Chapter 5 - Lists

Lists

List Basics

Learning Objectives - List Basics

- **Describe a list**
- **Identify what data can be stored in a list**
- **Populate a list with a for loop**
- **Use an index to access an element**
- **Use an index to modify elements in a list**

List Basics 1

What is a List?

Lists are a built-in data structure that groups information together. Lists do not have to be of the same data type, but often are. Lists are declared with the `[]` brackets, and a comma separates each item in a list. Lists are mutable, which means you can alter them in a variety of ways.

```
int_list = [1, 2, 3, 4, 5]
string_list = ["John", "Paul", "George", "Ringo"]
mixed_list = [0.87, "hello", True, 17]

print(int_list)
print(string_list)
print(mixed_list)
```

challenge

What happens if you:

- Remove the commas from a list?
- Remove the quotes from around `"hello"`?
- Declare a variable `hello` and assign it the value `"hi"` (`hello = "hi"`)?

Populating a List

You can use the `range` function to create a sequence of numbers for a list. The syntax is slightly different from a for loop. There is an extra `i` before the for loop.

```
my_list = [i for i in range(1, 51)]

print(my_list)
```

`my_list` is composed of the variable `i`, which is defined by all numbers from 1 up to but not including 51 (so all integers from 1 to 50).

challenge

What happens if you:

- Change the range to `range(50)`?
- Change the range to `range(0, 50, 2)`?
- Change the range to `range(50, 0, -1)`?

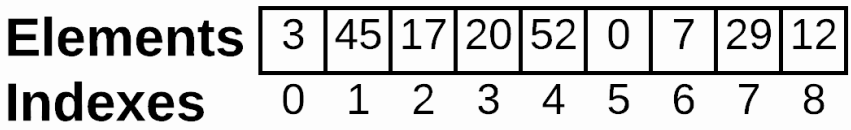
▼ **The Empty List**

There is a special list called an empty list. This is a list that has no elements. An empty list looks like this: `my_list = []`. We will see how to add elements to an empty list in a later lesson.

List Basics 2

List and its Index

Each piece of data in a list is called an element. You can access elements of a list with an index, which is like an address for each element. Start counting with index 0. The first element in the image below would be `my_list[0]`.



List Indices

```
my_list = [5, 10, 15, 20]

print(my_list[0])
```

challenge

What happens if you:

- Change the index to 2 (`print(my_list[2]`)?
- Change the index to 4 (`print(my_list[4]`)?
- Change the index to -1 (`print(my_list[-1]`)?
- Change the index to 1.5 (`print(my_list[1.5]`)?

▼ **Negative Indexes**

Python allows you to use a negative number as an index. `-1` is the last element of the list, `-2` is the second to last, etc. It is possible to generate an error message with a negative index. Try the code above with `-5` as the index.

Modifying a List

You can use the assignment operator (`=`) to change the value of an element. Be sure the reference the element with the appropriate index.

```
my_list = [1, 2, 3]
print(my_list)

my_list[0] = 4
my_list[1] = 5
my_list[2] = 6
print(my_list)
```

challenge

What happens if you:

- Change an assignment to be `my_list[0] = "hello"`?
- Change an assignment to be `my_list[0] = 5 % 2 > 0`?
- Change an assignment to be `my_list[0] = my_list[2]`?