09-python-files and data

Files

For this section, we'll learn how to open and read text files.
When we open up files, we need to consider the location or **path** to the file.
The path to the file is where the **text file** is with respect to the **python file**. This includes information about the file's **name** and the **folder** that the file lives in.

Opening Files

To open a file, we use the `open()` function.
The `open()` function returnns a **file stream**.
Once open, the **file stream** is like a pipe that we can get information from.

```
# information.txt is a file we want to connect to (same folder)
file = open("information.txt")
```

Reading their contents

When reading a file stream, you receive the first part of the file first. Then the second, ..., all the way to the end.
To read a part of the file stream, we use the `readline()` method. It gives one line of informatio. If you call it again, you'll get the next line.

```
# omitted code above
# read a line of text from file
line = file.readline() # returns string
second_line = file.readline() # next line
```

If you want to read all lines, you can iterate over the file stream.

```
# omitted code above
# read every line until the end
for line in file:
    # do something with that line's data
    print(line)
```

Managing File Streams

When we open a file stream, we should always close it when we finish. This helps to lower the risk of corrupting any data in the file.
To close a file stream, use the `close()` method

```
# omitted code above
file.close() #closes stream safety
```

Use the `with` epression to deal with closures

```
with open("information.txt") as file:
    line = file.readline()
    for line in file:
        print(line)
file.readline() #this will break
```

Lists

Lists are a type of data that are helpful in strong more than one piece
of information that is related. With lists, order matters. To create a list that we `use{}` called bracets.

```
some_list=[]
```

Converting a string to a list

There are times where we want to get information from a string.One use case where this is applicable is the example of a `.csv` or comma-separated vaules file.

```
Name,Age,Favourite Superhero
Mr.ubail,67,Batman
Bruce Wayne,32,Superman
You here,32,Planter Fascilities Man
# omitted code above
info = "Mr.ubail,67,Batman"
#Spilt the string wherever there's a,
info_list = info.spilt(",")
                           #["Mr.ubial","67","Batman"]
```

Getting a Specific Item from a List

To access a specific element inside the list,we use `[]` bracket notation. To get a certain thing from inside the list, we need to "know" the address,
which we call the **index Indices** are always integers.

~~~
# omitted code above
### Getting a Specific Item from a List
To access a specific element inside the list, we use the `[]` bracket
notation. To get a certain thing from inside the list, we need to "know" the
address, which we call the **index**.
**Indices** are always integers.
```python
# omitted code above
~~~

```python
# index     ->      0         2        3
# info_list -> ["Mr. Ubial", "67", "Batman"]
# access age?
print(f"Mr. Ubial's age is {info_list[1]}.")
# how do we access fave superhero? ----> -1
print(f"Mr. Ubs' fave hero is {info_list[-1]}.")
```