

## Lesson Two

Math 6080 (for the Masters Teaching Program), Summer 2020

**Part 1. Strings in Python.** Strings in Python are sequences of characters.

Python knows you give it a string when you enclose the text in (single or double) quotation marks (next to the return key). For example, if you type HAPPY at the Python prompt, then Python gets confused (try it!). If you type:

```
'HAPPY' or "HAPPY"
```

then Python is happy and tells you so by repeating the string back to you:

```
'HAPPY'
```

(when given a choice, Python chooses single quotations).

Spaces are characters, and if your string contains, say, an apostrophe, you should choose the double quotations, as will Python when it throws the string back at you:

```
"I don't understand"
```

is a string of 18 characters, two of which are spaces and one is the apostrophe.

Python can do some arithmetic and logic on strings.

- `'x' + 'y'` returns the **concatenated** string `'x (followed by) y'`
- `2*'x'` returns the string `'xx'` and `3*"x"` returns `'xxx'`, etc.
- `len('x')` returns the (integer) length of the string `'x'`. Thus,

```
len("I don't understand") returns 18
```

(Notice that the `len('x' + 'y') = len('x') + len('y')`).

- Individual characters in strings can be referenced:

```
'x'[n]
```

is the  $n + 1$ st character of the string `'x'` (viewed by Python as a string). Thus:

```
'Happy Birthday'[0] is 'H' and 'Happy Birthday'[5] is 'y'
```

- Substrings can be extracted from strings via

```
'x'[m : n]
```

with  $m$  as the starting point and  $n$  the ending point of the substring. Thus:

```
'Happy Birthday'[1 : 4] is 'app'
```

- Strings can be compared with `==` and `!=` operations.

```
'2' == '2' returns False (why?)
```

- Strings can be printed. `print('HAPPY')` returns HAPPY (with no quotes).
- Integers and real numbers can also be printed.

```
print(.3) returns 0.3
```

**Exercise.** Play with concatenating, multiplying, substrings and printing strings. If `'x'` is a string, then `'x'[-n]` has a meaning. What is it? Same for `'x'[-m : -n]`.

**Reference.** <https://www.w3schools.com/python/> (Python strings)