Faculty of Engineering

Mansoura national university

Artificial Intelligence Engineering Prog.

Machine Learning (CSE 2524)

Lab (1)

python revision (1)

By / Omar Mahmoud Wahba

TA at CSED – FOE - MU

Lab(1)

Lab - Outlines

- Course overview.
- requirements.
- print function.
- variables in python.
- input function
- if statements.

Course Overview

what’s Machine Learning?.

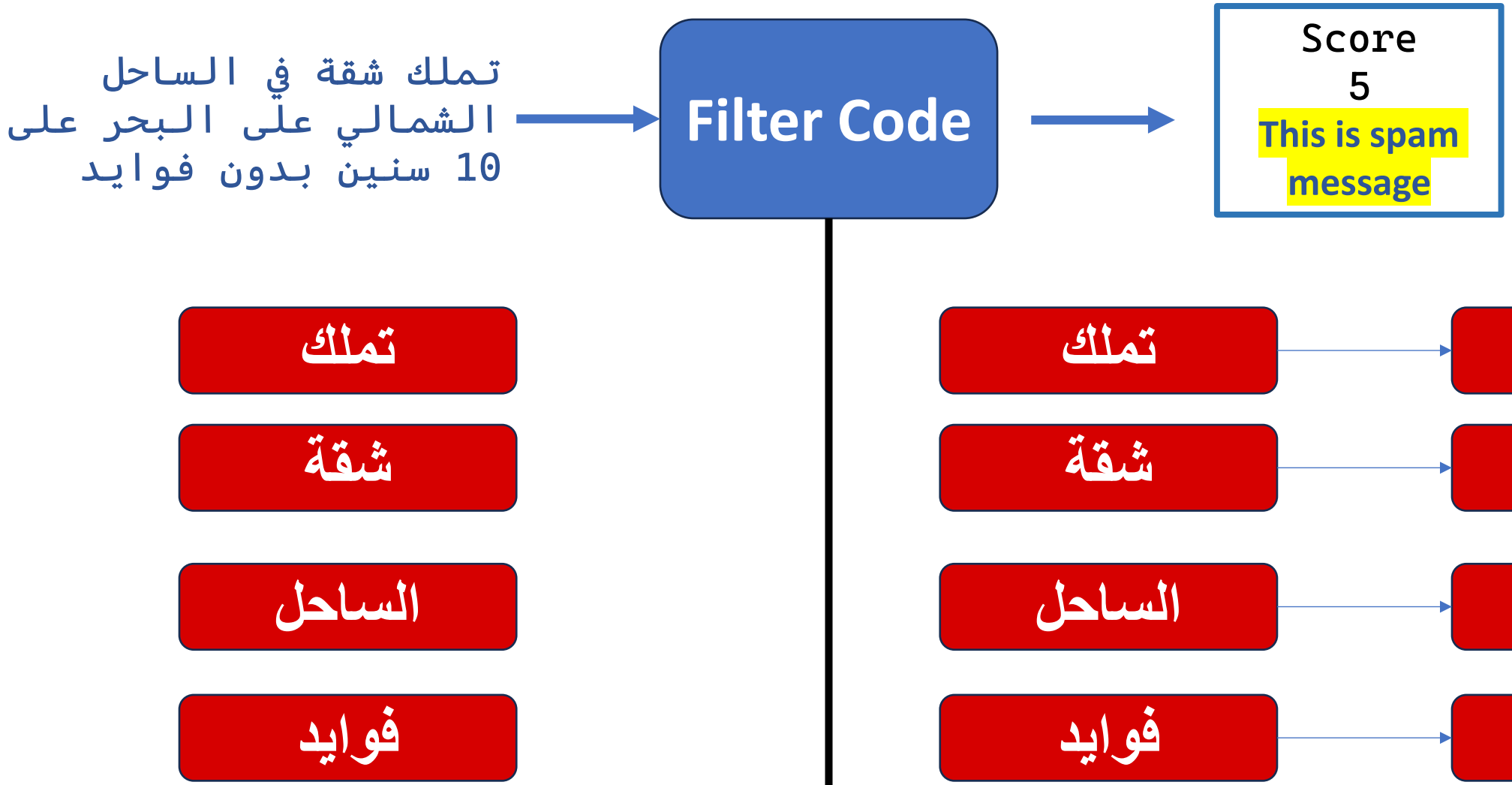
Machine learning is:

The field of study that gives computers the ability to learn without being **explicitly programmed**.

breakdown the concept:

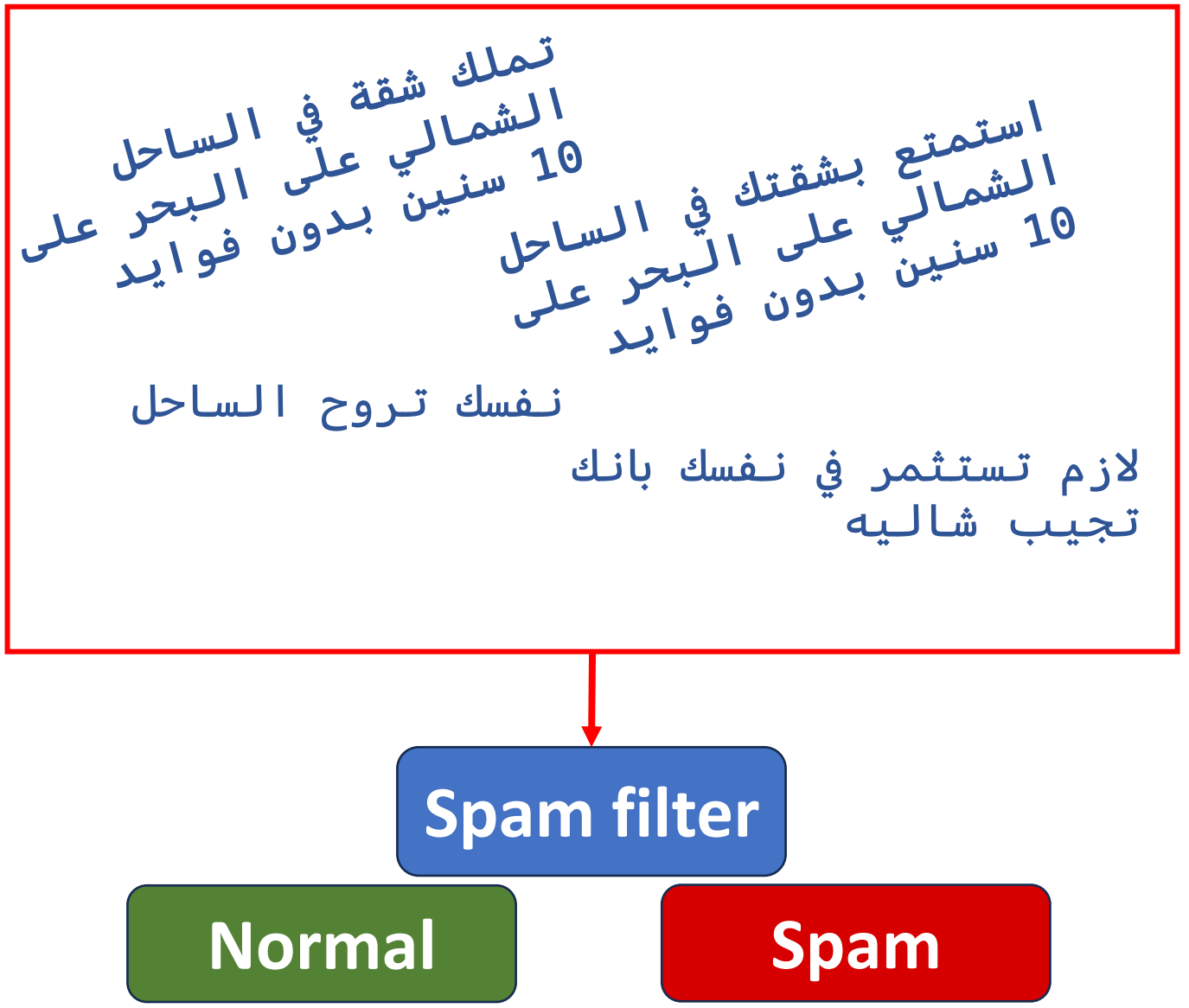
To help break down the concept lets suppose you want to create a spam filter for your WhatsApp to filter brokers from reaching you.

what’s Machine Learning?



what’s Machine Learning?.

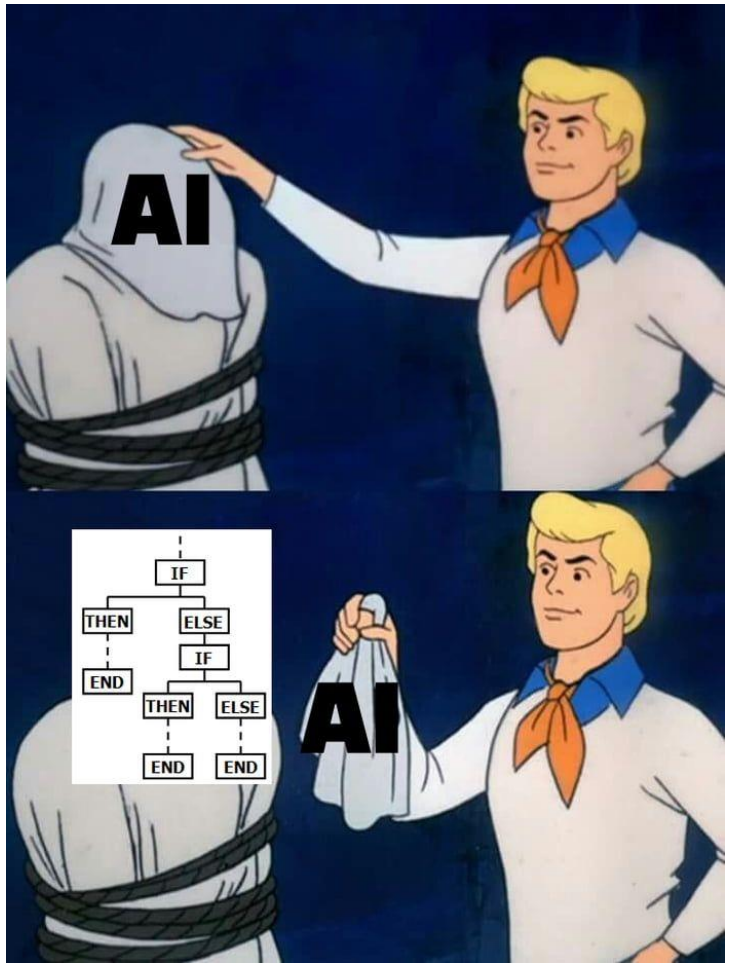
- machine learning can help us solve those things as it can learn from data.
- we will give the machine learning code huge amount of spam message to learn from it.
- The code will be intelligent enough to know the frequently used words in spam messages and know anything that looks like it
- This what’s called **spam filer**.



what’s Machine Learning?.

See how hard to write a spam filter in if, if else, else form as you need to cover **very long and infinite number of cases** this where beauty of machine learning happens.

That’s why they often called machine learning a **smart if** :D.



Course Grading

- You will be graded for

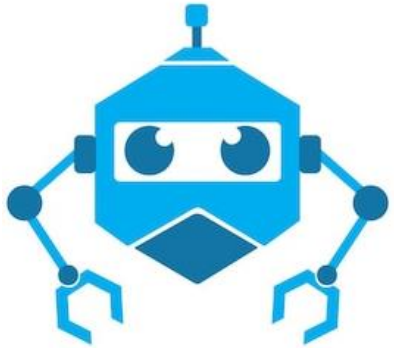
Attendance

Activity

Each lab will have **activity** you will need to do it.

Assignments

- Assignment description will be in lab material.
- Each assignment will be uploaded to MNU model.
- **Each assignment will have a deadline.**



Final project or practical Exam

AGAIN !! DEADLINES..

Labs Plan

Labs Plan.

- This this scope of the ML labs

python rev

revision of python:

- variables.
- if / if else
- loops
- functions
- lists / tuples.

Important libraries

using Pip.
Import libraries

- NumPy.
- matplotlib.
- pandas.

Machine learning fundamentals

python

Why python?

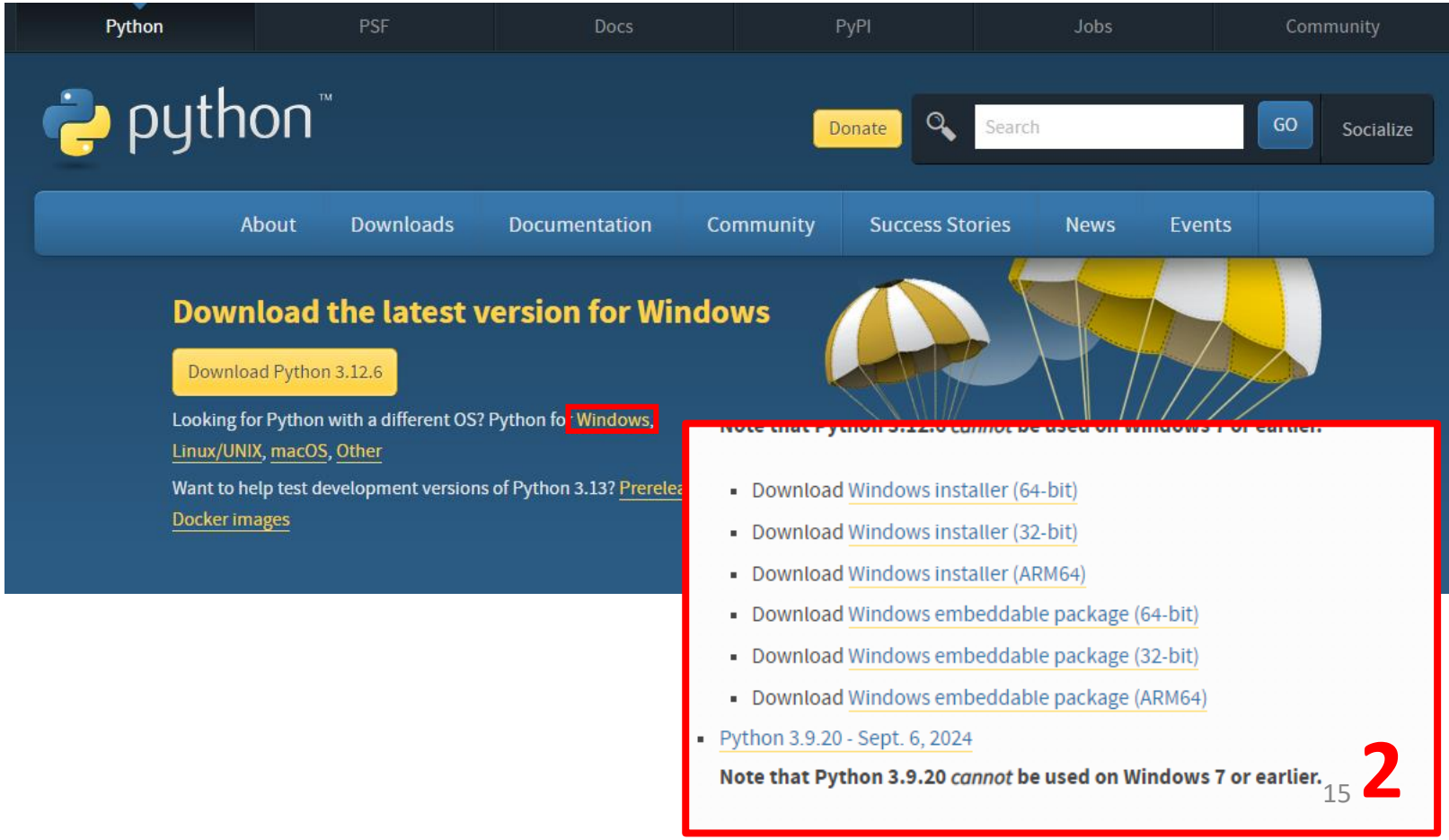
here are 5 most important features of python used in data science:

- **Easy to Learn and Use:**
  Python has a simple syntax, making it easy for beginners and professionals alike to learn and write code quickly.
- **Rich Ecosystem of Libraries:**
  Python offers powerful data science libraries like **NumPy**, **Pandas**, **Matplotlib**, **SciPy**, and **Scikit-learn** for data manipulation, analysis, and visualization.
- **Excellent for Machine Learning:**
  Libraries such as **TensorFlow**, **PyTorch**, and **Keras** make Python a top choice for machine learning and deep learning projects.
- **Strong Community Support:**
  Python has an active community of developers, making it easy to find tutorials, forums, and solutions to problems.
- **wide use & integration:**
  Python integrates well with other technologies and can be used for data preprocessing, model development, deployment, and production-level code.

python installation.

First go to python official download website:
https://www.python.org/downloads/

1. **Navigate to your current OS (windows)**
2. **install python version 3.12.**
3. **click on windows installer.**



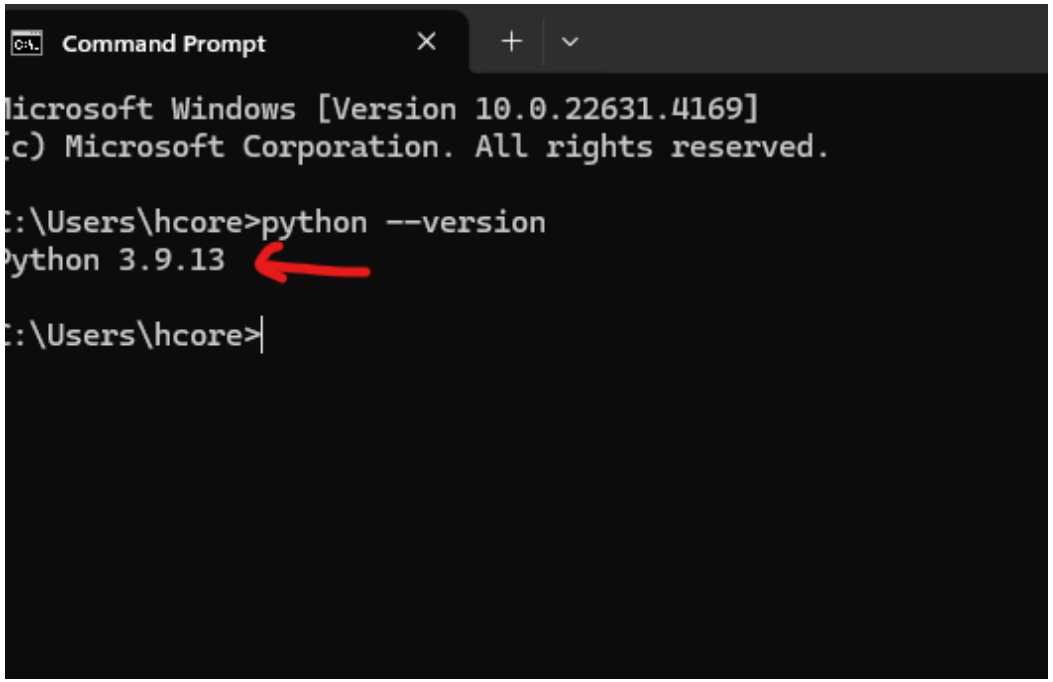
python installation.

To verify that python is installed

1. open terminal:
    1. in windows by right click on windows icon

       in taskbar then choose terminal (win11)

       or cmd (win10).
2. type “python --version”.
3. you should see something like the picture.
   note: you can have a different version number this is normal.



Python IDE

python IDE.

IDE : integration development environment

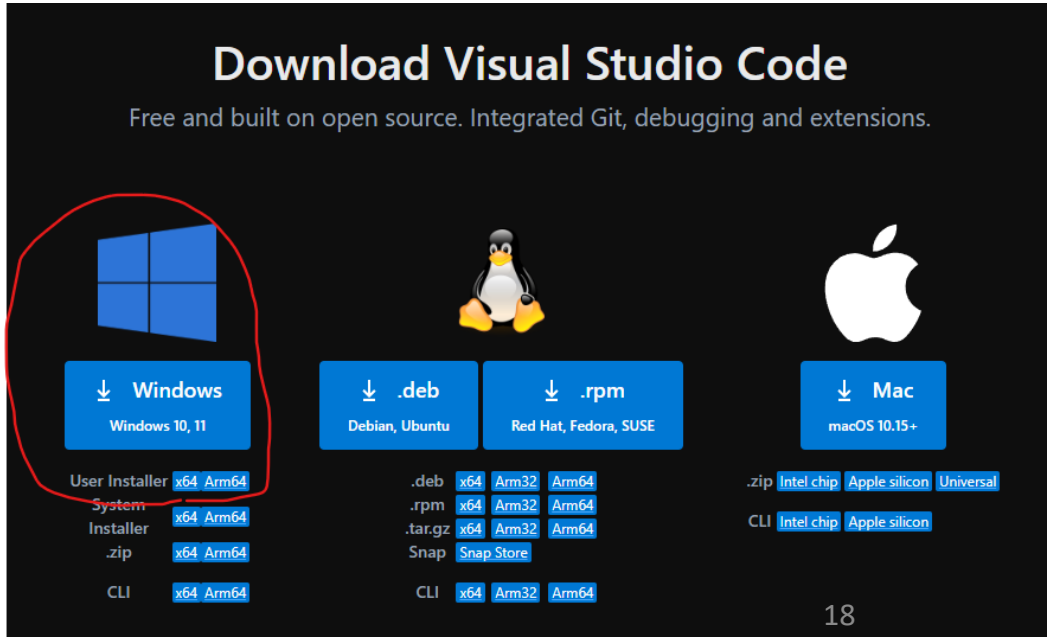
ide is used to write, debug and run code.

known python IDEs:

- PyCharm
- **Visual studio code (VSC).**
- **Jupyter notebook.**
- spyder.

We will use VSC and here is the steps to install it

- **Go to VSC official website: https://code.visualstudio.com/download**
- **download the program based on your OS and architecture.**
- **install the program.**



VS code.

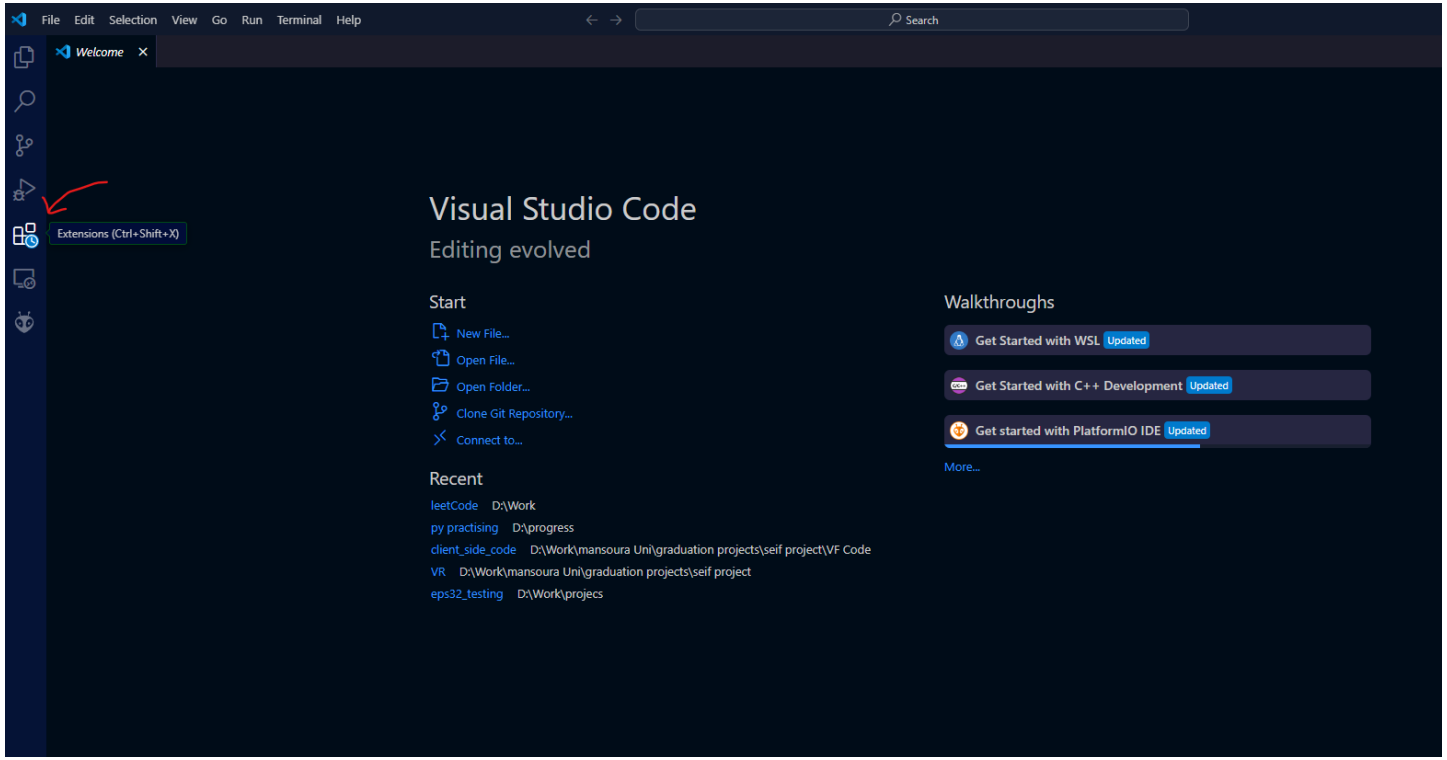
- VS code is a very famous ide and code editor for many programming languages.
- VS code biggest feature is **extensions**.
- **Extensions** are custom features that can be added to VSC to make him support new files or programming language.

To download extensions:

1. **from the left side of VS code go to the extension icon.**
2. **After this you will see the extensions, you already downloaded, or you can search for new extensions to download**

<u>We will need the following extension</u>:

- **python by Microsoft.**
- **code runner by Jun Han.**



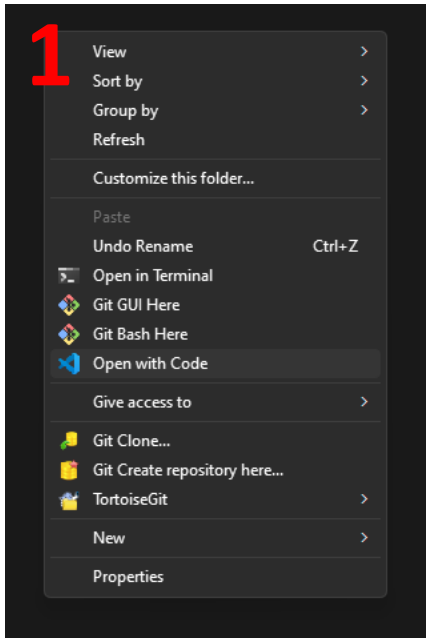
Your first python project

Hello world

VS code.

To create a new python script:

1. create a new directory (folder) for the future scripts.
2. right click on the folder and choose “**Open with Code**”.
3. Now go to the file section from the left side of the VSC.
4. click on the new file icon, then enter any file name with extension **“.py” this is very important**.
5. A new file will open in the work space type “ **print (“hello World”)** ”

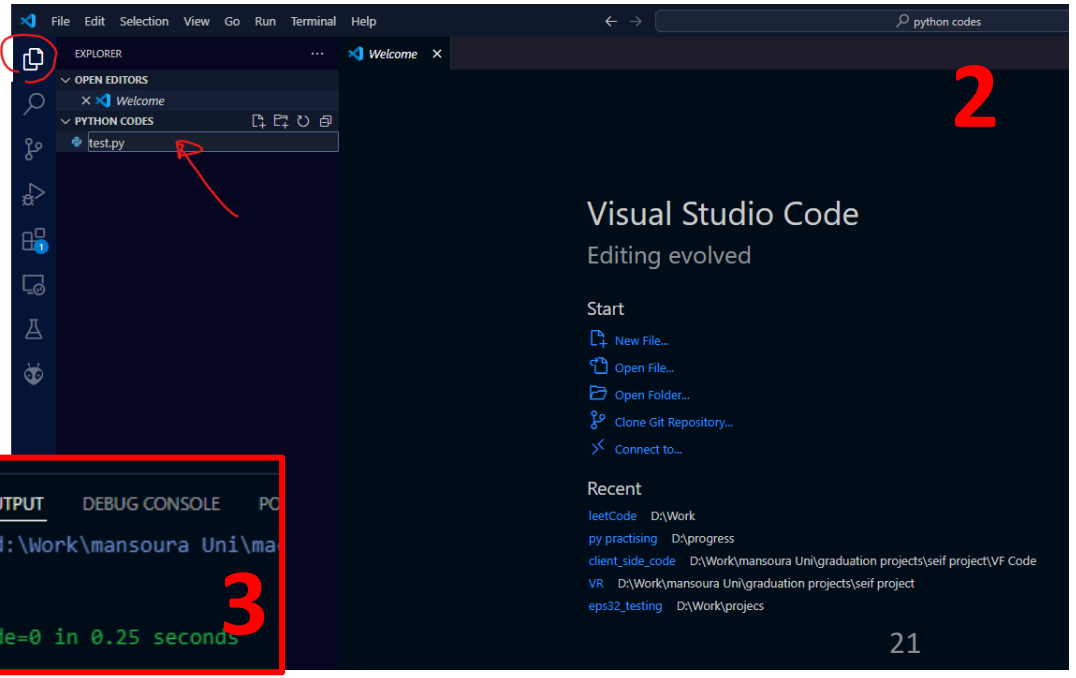


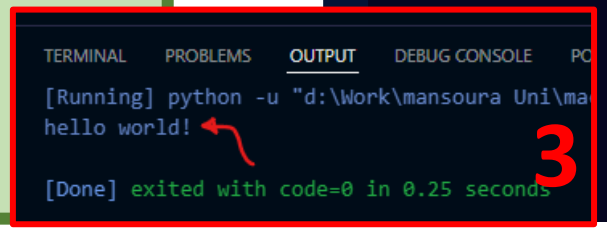
<u>To run the script</u>:

- **from Terminal >> new terminal.**
- **A terminal window will appear at the bottom, Type the line python test.py**
  note: change the file name with the one you wrote earlier.

<u>OR</u>

from the run icon in the upper right corner.





print function

python - print

print is used to output data or text to the terminal

```
print("hello world!")
```

```
print("hello ", "AIE", "It is me!")
```

```
print ("Hello AIE" , "\nthis is omar ", 26+1)
```

remember the scape sequence \n >> newline
\t >> tab

output

Hello world

Hello AIE It is me!

Hello AIE

this is omar 27

python - print

print is used to output data or text to the terminal

output

```
print("Hello ")
print("AIE!")
```

Hello

AIE!

```
print("Hello",end=" ")
print("AIE!")
```

Hello AIE!

```
print("hello",end="")
print("today we have ",3,"activity.")
```

hellotoday we have 3 activity.

end is used to define what should be printed at the end
by default, it is newline “\n”

Variables in python

variables in python.

variables are spaces located in computers' memory used to store specific data for different lifetimes.

- **declare & initialize a variable.**

```
x = 10 #declaring variables
y = 15.5
z = “hello”
print( x , “  “, y)
```

- **Note – important --.**

```
b = 18
b = "hello"
print(b)
```

output : hello

python variables is a dynamic type

here we changed the b variable from integer to string

Variables Data type

Data types.

python primitive(basic) data types.

Numbers

Integer Numbers:

1,2,3,50

0 , -1 , -5 , -90

Float numbers:

12.5 , 1000.556 , 10.0

```
>>> b = 10.0
>>> type(b)
<class 'float'>
```

Strings

characters:

a,b,c,d, .. x,y,z

A,B,C,D

& , ^ , < , >

strings :

Omar wahba , Helllo world

```
>>> c = "123"
>>> type(c)
<class 'str'>
>>> c = "h"
>>> type(c)
<class 'str'>
```

boolean

logical data type contains only

True , False

```
>>> x = True
>>> type(x)
<class 'bool'>
```

Arithmetic Operations

Arithmetic Operations

what's the output of the following code?

```
a = 10
b = 2
print("res = ",a + b)
print("res = ",a - b)
print("res = ",a * b)
print("res = ",a / b)
print("res = ",a // b)
print("res = ",a % b)
print("res = ",a ** b)
```

output

```
res =  12
res =  8
res =  20
res =  5.0
res =  5
res =  0
res =  100
```

Explanation

// >> floor division → integer division (don't care about floating points)

% >> reminder

** >> Power or exponential

Input function

python - input

input function is used to take data from the user.

```
name = input("please enter your name: ")
print("welcome ",name)
------------------output------------------
>> please enter your name: omar wahba
welcome  omar wahba
```

optional message to be displayed to the user

Note:
The input function **pauses** the program execution until the user enter the data

Note the following code

```
num = input("please enter a number: ")
print("result = ", num + 5)
```

```
------------------output------------------
>> TypeError: can only concatenate str (not "int") to str
```

Note:
the input function only return type 'str'.
to convert to an integer simply modify this line:

```
num = int(input("please enter a number: "))
```

try to search about casting in python

Strings

Python – Strings

what's the output of the following code?

```
str1 = "Hello"
str2 = " World"
result = str1 + str2
print("res of (+):", result)

str3 = "AIE! "
result = str3 * 3
print("res of (*):", result)
```

output

```
Hello World
AIE! AIE! AIE!
```

Explanation

(+) → String concatenation

(*) → Repetition

Python – Strings

what's the output of the following code?

```
# 3. Indexing
str4 = "Hello, Python!"
f_char = str4[0]
l_char = str4[-1]
substring = str4[7:13]
print("Indexing [0]:", f_char)
print("Indexing [-1]:", l_char)
print("Slicing [7:13]:", substring)

# 4. String Length
length = len(str4)
print("Length of string:", length)
```

output

```
Indexing [0]: H
Indexing [-1]: !
Slicing [7:13]: Python
Length of string: 14
```

Explanation

[ ] → This is called indexing is: accessing individual elements (characters, items) from a sequence.

len() → function is used to get number of elements in a sequence.

Control Flow if / else

Python – Strings

if / else is used to execute some code based on a certain condition

```
testNum = int(input("pls enter a num: "))
if testNum > 0:
    if testNum % 2 == 0:
        print("Even number")
    else:
        print("Odd number")

#outside if
```

- **if block of code**
- **This block will be executed when the condition = True.**

else condition

- gets executed if the condition is False.
- The else got a block like if

Important note

In Python, **code blocks** are managed using **indentation** rather than curly braces {}.

Python – Strings

if / elif / else is used when there is multiple conditions

```
a = 1
b = 5
operation = '+'
res = 0
if operation == '+':
    res = a+b
elif operation == '-':
    res = a - b
elif operation == '*':
    res = a * b
elif operation == '/':
    res = a // b
else:
    print('invalid operation')
print("result = ",res)
```

output

6

if elif else

- it is very fast compared to normal (if) as python doesn't need to check other conditions.

Relational operators

Relation operators

Those operators are used to compare different objects or create conditions

```
print(1 > 5)
print(1 < 5)
print(1 == 5)
print(1 != 5)
print("OMAR" == "omar")
print( 5 > 5)
print( 5 >= 5)
print( 3 <= 5)
```

output

```
False
True
False
True
False
False
True
True
```

logical operators

Relation operators

Those operators are used to combine multiple conditions

```
print(15 > 5 and 1 > 5)
print(15 > 5 or 1 < 5)
print(15 > 5 and 1 < 5)
print(15 < 5 or -1 > 4)
print(not 15 == 15)
print(not ( (13 == 13 and 11 < 100) or 17 < 30) )
```

output

```
False
True
True
False
False
False
```

Logical operators can be used in (if) conditions like

if gpa >= 3.6 and gpa < 4 :

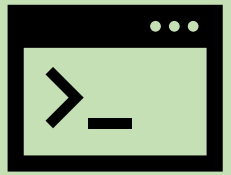
Activity (1)
Simple Coffee shop App

Act (1).

create a Python script to help a coffee shop calculate customers' receipts:

- **The café has one drink (coffee) and two cup size option (L : 10 LE , M : 5).**
- **The program should ask the user to enter the number of cups first.**
- **Then, the user will enter Large or medium using the letters "L" or "M"**
- **The program should calculate the total price and print it for the user.**

Example of program running



```
pls, enter number of cups: 2
pls enter L for Large size or M for medium size: L
====================== Check out ============================
your Final cost is :  20  LE.
====================== Thank you ============================
```

Assignment (1)

Assign (1)

Q1) Triangle Type Checker

Problem:

Write a program that takes the lengths of three sides of a triangle and determines if it is a valid triangle. If **valid**, classify the triangle as:

- Equilateral: All sides are equal
- Isosceles: Two sides are equal
- Scalene: All sides are different

Hint: valid triangle when the sum of two sides must be greater than the other side.

Q2) Number Classification

Problem:

Write a program that takes an integer as input and classifies it as:

- Positive
- Negative
- Zero Additionally, if the number is positive, check whether it is **even** or **odd**.

Assign (1)

Q3) Leap Year Checker

Problem:

Write a program that takes a year as input and checks if it is a leap year. A year is a leap year if:

- It is divisible by 4,
- But not divisible by 100, unless it's also divisible by 400.

More resources for python

Full Python course:

https://www.youtube.com/watch?v=XKHEtdqhLK8&ab_channel=BroCode

you can start from part 1 to part 24.

More added knowledge is also recommended

Thank You!!