Chapter 4: Basic Line and Scatter Plots

4-1:

- `plot` function creates line plot and `scatter` function creates scatter plots.
- to import a function only we can use:
  `from modulename.submodule import function`
  or, to import all functions in that module:
  `import module.submodule as alias`

4-2:

```
import matplotlib.pyplot as plt
#a simple line plot
plt.plot(xvalues, yvalues)
```

4-3:

We can add the following lines:

```
plt.xlabel("x-axis label")
plt.ylabel("y-axis label") \
```

4-4:

Function variables are positional, so both call will work but will produce different result. Because (x, y, z) will assign x to the first variable in the finction and so on. But in list [z, y, x], z will be 1st and so on. resulting in different solution.

4-5

A list is list of objects or items that can be edited such as replaced, added or removed from the original list. It uses square brackets - `[]`.
A tuple is almost same as a list but it is not editable, but items in a tuple can be used into a new variable. It uses round brackets - `()`

4-6:

1. `len(data)` for length
2. `data[0]`
3. `data[2]`
4. `data[-1]`
5. `data[3, 10]`

species_name = """Homo sapiens"""

4-7

All of them are used for strings. However, triple quotes enable multiple lines of string.

4-8

```
all_labels = label1 + ',' + label2 + ',' + label3
```

4-9

```
all_labels_short = "Cultures " + label1['-1'] + label2['-1'] + label3['-1']
```

4-10

```
id_label = "Sample" + str(sample_id)
```

4-11

If it interpretes `#` in the beginning of a line.

4-12

Using `*` at the beginning of each line.

```
* 3Erlenmeyerflasks.
* testtubes.
* 1Bunsenburner.
```

4-13

By using a backslash `\`.

e.g.:

```
q413 = "How does Python know" + \
       "when it has encountered" + \
       "a comment line?"
```