

[◀ Back to Week 2](#)[✕ Lessons](#)[Prev](#)[Next](#)

Python contains many container types that can hold other elements. The **list** is one of the most basic and most commonly used. You will rarely, if ever, see a Python program that does not use lists.

List Literals

You can create a list in your program by putting a comma separated sequence of items between square brackets, as follows:

```
1 lst1 = [1, 5, 9, -32]
2 print(lst1)
3
4 lst2 = ["dog", "cow", "horse"]
5 print(lst2)
6
7 emptylst = []
8 print(emptylst)
```

Note that there can be zero or more elements in the list and the elements of the list can be anything you would like. It is generally good practice to only include elements of the **same** type as each other within a list. You will see other programmers violate this practice. You should not. It leads to bugs and confusion when you (the programmer) have to remember the different types of the elements within a single list.

When you print a list in Python, it appears identically to how you would define it as a list literal.

Range

Python has a builtin function, called **range**, which allows you to easily construct sequences of integers. The function **range** does not return a list, rather it returns a special *range object*. However, you can pass it to the **list** function to convert the range object into a list, if you need or want a list instead. The **range** function takes from 1 to 3 arguments. These arguments are converted into a *start*, *stop*, and *step* value as follows:

- If a single argument is passed to **range**, the start value will be 0, the stop value will be the argument, and the step value will be 1.
- If two arguments are passed to **range**, the start value will be the first argument, the stop value will be the second argument, and the step value will be 1.
- If three arguments are passed to **range**, the start value will be the first argument, the stop value will be the second argument, and the step value will be the third argument.

The range will then contain the sequence of integers which starts at the start value. Each successive value in the range will be the previous value in the range plus the step value. The range will continue up to (or down to, if the step is negative), but not including, the stop value.

List Indexing

You access elements of a list using their index. The first element of a list is located at index 0 (not 1!). The last element of a list named **lst** is located at index **len(lst) - 1** (not **len(lst)**). The syntax to access an element of a list is to place the index in square brackets after the variable that refers to a list.

```
1 lst = list(range(10))
2
3 # First element
4 print(lst[0])
5
6 # Third element
7 print(lst[2])
8
9 # Length
10 print(len(lst))
11
12 # Last element
13 print(lst[9])
14 print(lst[len(lst) - 1])
15 print(lst[-1])
```

Note that you can also access elements of a list using negative indices. The last element of a list is located at index -1. The second to last element is located at index -2. And so on.

List Slicing

Python allows you to extract a sublist from a list using *slicing*. When you slice a list you access it with two indices separated by a colon. This will produce a new list with the elements with indices starting at the first index up to, but not including, the second index. If the index before the colon is missing, the slice starts at the first element of the list. If the index after the colon is missing, the slice ends at the last element of the list.

```
1 lst = list(range(10))
2
3 # Slice with 3 elements
4 print(lst[4:7])
5
6 # Slice with the last 2 elements of the list
7 print(lst[8:10])
8 print(lst[8:])
9 print(lst[-2:])
10
11 # Slice with the first 4 elements of the list
12 print(lst[0:4])
13 print(lst[:4])
14
15 # Empty slices
16 print(lst[20:25])
17 print(lst[7:3])
```

Negative indices may be used in slicing, and they have exactly the same meaning that they did when used as indices. You can even mix positive and negative indices in a slice.

Rather than causing errors, if there are no elements in the list between the indices in the slice, then an empty list is produced.