if/else

```
if test expression:
    Something happens
elif test expression:
    Something else happens
else:
    Another something else happens
```

Only one of the `something` will be triggered. If conditons overlap, first one will overtake.

> Now,copy the content of `ex_if.py` in one of your file and execute it using python3.

for loop

```
for val in sequence:
        Something happens
```

> Now, go to the `examples` folder, copy the content of `ex_for.py` in one of your file and execute it using python3.

Exercises

1. Fibonacci

Write a script that asks the user how many Fibonacci numbers to generate, then generates them and prints them as a list. Fibonacci sequence: $x_n = x_{n-1} + x_{n-2}$

2. Divisors

https://www.practicepython.org/exercise/2014/02/26/04-divisors.html

3. Common elements in lists

```
l1 = ["42", "11", "33", "97", "63", "86", "4", "46", "72", "88", "59", "55", "13'
l2 = ["24", "98", "56", "59", "3", "42", "14", "37", "75", "5", "34", "63", "4" ]
```

Write a script that prints all elements common to both list, then prints all elements common to both list and divisible by 7 (see `ex_listcommon.py` for solution).

import

`import <module_name>`

Let's go ahead and open python3 (`Ctrl+Alt+T` input `python3` then `Enter`):

```
>>> import time
```

We just loaded everything included in the `time` package. Ond of the function is `sleep` .

```
>>> time.sleep(7)
```

Loading a package takes resources (memory, time). Sometimes we just want one function:

```
>>> from time import sleep
>>> sleep(2)
```

- Basic packages are provided with python (https://docs.python.org/3/library/), anyone can write a package that we can the download online.

Caesar's cipher: string package (1/2)

Write a caesar cipher program to encode and decode messages.

Below are some helpful functions.

Need the alphabet as a string ?

```
>>> import string
>>> string.ascii_lowercase
```

Position of letter `t` in the alphabet (start counting from 0) ?

```
>>> import string
>>> alphabet = string.ascii_lowercase
>>> alphabet.find("t")
```

https://docs.python.org/3/library/string.html

Caesar's cipher: string package (2/2)

Add character to a new string using character's position in the alphabet:

```
>>> import string
>>> alphabet = string.ascii_lowercase
>>> new_character = alphabet[8]
>>> encrypted_message = ""
>>> encrypted_message += new_character
>>> new_character = alphabet[22]
>>> encrypted_message += new_character
>>> encrypted_message
```

You should now be able to write a nice Caesar's cipher encoder/decoder (solution available: `ex_caesarcipher.py`).