Object \ \ Date

[Functions]

pros:- chuncks of code that allows you encapsulate the task.

encapsulation :- انك تنفذ التاسك في أمر واحد بدل أكثر من steps

functions organise and optimize code.

localvariable:- it can only used within the body of this function

مينفعش استخدمه خارج اطار ال body بتاع function.

* Can't access outside the body of function.

without no gaps ← function_name

function header: def function_name (,) :

constant colon

arguments or input Parameters

constant at any function — body of function

return → output

* The rule of function name includes variable name

Variable scope:- which variables you have access to in python

Some functions don't return a value → is none

Object   \ \ Date

*لو انا حاطط لل parameter ⟸ default Value
ومحددتش value اثناء call هيحسبله Value بناءً على
default Value اللي انا حاطيتها من البداية

we can Call function in two ways 1- by position
cylinder_Volume (10, 7)
2- by name
cylinder_volume(height= 10, radius = 7)

scope of function which parts of program a variable can be refrenced.
*variable inside function you can use inside this function

*variable outside " you can use outside any function and any place in the program.

*local variable we can use the same name in many different functions.

*global variable the value cannot be modified.
*if you want to modify it should be passed in an argument

Documentation :- makes the Code easier to read and more usable

documentation strings or docstrings ⇒

DocStrings :- A Type of comment used to explain the purpose of a function AND How it should be used

*بتساعدك ازاى تستخدم ال function والغرض الاساسي منه عاملها ايه

* Docstrings surrounded by triple quotes.
First line is the breif explanation of the function's purpose.

*ممكن اكتفى بـ first line بس لو مفهمتش هبتدى اشرح
ال arguments بتاعتى
وممكن اشرح ال output

* everything in docstring is optional

* docstring is the part of cleaning and perfect Code

* They assist you and future users in understanding the Code you produced.

Object \ \ Date

lambda expression:- used to create Anonymous function (function ~~isn't~~ doesn't have name)

higer order function:- it contains other function as aparameter or return function as an output

The function that operates with another function are known as [higher order function]

properties of high order function:-

- A function is an instance of the object type
- You can store function in avariable
- You can pass function as aparameter
- You can store them in Data structure

lambda argument: expression

double = [lambda x : x * 2]

→ name of

lambda x, y : x * y

* only have one expression which is evaluated and returned

* بنستخدمها في حاجه مش هنحتاجها الا مرة وبعد كدا

"file objects"

* open Command isn't a best way !
* Context manager is the best and useful way.

open("test.txt") *عشان افتح أى file واتعامل معاه
file-name ↗

عشان افتح نفس directory
لو مش فى نفس الـ directory لازم تكتب path كامل.

for reading a file 'r'
for writing a file 'w'
for appending a file 'a'
for read and write file 'r+'

close()

* model name * in this way
* we must close the file.
* لو نسيتها مقفلها هيعمل error.
* لازم

Context manager

* close the file automatically and avoid any error or exc
with open("filename.txt") as name of variable:
← اقدر access عليه عادى
* لو عملنا close وجيت عملت read() مش read()
← مقدرش اعمل access تانى على file.

الأمل

pass → بتمنع أي error بيحصل لو انا مش عارف هعمل ايه

بنستخدمها في if / loops / fun

* when the user doesn't want any code to execute ⇐

readlines() → list بيطبع المحتوى على شكل

ولكن بيبقى فيه مشكلة في \n

extra new line

readline() → content بيقرأ سطر واحد منه

لو كررتها أكثر من مرة ورا بعض

هيطبع السطر اللي بعده by default بيعمل

extra newline

end='' عشان الحل بحط

→ in print()

```
for line in variable_name:
    print(line, end='')
```

read(num) → number of character that I want to print it.

mpty string هيطبع content of file لو عدد الـ characters أكتر من

seek() X ⇐ Can't understand

Object

Date

writing files → لو مش موجود create واحد جديد تلقائي

```
f = open('name.txt', 'w')
f.write('...')
```

لو

[error and exception]

```
try:
    pass
except:
    pass
```

← الكود اللي لو حصل هيحدث error

عشان اتجنب الـ error دا

بعمل handling

except لو الكود دا اتنفذ صح يروح ينفذ الكود الموجود من

```
else:
```

← لو محصلش الـ error الموجود من try

عشان يروح لل except يروح على else

```
Finally
```