

١٧ ربیع الآخر
باب

الاحد
SUNDAY

أكتوبر
20 OCTOBER

* Collections: - single "variable" used to store multiple values

dictionary → مخزن في الآخر يبعد باقي الأنواع

List = [] ordered and changeable. Duplicates OK.

Set = {} unordered and immutable, but

Add/Remove OK. No duplicates

Tuple = () ordered and unchangeable. Duplicates OK.
and Faster

list:-

Fruits = ["apple", "orange", "banana", "coconut"]

print(Fruits) → ['apple', 'orange', 'banana', 'coconut']

print(Fruits[0]) → apple [1] → orange [3] → coconut
[4] → error: out of range

[: 3]

print(Fruits[0:3]) → ['apple', 'orange', 'banana']

print(Fruits[::2]) → ['apple', 'banana']

[::-1] → ['coconut', 'banana', 'orange', 'apple']

١٦ ربیع الآخر
باب

السبت
SATURDAY

١٩ أكتوبر
19 OCTOBER

For fruit in range fruits: → apple
print(fruit)

orange

banana

coconut

print(dir(Fruits)) → prints attributes that be used

print(help(Fruits)) → prints a description of each of the methods printed from dir()

print(len(Fruits)) → 4

print("apple" in Fruits) → True

→ checks if the element exists in the list
→ the output: (True or False)

Fruits[0] = "pineapple" → apple

apple →

For fruit in Fruits: → pineapple
print(fruit)

orange

banana

coconut

١٥ ربیع الآخر
باب

الجمعة
FRIDAY

١٨ أكتوبر
18 OCTOBER

Fruits.append("pineapple") → إضافة عنصر لآخر
print(Fruits) → ['apple', ..., 'pineapple'] List

Fruits.remove("apple") → احذف عنصر من الـList
print(Fruits) → ['orange', 'banana', 'coconut']

Fruits.insert(1, "pineapple") → إضافة عنصر في مكان محدد
print(Fruits) → ['apple', 'pineapple', 'orange', ...]

Fruits.sort() → يرتيب الكلمات حسب أول حرف أبجدياً
print(Fruits)
→ ['apple', 'banana', 'coconut', 'orange'] أو الأرقام بالترتيب

Fruits.reverse() → يعكس ترتيب العناصر حسب ترتيب
print(Fruits)
إدخالها ويسort ترتيبها أبجدياً
→ ['coconut', 'banana', 'orange', 'apple']

Fruits.sort()
Fruits.reverse() → تأكيد ترتيب العناصر أبجدياً
print(Fruits) → ['orange', 'coconut', 'banana', 'apple']

Fruit.clear() → احذف جميع عناصر الـList
print(Fruit) → []

١٤ ربیع الآخر
باب

الخميس
THURSDAY

١٦ أكتوبر
17 OCTOBER

Fruits.index("apple") → تحديد مكان عنصر في الـList
print(Fruits.index("apple"))
print(Fruits.index("pineapple")) → error

print(Fruits.count("banana")) → ترتيب عناصر في الـList
print(Fruits.count("pineapple")) → 0

Sets:- {} unordered and immutable, but Add/Remove
No duplicates OK.

Fruits = {"apple", "orange", "banana", "coconut"}

print(Fruits) → {'orange', 'apple', 'banana', 'coconut'}
يجلب جميع العناصر لكن ترتيب عشوائي

print(dir(Fruits)) → يجلب الميثود بالـSets وهي
انه متشابه مع lists ولكنها لا ترتيب
print(help(Fruits)) → يشرح الميثود

ومن الميثود المتبقي Len ← تحديد عدد العناصر
count ← ترتيب العناصر كد من وجود العنصر في الـSet
len ←

لكن لا يوجد الميثود set ← نجد الميثود index ←
print(Fruits[0]) → Error

١٢ ربیع الآخر
٦ بابه

الاربعاء
WEDNESDAY

١٦ أكتوبر
16 OCTOBER

Fruits.append("pineapple")
يمكنك إضافة أودن في عنصر
Fruits.remove("apple")
ـ

يمكنك إزالة أول عنصر لكن في الذئب
print(Fruits)
يعني قيم شواعي
→ ['apple', 'banana', 'coconut']
→ ['orange', 'coconut', 'apple']

Fruits.clear()
ـ لحذف جميع العناصر
print(Fruits) → set()

fruits = ["apple", "apple", "coconut"]
ـ يحفظ
print(Fruits)
ـ النسخ المكررة من العناصر
→ {"coconut", "apple"}

ـ الـ Sets تعمل بشكل أفضل مع الشوايات مثل الألوان

Tuples :- () ordered and unchangeable.
Duplicates OK. Faster

Fruits = ("apple", "orange", "banana", "coconut", "coconut")
ـ يطبع كل محتويات الـ tuple
print(Fruits) → ('apple', 'orange', 'banana', 'coconut',
'coconut')

١٢ ربیع الآخر
٥ بابه

الثلاثاء
TUESDAY

١٥ أكتوبر
15 OCTOBER

ـ لا تحتوى الـ Tuples على
ـ كثيرة وهم موجودين في
ـ print(dir(Fruits)) → print(help(Fruits))

ـ مثل ما شاء سأقا يوجد
ـ methods مثل `len` ← `len` methods
ـ ولدينا 2 methods
ـ print(Fruits.index("apple")) → 0

ـ print(Fruits.count("coconut")) → 2

2D collections:-

fruits = ["apple", "orange", "banana", "coconut"]
vegetables = ["celery", "carrots", "potatoes"]
meats = ["chicken", "fish", "turkey"]

groceries = [fruits, vegetables, meats]

ـ يطبع كل محتويات الـ lists
ـ كل الـ lists
ـ print(groceries) → [[apple, orange, banana, coconut], [celery,
carrots], [chicken, fish, turkey]]

١١ ربیع الآخر	الاثنین MONDAY	١٤ أکتوبر 14 OCTOBER
---------------	-------------------	-------------------------

باب :

يطبع محتويات الـ List المختلقة \rightarrow
`print(groceries[2])`
 $\rightarrow ['chicken', 'fish', 'turkey']$

ترقيق النص في الأداة \downarrow
`List`
`print(groceries[0][1])`
 $\rightarrow ('apple')$
 $\rightarrow orange$

يمكن كتابة العناصر في collection 2D مباشرة بدل اسم `list`
`groceries`

```
groceries = [
    ["apple", "orange", "banana", "coconut"],
    ["celery", "carrots", "potatoes"],
    ["chicken", "fish", "turkey"]]
```

`print(groceries[0][0])` \rightarrow apple

For collection in groceries:

```
print(collection)
 $\rightarrow ['apple', 'orange', - - -]
['celery', 'carrots', - - -]
['chicken', - - -]$ 
```

١٠ ربیع الآخر	الاحد SUNDAY	١٢ أکتوبر 13 OCTOBER
---------------	-----------------	-------------------------

باب :

For collection in groceries: \rightarrow apple
 For Food in collection: orange
`print(Food)`

`(print(Food, end=" "))`

`print()`

```
 $\rightarrow$  apple, coconut
celery, potatoes
chicken, turkey
```

Dictionary: a collection of {key : value} pairs
 ordered and changeable. No duplicates

```
capitals = {
    "USA": "Washington D.C",
    "Egypt": "Cairo",
    "China": "Beijing",
    "Russia": "Moscow"}
```

`print(dir(capitals))` \rightarrow list methods
`print(help(capitals))` \rightarrow methods

٩ ربـيع الآخر
باب

السبـت
SATURDAY

١٢ أكتـوبر
12 OCTOBER

print(capitals.get("Egypt"))
→ Cairo
print(capitals.get("Japan")) → None

استخدام if
if capitals.get("Japan"):
print("That capital exists")
else:
print("That capital doesn't exist")
→ That capital doesn't exist

capitals.update({"Germany": "Berlin"})
يستخدم لـ خلاف قيمة key
print(capitals) → {'USA': 'Washington DC',
'Russia': 'Moscow', 'Germany': 'Berlin'}

capitals.update({'USA': 'Detroit'})
print(capitals) → {'USA': 'Detroit', ...}

capitals.pop("USA") →
→ {'Egypt': 'Cairo', 'China': 'Beijing', 'Russia': 'Moscow'}

capitals.popitem() → dictionary[] في Key آخر
وهي هنا لا تكون Moscow

٨ ربـيع الآخر
باب ١٩٤١

الجمـعة
FRIDAY

١١ أكتـوبر
11 OCTOBER

capitals.clear() → dictionary[]
print(capitals) → {}

keys = capitals.keys() →
print(Keys) → dict_keys(['USA', 'Egypt', 'China', 'Russia'])

For Key in capitals.keys():
print(key, end=" ")
→ USA Egypt China Russia

values = capitals.values() →
print(values) → dict_values(['Washington D.C.', 'Cairo', ...])
ويتم تحليل values

items = capitals.items() → 2D Tuples
→ items = [(1, 1), (1, 1)]

print(items) → dict_items([('USA', 'Washington D.C.'),
('Egypt', 'Cairo'), ('China', 'Beijing'), ('Russia', 'Moscow')])

for Key, value in capitals.items():
print(f"[Key]: {value}")
→ USA: Washington D.C.
Egypt: Cairo
China: Beijing
Russia: Moscow

٧ ربیع الآخر
٢٩ توت

الخميس
THURSDAY

١٠ أكتوبر
10 OCTOBER

* Generating random numbers :-

تتيح لنا استخدام المدير من الـ `random` \rightarrow `method`
 `print(help(random))` \rightarrow تطبع الـ `method` استخدموها

`number = random.randint(1,6)` \rightarrow أجزاء ١، ٢، ٣، ٤، ٥، ٦ عشوائي من
`print(number)` \rightarrow ٤ \rightarrow ١ \rightarrow ٦ جويعاً أي عدد يحدد بدرايتها ونهايتها

Low = 1

High = 100

`number = random.randint(Low, High)`

`print(number)` \rightarrow ٥٥ \rightarrow ٦٥ \rightarrow ١ \rightarrow ١٠٠

أجزاء ١، ٢، ٣، ٤، ٥، ٦ عشوائي (Float point) \rightarrow `number = random.random()` \rightarrow `print(number)` \rightarrow ٠.٣٢ --- ٠.٦٢

`options = ("rock", "paper", "scissors")` \rightarrow أجزاء ١، ٢، ٣ عشوائية \rightarrow `option = random.choice(options)` \rightarrow `print(option)` \rightarrow paper

٦ ربیع الآخر
٢٩ توت

الإربعاء
WEDNESDAY

٩ أكتوبر
9 OCTOBER

`cards = ["2", "3", "4", "5", "6", "7", "8", "9", "10", "J", "Q", "K", "A"]`
`random.shuffle(cards)` \rightarrow تغيير الترتيب عشوائياً \rightarrow `print(cards)`
 \rightarrow `['K', '5', '7', '9', 'A', 'J', 'Q', '2', '4', '3', '6', '8', '10']`

* Encryption :-

`import random`
`import string`

`chars = string.punctuation + string.digits +`
`" " + string.ascii_letters`
 \rightarrow تكتب كل حرف من حروف الأختبار \rightarrow
`print(chars)` \rightarrow ! \$ # % ^ & @ 1 2 3 4 5 6 7 8 9 a b c d e f g h i j k l m n o p q r s t u v y z A B C D E F G H I J K L M N O P Q R S T U V Y Z

لتحويله إلى `List` وفصل الناتم بـ `,` \rightarrow
`print(chars)` \rightarrow [! , # , % , ^ , & , @ , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , a , b , c , d , e , f , g , h , i , j , k , l , m , n , o , p , q , r , s , t , u , v , y , z , A , B , C , D , E , F , G , H , I , J , K , L , M , N , O , P , Q , R , S , T , U , V , Y , Z]

لتحفيظ `chars` في `key` \rightarrow `key = chars.copy()` \rightarrow Keys \rightarrow `random.shuffle(key)` \rightarrow تغيير ترتيب المثلث عشوائياً \rightarrow

`print(key)` \rightarrow [0 , \$, W , 9 , < , ~ , ' , 3 ,

٥ ربیع الآخر
٢٨ سوت

الثلاثاء
TUESDAY

٨ أكتوبر
8 OCTOBER

*Function:- A block of reusable code
place () after the function name to invoke it

```
def happy_birthday(): → function
    print("happy birthday to you!") ← وين ايه اسوبورن
    print("You are old!")
happy_birthday() → نكتب اسم Function لتنفيذها
                    → happy birthday to you!
                    You are old!
```

(1) إلرسال بيانات ميزة لا تكتب بيانات داخل الـ Function
في المعاشر ونكتب البيانات في سطر تنفيذها ويجب مطابقة عدد التفاصير المردخلة مع عدد الـ variables ويلزم الترتيب
def happy_birthday(name, age):
 print(f" happy birthday to {name}!")
 print(f" You are {age}!")
happy_birthday("Essam", 20)
→ happy birthday to Essam!
You are 20

٤ ربیع الآخر
٢٧ سوت

الاثنين
MONDAY

٧ أكتوبر
7 OCTOBER

return = statement used to end a function
and send a result back to the caller

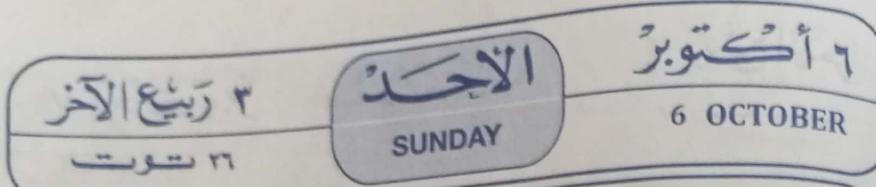
```
def add(x, y)
    z = x + y
    return z
```

تحويل التعبير الى Function
 ↗ string ↗ return ↗
 ↗ return f"title-{first}-{last}"

*default arguments = A default value for certain parameters
default is used when that argument is omitted
make your Functions more Flexible, reduces arguments
1. positional, 2. DEFAULT, 3. Keyword, 4. arbitrary

```
def net_price(list_price, discount=0, tax=0.05):
    return list_price * (1 - discount) * (1 + tax)
print(net_price(500)) → 525.0
print(net_price(500, 0.1)) → 472.5
print(net_price(500, 0.1, 0)) → 450
```

```
def net_price(list_price=0, discount, tax)
→ Error: 20:095 default arguments must  
be positioned arguments!
```



Key word arguments = argument preceded by an identifier
 → order of arguments doesn't matter
 → helps with readability
 1. positional, 2. default, 3. KEYWORD, 4. arbitrary

```
def hello(greeting, title, name):
    print(f'{greeting}{title}{name}')
```

```
hello("Hello", name="Essam", title="Eng.")
```

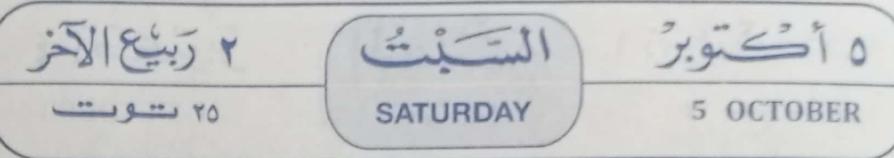
Key word arguments go positional arguments
 يُؤثِّرُ ترتيبَ الأرجمنتس على ترتيبِ
 فيجبُ ترتيبَ الأرجمنتس وفقاً لـ "args"
 بدون الحاجةِ لِلإلتزامِ بِترتيبِ
 → Hello Eng. Essam

sep , end جو ميزةٌ في keyword arguments يوجد

For x in range(1, 11)

```
print(x, end=" ") → 1 2 3 4 5 6 7 8 9 10
```

```
print("1", "2", "3", sep="-") → 1-2-3
```



Arbitrary arguments = varying amount of arguments

Parameters (أرجمنتس) :-

- *args = allows you to pass multiple non-key arguments
- **kwargs = allows you to pass multiple keyword arguments
- * = unpacking operator

1. positional, 2. default, 3. Keyword, 4. ARBITRARY

def add(*args):

print(type(args)) → <class 'tuple'>
 tuples (أرجمنتس) لـ "args" ←
 For loop (iterate over it) لـ "args" ←
 أو التكرار لـ "args"

def add(*args):

total = 0
 for arg in args:
 total += arg
 return total

يمكن استبدال "args" ←
 بـ "nums" ←
 def add(*nums):

```
print(add(1, 2, 3)) → 6
```

```
print(add(1, 4, 7, 9)) → 21
```

```
print(type(kwargs)) → <class 'dict'>
```

Kwargs (أرجمنتس) ←
 dictionary (دكتريوني) ←
 استخدامه لـ "kwargs" ←

١ ربیع الآخر ١٤٤٦

٢٢ توت

الجمعة

FRIDAY

٤ أكتوبر

4 OCTOBER

```
def print_address(**Kwargs):
    for key,value in Kwargs.items():
        print(f'{key}:{value}')
print_address(street="21 Jump St.",village="Konoha",
            planet="Vegeta")
```

→ street : 21 jump St.

village : Konoha

planet : Vegeta

يمكن استخدام **Kwargs و args
يلزمه كتابة كذا قبل args

```
def shipping_Label(*args,**Kwargs):
```

for arg in args:

print(arg,end=" ")

print() بدل من print(" ")

print(f'{Kwargs.get('street')}').

print(f'{Kwargs.get('village')},
{Kwargs.get('planet')}')

```
> shipping_Label("Eng.", "Essam", "Shenhab",  
street="21 jump St.", village="Konoha", planet="Vegeta")
```

→ Eng.Essam Shenhab

21 jump street

Konoha , Vegeta

٣٠ ربیع الأول

٢٢ توت

الخميس

THURSDAY

٤ أكتوبر

3 OCTOBER

*module = → file containing code you want to include in your program
→ use 'import' keyword to include a module (built-in or your own)
→ useful to break up a large program into smaller separate files

print(help("modules")) → يمكن استدعاء modules time , sys , math ...
print(help("module")) → print(help("math"))

import math → math يمكن من المودules لـ

print(math.pi) → Functions like variables or class
→ 3.141 ...

import math as m → module like sin() ...
print(m.pi) → m like pi ...

from math import e → يمكن اسماً variable like sin() ...
from ... import ...

a,b = 1,2

print(e**a) → 2.71

print(e**b) → 7.389

print(e**c) → 27

٢٩ ربیع الأول
٢٢ توت

الاربعاء
WEDNESDAY

أكتوبر
2 OCTOBER

```
import math
a, b, c = 1, 2, 3
print(math.e**a) → 2.71...
print(math.e**b) → 7.389...
print(math.e**c) → 20.085...
```

لما يقصد الكاتب هنا ←
في project Folder. في his/her module
example/main.py وليكون فيه ملفين: new python file
example.py ما تشاء في

```
pi = 3.14
def square(x):
    return x**2
main.py استدعاء pi
import example
result = example.pi
print(result) → 3.14
```

```
result = square(3)
print(result) → 9
```

٢٨ ربیع الأول
٢١ توت

الثلاثاء
TUESDAY

١ أكتوبر ٢٠٢٤
1 OCTOBER 2024

*variable scope = where variable is visible and accessible
scope resolution = (LEGB) Local → Enclosed → Global → Built-in

Local:-

def Func1():

a = 1
print(a)

def Func2():

b = 2
print(b)

Func1()

Func2()

→ 1
2

def Func2():

a = 1
print(b)

def Func1():

b = 2
print(a)

Func1()

Func2()

→ Error: a and b not defined

ولكن ما يجري في Func1() b greater than a
في Func2() a greater than b وبالتالي

def Func1():

x = 1 → Local
print(x)

def Func2():

x = 2 → Local
print(x)

Func1()

Func2()

→ 1
2

الآن هناك نوعين مختلفين من variable
من نفس الاسم

مذكرة

NOTES

Enclosed:-

نستطيع كتابة Function داخل Function

```
def Func1():
```

```
    x=1 → enclosed
```

```
    def Func2():
```

```
        print(x)
```

```
    Func2()
```

```
Func1() → 1
```

Global:- Functions خارج الـ Variables

```
def func1():
```

```
    print(x)
```

```
def Func2():
```

```
    print(x)
```

```
x=3 → global
```

```
Func1()
```

```
Func2()
```

```
→ 3
```

```
3
```

```
def Func1()
```

```
    x=1 → Local
```

```
    print(x)
```

```
def Func2()
```

```
    x=2 → Local
```

```
    print(x)
```

```
x=3 → global
```

```
Func1()
```

```
Func2()
```

```
→ 1
```

```
2
```

٢٧ ربیع الأول

٢٠ توت

الاثنين

MONDAY

٢٠ سبتمبر

30 SEPTEMBER

Built-in:-

```
from math import e → built-in
```

```
print(e)
```

```
→ 2.71 ...
```

```
From math import e → built-in
```

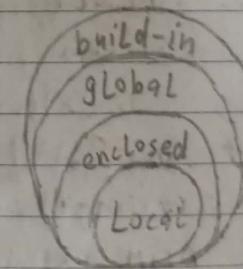
```
def Func1()
```

```
    print(e)
```

```
e=3 → global
```

```
Func1
```

```
→ 3
```



٢٦ ربیع الأول
١٩ سوتوت

الاحد
SUNDAY

٢٩ سبتمبر
29 SEPTEMBER

*exception = events detected during execution
that interrupt the flow of a program
حدث عمل تنفيذ البرنامج مثل القسم على حرف أو كلمة

try:

يبدأ بـ try:

```
numerator = int(input())
denominator = int(input())
result = numerator / denominator
```

يُطلب من المقتبس على حرف →
except ZeroDivisionError as e:
print(e) → يتم تسمية الخطأ برمز لتسهيل كتابته

print("can't divide by zero")

يحدث عند تلقي قيمة خاطئة →
except ValueError as e:
print(e) → مثل حرف في input()

print("Enter only number")

يحدث عند تلقي خطأ عوم →
except Exception as e:
print(e) → ويكتب احتياماً إن لم يتم الأخذ بالاعتراض آخر
print("something went wrong")

else: → يُطلب تنفيذ ما لم يتم حدوث أي خطأ
print(result)

Finally: → تكتب دائمًا في الآخر وتنفذ كل لو حدث خطأ
print("This will always execute")

e. handling open files
خايدتها مساعدة في الكود أو لأننا نستخدمنا
open files، finally block، finally block وفتتفهم أكثر في

٢٥ ربیع الأول
١٨ سوتوت

الستینٹ
SATURDAY

٢٨ سبتمبر
28 SEPTEMBER

*file detection

بلام استدعاء os module
مكان تواجد الملف المطلوب موجود في
لكن تكتب "D:\\test.txt" ،
ذكر لافتة escape sequence

لتتأكد من وجود الملف →
if os.path.exists(path):
print("That location exists!")

لتتأكد إن كان الملف ملف → file
if os.path.isfile(path):
print("That is a file")

لتتأكد إن كان الملف مجلد → directory
if os.path.isdir(path):
print("That is a directory")

إذا كان الملف غير موجود → Folder
else:
print("That Location doesn't exist!")

إذا كان الملف موجود ← test.txt
That location exists!
That is a file

وإذا كان الملف مجلد ← Folder
That location exists!
That is a directory

وإذا لم يتم التوقيع ←
That location doesn't exist!

٢٩ سبتمبر
الاحد
SUNDAY

٢٩ سبتمبر
٢٩ سبتمبر
29 SEPTEMBER

* exception = events detected during execution
that interrupt the flow of a program
حدث عمل تنهي البرنامج مثل العصا على حشرة كلبة

try:
 numerator = int(input())

denominator = int(input())

result = numerator / denominator

يحدث عند القسم على صفر →
except ZeroDivisionError as e:
 print(e)
 print("Can't divide by zero")

يحدث عند تلقى قيمة خاطئة →
except ValueError as e:
 print(e)
 print("Enter only number")

يحدث عند تلقى خطاً عمومياً →
except Exception as e:
 print(e)
 print("Something went wrong")

else: →
 print(result)

Finally: →
 print("This will always execute")

خايدتها مسح المعلم في الكورن أو اننا مستخدمنا
file handling لزها ثها يار finally block open Files

٢٥ سبتمبر
السبت
SATURDAY

١٨ سبتمبر

٢٨ سبتمبر
السبت
SATURDAY

٢٨ سبتمبر

file detection

لهم استدعاء الدليل

مكان توافد الملف الذي أكتبون وجوده
أو لا ←
import os
path = "D:\\test.txt"
print(path)

إذا تذكر وجود الملف →
if os.path.exists(path):
 print("That location exists!")

لما ذكر ←
if os.path.isfile(path):
 print("That is a file!")

لما ذكر ←
elif os.path.isdir(path):
 print("That is a directory!")

إذا كان الملف غير موجود →
else:
 print("That Location doesn't exist!")

إذا كان test.txt ←
That Location exists!
That is a File

وإنما كان ←
That Location exists!
That is a directory

وإنما لم يتم العثور ←
That Location doesn't exist!

٢٤ ربیع الأول
١٧ توت

الجمعة
FRIDAY

٢٧ سبتمبر
27 SEPTEMBER

Reading a File:-

نفتح file ونسميه test ونكتب بداخله مثل:
→ Hey you!
You are awsome.

نبدأ بـ r

With open(^{Project}Folder, "r") as file:
أو ال Path

with open('text.txt') as file:
with open('D:\text.txt') as file:
print(file.read()) → file.read()

وبس كده لكن ممكن نزود لفاف الملف

ولكن يفضل استخدام except و try لعدم مقاطعة الكور

try:

with open('test.txt') as file:
print(file.read())

except FileNotFoundError:

print("That file was not found")

→ Hey you!

You are awsome.

إذا كان الملف موجود

→ That file was not Found
إذا لم يتم العثور على الملف

٢٣ ربیع الأول
١٦ توت

الخميس
THURSDAY

٢٦ سبتمبر
26 SEPTEMBER

Writing a File:-

نوعاً ما نكتب طرق قراءة الملف
لكتبة مازين إما فتحه أو كتابته
with open('text.txt', 'w') as file: → w → write
file.write(text)
→ r → read
→ a → append

ومن فتح text.txt الكتابة في الملف
كل مره يعمل overwrite و إذا أردنا إما فتح
كلام جديد بدون overwrite
نكتب 'w' بدلاً 'a'

Copying a File:-

copyFile() = copies contents of a file

copy() = copyFile() + permission mode + destination
can be a directory

copy2() = copy() + copies metadata (file's creation
and modification times)

import shutil
يستخدم الشتيل يستعمل الأدوات بدل
استخدام اخبارنا بين الـ ٣ أدوات بدل
copyFile() حالياً

shutil.copyfile('D:\text.txt', 'E:\Task2\copy.txt'),

سيقوم دليلاً جديداً باسم copy في المكان المختار

استخدام copy() و copy2() و copyfile() ولكن تختلف على
المشروع أو الغرض

٢٢ ربیع الأول
١٥ توت

الاربعاء
WEDNESDAY

٢٥ سبتمبر
25 SEPTEMBER

* Moving a File:-

Project
أي مكان على الجهاز
إن كان في ملف ادعه
↓
import os
source = "D:\\test.txt" أو "المكان الأصلي الملف"
destination = "E:\\test.txt" أو "المكان المراد نقله إليه"

يفضل كتابة الكود داخل except و try

try:

if os.path.exists(destination):
 print("There is already a file there")
 بنفس المكان

الاسم موجود لكن لا يحدث overwriting يمكن حذفها الجزء final
لم تتمكن يوم من نوع الملف

else:
 os.replace(source, destination)
 print(source + " was moved")
 إلى destination

except FileNotFoundError:
 print(source + " was not found")
إذا لم يتم العثور على الملف

Folders, Files
* إذاً الكود يستعمل نقل

٢١ ربیع الأول
١٤ توت

الثلاثاء
TUESDAY

٢٤ سبتمبر
24 SEPTEMBER

* Deleting a File:-

import os
import shutil
path = "FolderName" أو "Name of File"
try:
 os.remove(path) → delete a file
 os.rmdir(path) → delete an empty folder
 shutil.rmtree(path) → delete a directory containing files (but dangerous)

except FileNotFoundError:
 print("That file was not found")
إذا لم يتم العثور على الملف

except PermissionError:
 print("You do not have the permission")
إذا طلعت خطأ لا تملك

except OSError:
 print("You cannot delete using that function")
إذا حاول حذف فolder أو file

else:
 print(path + " was deleted")
إذا تم إثمار على الملف أو folder وتم حذفه

*