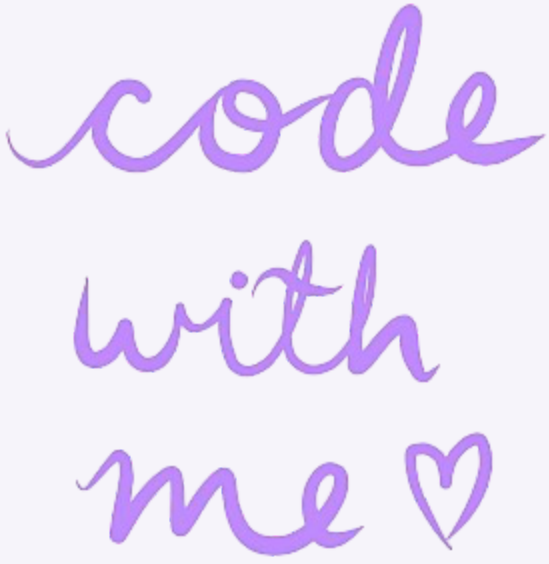
Code with Me Meeting #3



Strings

WHAT IS A STRING:

- Remember `print("Hello World!")` ?
  - "Hello World" is what you call a string
  - A string is a sequence of characters
  - In English, characters are the alphabet, numbers, ?, $, etc.
  - Your computer stores all of these in binary, ie, 0s and 1s.
- To tell your computer that it is a string, you enclose it in ' or ""
  - Any characters inside quotes are a string
- Triple quotes can be used to write multiple lines within a string
- `mystring = 'Hello'`

```
mystring = "Hello"
mystring = ' ' ' Hello ' ' '
mystring = " " " Hello World,
                I am learning Python " " "
```

- What will be the output for this?
- You can also specify a new line in a string within just one set of quotes: `mystring = 'Hello World, \n, I am learning Python'`

ACCESS CHARACTERS IN A STRING:

- We can locate the 1st, 2nd, 3rd, etc character in a string using square brackets.
- Ex: `mystring[0]` gives the first character of the string.
- When coding, 0 usually refers to the first character

SLICING A STRING:

- We can extract sections of characters from a string.
- This is called slicing.
- `mystring[1:5]` returns the 2nd to 6th characters in a string.
- Try with an example. Use the string "`I am learning how to code in Python`"
- Try slicing any number of characters, and see if the output is what you expected. Tell me how to output the following:

1. "learning how to code in Python"
2. "I am"
3. "I am learning how to code in Python"

CONCATENATING STRINGS:

- Concatenating strings is basically adding two or more strings together to make one big string.
- Just use + between the two, ex: `print('Hello' + ' World')` gives `Hello World`

NOTES:

- Many other things we can do to strings, we will cover those throughout the course.

NUMBERS:

- In Python, numbers are classified as either `int`, `float`, or `complex`
- `int` is an integer. This means it can be any number without a decimal point. Eg: -71234, 0, 800, 5,-2, etc.
- `float` is a floating-point number, meaning it can be any real number.
  - It is always represented with a decimal point.
  - Ex: -0.005, 1.0, 795.25, etc.
- `complex` is a mixture of numbers and letters, like `4 + 2j`
  - We won't be using them much because they're too *complex*

CONVERTING DATA TYPES:

- Everything we have seen just now, `string`, `int`, `float`, and `complex` are called datatypes
- In Python, we can convert data from one datatype to another.
- We use the built-in functions `str()`, `int()`, `float()`, and `complex()`
- Examples:
  - `float(7)` will return `7.0`
  - `int(2.4)` will return `2` (Rounds out the decimal)
  - `str(555)` will convert the integer `555` to a string
  - `int('555')` will convert the string `'555'` to an integer
  - Same way for floats. Try converting a string without numbers to an `int` or a `float`. See what happens.

CHECKING THE DATATYPE:

- Lastly, you can check the classification of any data using the type() function.
- Ex:
```
a = 5
print(type(a))
```
- Can you guess what this returns? Try it out!