code “for each item in a sequence”. Create a new file named `loops.py`, and enter the following code:

```
print("hello")
for x in [2, 7, 1]:
    print("the number is", x)
print("goodbye")
```



Questions (15 min)

Start time:

1. Run the `loops.py` program. How many times does the indented line of code execute under the **for** loop?
2. How many times does the line of code NOT indented execute after the **for** loop?
3. Identify the value of `x` each time the indented line of code is executed.
 - a) 1st time:
 - b) 2nd time:
 - c) 3rd time:
4. Modify the list `[2, 7, 1]` in the following ways, and rerun the program each time. Indicate how many times the **for** loop executes.
 - a) non-consecutive numbers: `[5, -7, 0]`
 - b) numbers decreasing in value: `[3, 2, 1, 0]`
 - c) all have the same value: `[4, 4]`
 - d) single value in a list: `[8]`

5. In general, what determines the number of times that the loop repeats?
6. What determines the value of the variable `x`? Explain your answer in terms of what is assigned (`x = ...`) each time the loop runs.
7. Modify the program as follows:
 - a) Write a statement that assigns `[0, 1, 2, 3, 4]` to the variable `numbers`.
 - b) Rewrite the `for x ...` statement to use the variable `numbers` instead.
 - c) Does the assignment need to come before or after the `for` statement?
8. Add the following code at the end of your program:

```
for c in "Hi!":  
    print(c)
```

 - a) What is the output of this `for` statement?
 - b) What determined how many times `print(c)` was called?
 - c) Explain what a `for` statement does with strings.
9. What other data types (besides lists and strings) can a `for` loop handle? Experiment by adding examples to your `loops.py` program. Summarize here what works and what doesn't.

Model 2 The `range` Function

The Python `range` function will generate a list of numbers. The `range` function can take up to three numbers as arguments. Fill in the table below by typing the code into a Python Shell:

Python code	Shell output
<code>range(5)</code>	
<code>list(range(5))</code>	
<code>x = range(3)</code>	
<code>print(x)</code>	
<code>print(list(x))</code>	
<code>list(range(5, 10))</code>	
<code>list(range(-3, 4))</code>	
<code>list(range(4, 10, 2))</code>	
<code>for i in range(5): print(i)</code>	

Questions (15 min)

Start time:

10. Explain the difference in output between the first two lines of code (with and without the `list` function).

11. If the argument of the `range` function specifies a single number (x):

- What will be the first number listed?
- What will be the last number listed?
- How many numbers will be in the list?
- Use the range function to generate the sequence 0, 1, 2, 3.

12. If the argument of the `range` function specifies two numbers (x, y):

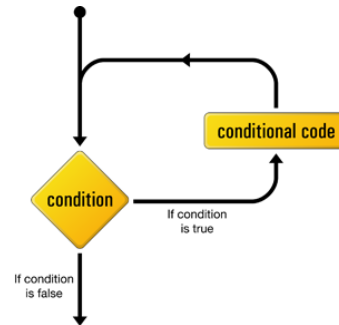
- What will be the first number listed?
- What will be the last number listed?
- How many numbers will be in the list?
- Use the range function to generate the sequence 1, 2, 3, 4.

13. If the argument of the `range` function specifies three numbers (x, y, z) :
- a) What will be the first number listed?
 - b) What does the third argument represent?
 - c) How many numbers will be in the list?
 - d) Use the range function to generate the sequence 1, 3, 5, 7.
14. In your Editor, make a copy of the Model 1 code. Then modify the `for` statement so that the number of times the loop executes is determined by a variable named `times`.
- a) How did you change the `for` statement?
 - b) How would you cause the loop to print the values 0 to 5?
15. Consider the two different types of `for` statements used in Model 1 and Model 2.
- a) If you wanted to execute a loop 100 times, which type of `for` statement would you choose and why?
 - b) If you wanted to use each item of an existing list inside the loop, which type of `for` statement would you choose and why?
16. Does the `range` function work with strings? If so, show an example. If not, show how to print the letters A to Z in a loop.

Model 3 while Statements

A more general looping structure is the `while` statement. Add the code below to your current `loops.py` program:

```
i = 0
while i < 3:
    print("the number is", i)
    i = i + 1
print("goodbye")
```



Questions (15 min)

Start time:

17. What must the value of the Boolean expression (after the `while`) be in order for the first print statement to execute?
18. Circle the statement that changes the variable `i` in the above code.
19. What happens to the value of the variable `i` during the execution of the loop?
20. Explain why the loop body does not execute again after it outputs “the number is 2”.
21. Reverse the order of the statements in the loop body:

```
while i < 3:
    i = i + 1
    print("the number is", i)
```

 - a) How does the order impact the output displayed by the `print` function?
 - b) Does the order impact the total number of lines that are output?

22. Identify three different ways to modify the code so that the loop only executes twice.

23. Describe the three parts of a `while` loop that control the number of times the loop executes.

24. Comment out the statement `i = i + 1`, and run the module. Then press Ctrl-C (hold down the Ctrl key and press C). Describe the behavior you see, and explain why it happened.

When writing a `while` loop, it's helpful to answer a few questions before you start:

- *What needs to be initialized before the loop?*
- *What condition must be true for the loop to repeat?*
- *What will change so that the loop eventually ends?*

25. Consider the function `add(n)` that prompts the user for n numbers and returns the sum of these values. For example, when `add(5)` is called, the user is asked to input five numbers. If the user inputs 3, 1, 5, 2, and 4, the function would return the value 15.

- Describe the variable that needs to be initialized before the loop begins.
- Describe the Boolean expression that must be true for the loop to continue.
- Describe what will need to change so that the loop will eventually end.
- Now list what else needs to happen inside the body of the loop for you to calculate the sum of the user input.
- Given your previous answer, are there any other values that need to be initialized before the start of the loop?