

PALs CSC-152

Section 02, Professor Kim, M-W-F

Tony Stone

November 12, 2025

Lesson II

1 The List Data Type

A **list** is a **collection** data type. It is a collection of various elements that can be stored in a single variable. In Python, we specify a list using brackets: "[]" and separate elements with commas ",". We will see many examples below!

1.1 List Operations

Sometimes we would like to access the items that are in a list. We can do that with **indexing**. In Python, and practically all programming languages, indexing begins at 0. So, the index of the first element is 0, the second is 1, the third is 2, etc. Here are some examples of indexing:

```
1 alist = ["my", "name", "is", "Tony"]
2 print(alist[0])
3 print(alist[2])
4 print(alist[3])
5 print(alist[-1])
```

Output:

```
1 my
2 name
3 is
4 Tony
5 Tony
```

Now that we have a list, the question is how are we going to use the list? When we index a list, we get that element, which we can then store into a variable to use for further operations.

```
1 nums = [1, 2, 3, 4, 5]
2 sumTwo = nums[3] + nums[4] # 4 + 5
3 print(sumTwo)
```

Output:

```
1 9
```

In addition to indexing, Lists have an additional operation known as **slicing**. This is very similar to indexing, only rather than indexing a single element, slicing indexes a range of elements. This operation returns a sub-list where the elements correspond to the index range. This range is specified by a colon, where the first number is the start of the slice, and the second number is the end. It is important to note that the first number is **inclusive** and the second is **exclusive**.

```

1 alist = ["my", "name", "is", "Tony"]
2 print(alist[1:4])

```

Output:

```

1 ["name", "is", "Tony"]

```

As you can see in this example, the indices of our list only goes up to 3, but because the second number of our range is exclusive, we use 4 to extract all elements from the list.

Just like with strings, we can concatenate lists! We use the "+" operator, in order to combine them. Lets look at some examples, using all the tricks with lists that we've learned so far:

```

1 nums = [1, 2, 3, 4, 5]
2 nums = nums + [6]
3 print(nums)
4 nums = nums + nums
5 print(nums)
6 newNums = nums[0:2]+nums[10:12]
7 print(newNums)

```

Output:

```

1 [1, 2, 3, 4, 5, 6]
2 [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6]
3 [1, 2, 5, 6]

```

1.2 List comprehension

List comprehension is a **Pythonic** way of creating lists. Do not feel intimidated by it's appearance. When we explore control structures and list creation in general, this will seem like a piece of cake. For now, let's just admire one of the many things that makes Python so special:

```

1 alist = [i for i in range(5)]
2 print(alist)

```

Output:

```

1 [0, 1, 2, 3, 4]

```

2 Key Terms

- **Collection:** A grouping of elements, with restrictions on inclusion based off of the specific type of collection.
- **List:** A collection data type used to store multiple items in a single variable.

- **Indexing:** Retrieving an item from a collection data type based off of the desired item's position.
- **slicing:** Indexing a collection with an index range, rather than a single index.
- **Pythonic:** Implementations that follow the conventions/quirks found in Python

3 Availability

- In Lecture:
 - Wednesday: 10:00am - 11:00am
- PAL Sessions:
 - Monday: 2:00pm - 3:00pm
 - Wednesday: 9:00am - 10:00am
 - Wednesday: 2:00pm - 3:00pm

References

- [1] Tony Gaddis. *Starting Out with Python*. Pearson, 5th edition, 2021.
- [2] GeeksforGeeks. Python Tutorial — Learn Python Programming Language - GeeksforGeeks — [geeksforgeeks.org. `https://www.geeksforgeeks.org/python/python-programming-language-tutorial/`](https://www.geeksforgeeks.org/python/python-programming-language-tutorial/). [Accessed 29-10-2025].
- [3] Bradley N. Miller and David L. Ranum. Problem solving with algorithms and data structures using python. [`https://runestone.academy/ns/books/published/pythonds/index.html`](https://runestone.academy/ns/books/published/pythonds/index.html), 2014. Interactive online edition — accessed: 2024.
- [4] Python Software Foundation. *Python: A dynamic, open source programming language*, 2025. Version 3.x documentation.
- [5] W3Schools. W3Schools.com — [`https://www.w3schools.com/python/default.asp`](https://www.w3schools.com/python/default.asp). [Accessed 29-10-2025].