

## Looping in Python (cont'd)

### While loop:

We can also iterate using while loops. They continue to execute while a certain condition remains truthy and will end when they become falsey.

### Controlled Exit:

The keyword **break** is used to give us the ability to exit out of while loops whenever we want.

#### While Loop Exercise

While loops are really hard for me to write tests for (until we learn about things like lists), so this exercise is a bit tricky to explain. :( Here's the main idea:

Use a `while` loop to generate a random number between 1 and 10 until you get the number 5. Every time the loop runs, increment the variable `i`.

Here are more detailed steps:

- Generate a random number between 1 and 10 using `randint(1, 10)`, storing the result in the `number` variable
- Write a while loop to keep regenerating a new random number between 1 and 10 while the random number is not equal to 5.

exercise.py

```
1 from random import randint # use randint(a, b) to generate a random number between
  a and b
2
3 number = 0 # store random number in here, each time through
4 i = 0 # i should be incremented by one each iteration
5
6 while number != 5:
7     number = randint(1, 10)
8     print(number)
9     i += 1
10    print(i)
```

Line 10, Column 12 All changes saved

Reset code

✓ Well done, your solution is correct!

Check solution

Continue

🔍 📄

## Lists

**What is a list?** It is a collection or grouping of items. It's a way of combining data into one variable or one structure.

Objectives:

- Describe, create, and access a list data structure
- Use built-in methods to modify and copy lists
- Iterate over lists using loops and list comprehensions
- Work with nested lists to build more complex data structures

**How are lists useful?** A fundamental data structure for organizing a collection of items.

**How many elements exist?** Let's use the first built-in function for lists—`len`, aka length.

## Creating Lists Exercise

It's time to create your own lists. **Define 2 separate lists:**

1. First, define a list called `my_stuff`. It must be at least 4 elements long. The data is completely up to you, but it must contain at least 1 string and 1 float.
2. Next, define a list called `nums`. It should be a list containing all the numbers between 1 and 99 (including 99). Don't type this out manually! Use what we learned at the end of the previous video!

exercise.py	<pre>1 # Define my_stuff 2 my_stuff = [1.99, 'sofa', 'desk', 'chair'] 3 # Define nums 4 nums = list(range(1, 100))</pre>	✓ Well done, your solution is correct!
Line 2, Column 17 All changes saved		Reset code
Check solution		Continue

## Accessing values in a list:

Like ranges, lists ALWAYS start counting at 0. Therefore, the first element will have an index of 0.

To check if a value is in a list, we use the keyword **in**.

## Accessing List Data

I'm having a party, and made a list of people I want to invite. Unfortunately, I'm a terrible friend and made a couple of spelling errors. Help me correct them!

- Change "Hanna" to "Hannah" (there should be an 'h' at the end)
- Change "Geoffrey" to "Jeffrey"
- Change "aparna" to "Aparna" (capitalize it)

**Hint:** You can use the following syntax to change a value of an existing list element at the specific index position:

```
list[index] = "NewValue"
```

exercise.py	<pre>1 # DON'T TOUCH THIS PLEASE! 2 people = ["Hanna", "Louisa", "Claudia", "Angela", "Geoffrey", "aparna"] 3 # DON'T TOUCH THIS PLEASE! 4 5 #Change "Hanna" to "Hannah" 6 people[0] = 'Hannah' 7 #Change "Geoffrey" to "Jeffrey" 8 people[4] = 'Jeffrey' 9 #Change "aparna" to "Aparna" (capitalize it) 10 people[5] = 'Aparna'</pre>	✓ Well done, your solution is correct!
Line 10, Column 9 All changes saved		Reset code
Check solution		Continue