

CSSE 120 – Introduction to Software Development (Robotics section)

Concept: Sequences and Indexing

A **sequence** is a type of thing in Python that represents a finite, ordered **collection** of things **indexed** by whole numbers. For example:

- A **list**: `['red', 'white', 'blue']`
- A **string**: `'Check out Joan Osborne, super musician'`
- A **tuple**: `(800, 400, 310)`

There are also types for *unordered* collections, for example, **sets** and **dictionaries**.

Sequences are powerful because they let you refer to an entire collection, as well as the items in the collection, using a *single name*.

- You can still get to the *items* (aka *elements*) in the collection, by **indexing**:

```
colors = ["red", "white", "blue"]
colors[0] has value "red"
colors[1] has value "white"
colors[2] has value "blue"
```

Indexing starts at ZERO, not at one

- And you can **loop through the items** in the collection, like this:

```
for color in colors:
    circle = zg.Circle(...)
    circle.setFill(color)
```

The number inside the square brackets is called the INDEX.

or like this:

```
for k in range(len(colors)):
    circle[k] = zg.Circle(...)
    circle[k].setFill(color)
```

The len function returns the LENGTH of the sequence, that is, the number of items in the sequence.

The sequence types that we will use most often are **list**, **string (str)** and **tuple** (and also **range**, but we will use **range** only in **for** statements).

- A **list** can contain objects of any type and is *mutable*. Literals are written using square brackets:

```
[45, 87.0, zg.Point(400, 30)]
```

- A **tuple** can also contain objects of any type, but is *not* mutable. Literals are written using parentheses:

```
(45, 87.0, zg.Point(400, 30))
(57,)
```

Notation for a SINGLE item in a tuple. Also, for tuples with more than one item you can omit the parentheses.

- A **string** can contain only Unicode characters and is *not* mutable. Literal strings are written using single or double quotation marks (either is fine):

```
'hi there'      "what's up doc?"
```

We'll discuss **mutability** in a forthcoming session.