Python

Ahmed Al Hilli
Al-Furat Al-Awsat Technical University

Workshop Syllabus

- Day 1: Introduction to Python
- Day 2: Conditional Statements (control statements)
- Day 3: Loops and list comprehension in Python
- Day 4: Functions and lambda functions in Python
- Day 5: Working with text files and Regular Expressions in Python

Ahmed Al Hilli
Al-Furat Al-Awsat Technical University

Day 1

Introduction to Python

Ahmed Al Hilli
Al-Furat Al-Awsat Technical University

What we will cover in this lecture

- Install Python
- Working with IDE
- Variables in Python

Ahmed Al Hilli
Al-Furat Al-Awsat Technical University

What will be covered today

- What is Python
- How to install Python
- Python Integrated Development Environment (IDE)
- Hello world example
- Variables, Variables assignment, and basic operations

Ahmed Al-Hilli
Al-Furat Al-Awsat Technical University

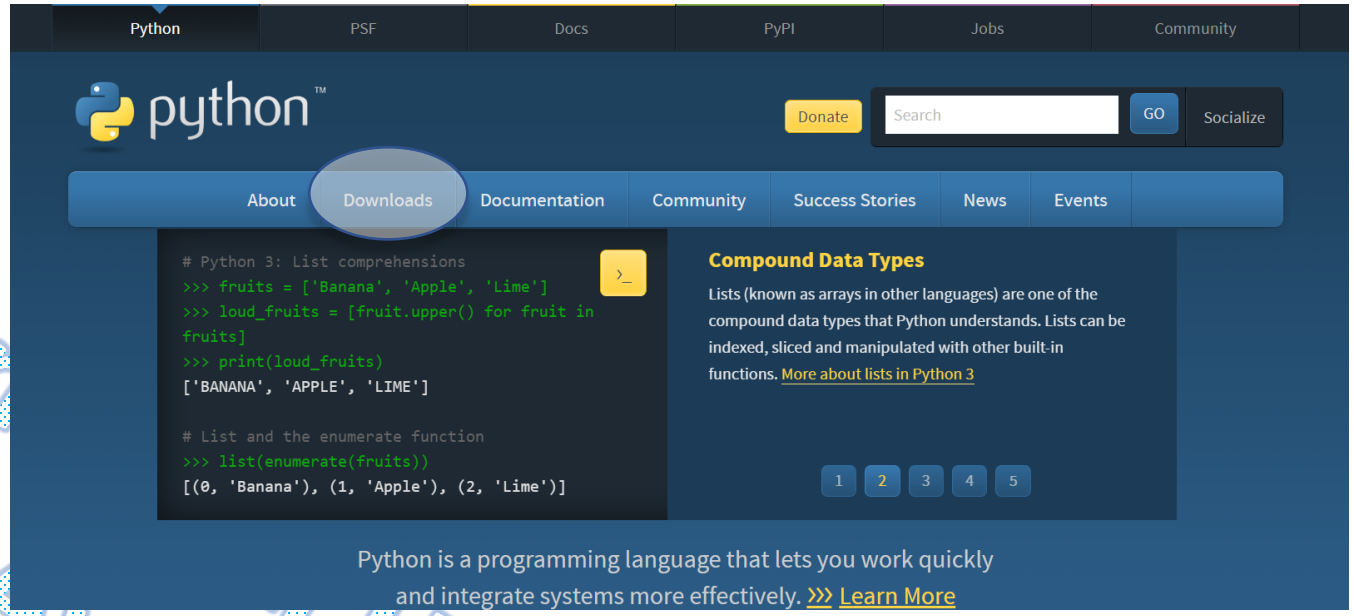
What is Python

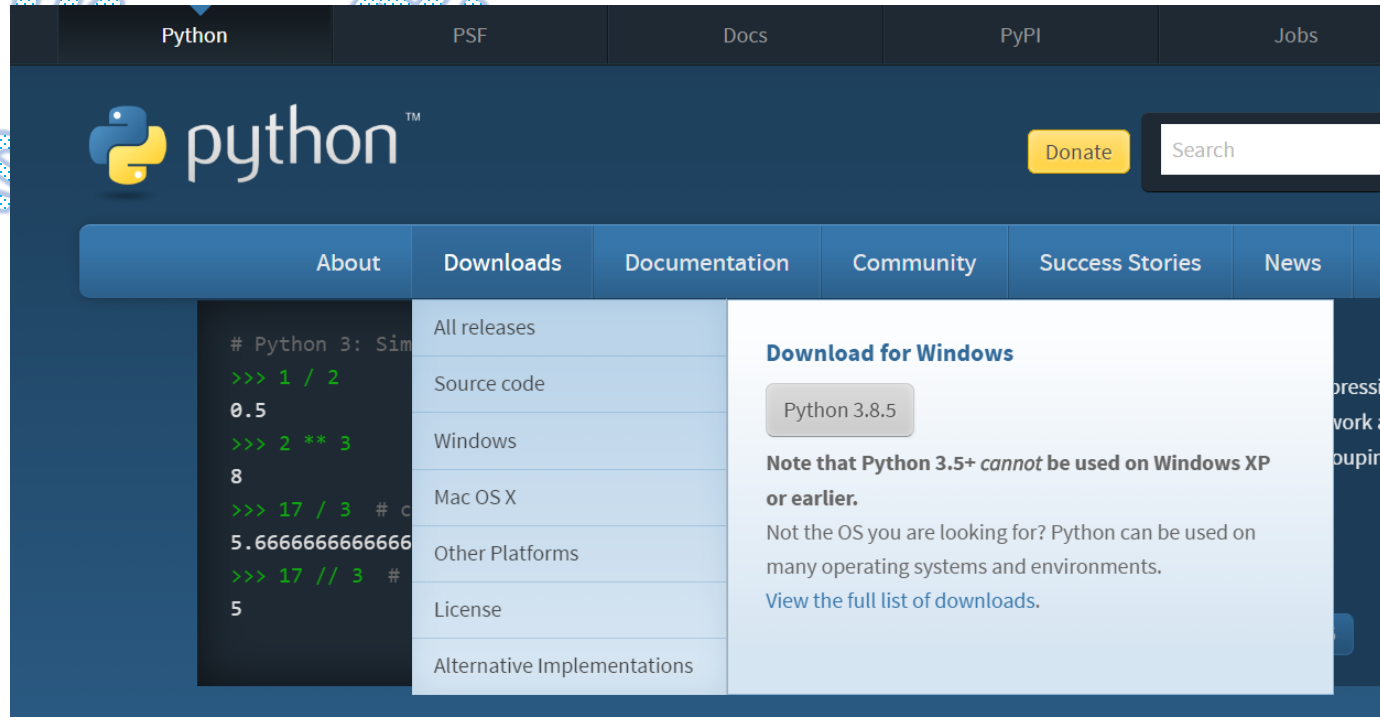
- According to Wikipedia, “Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.”

Ahmed Al-Hilli
Al-Furat Al-Awsat Technical University

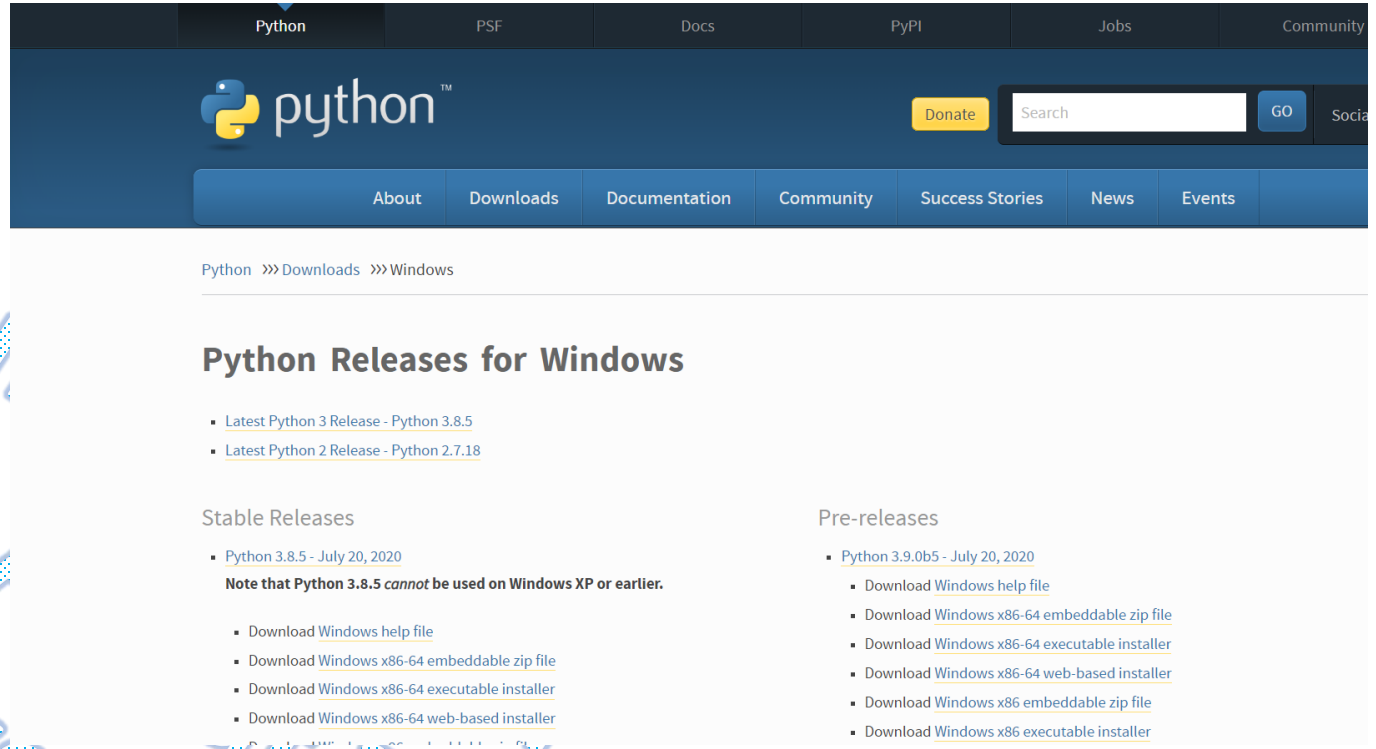
Install Python

- Go to www.python.org
- Scroll to download
- Choose your OS and version (we use 3.8 in this slides)
- After downloading, install





- Choose your OS (Windows in my case)



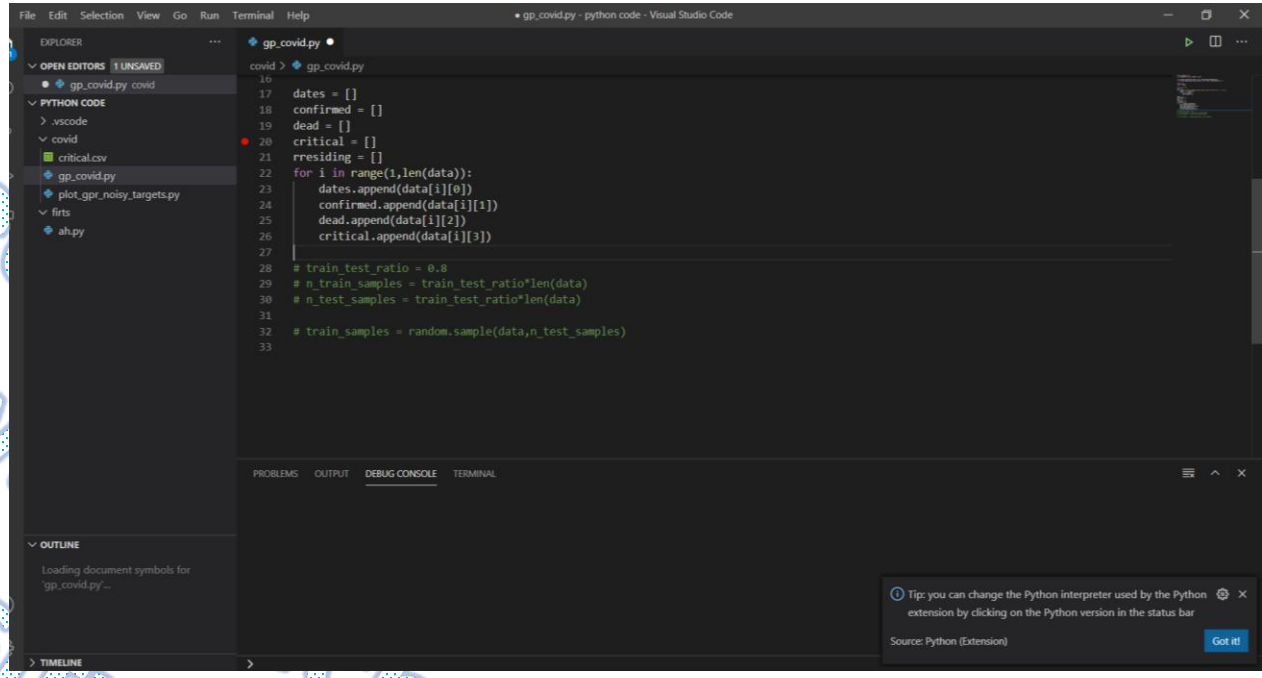
Al-Furat Al-Ahlia University Technical

- Make sure to enable
  “Add python 3.8 to PATH”

Ahmed Al Hilli
Al-Furat Al-Awsat Technical University

Integrated Development Environment (IDE)

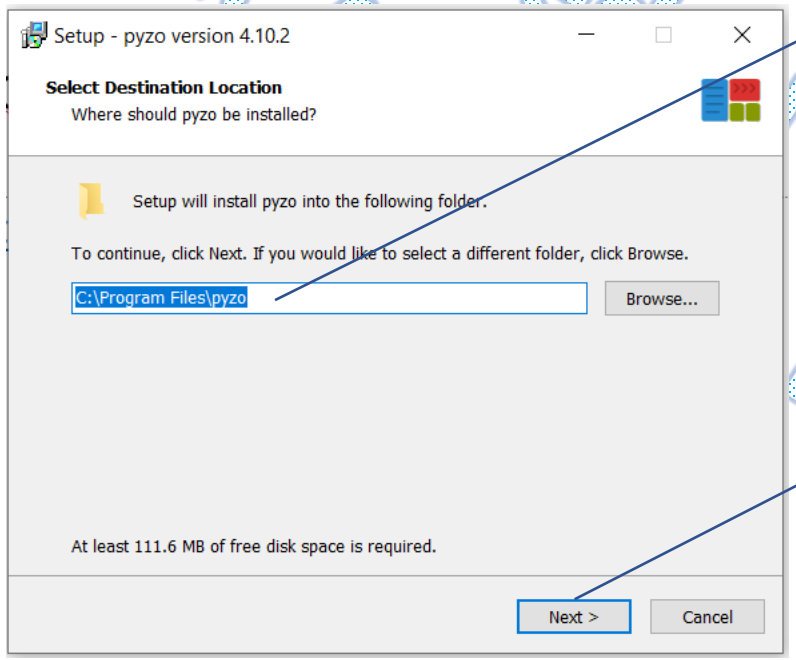
- IDE: An environment that helps programmers in writing programs
- Why using IDE instead of Editors?
  - Debugging
  - Advanced features (like autocomplete, showing help)
- Python IDEs:
  - pyzo (the one I prefer)
  - Microsoft Visual Studio
  - and many others
- Installing IDE:
  Simply download the preferred IDE, and install



Ahmed Al-Furaiji Al-Awadi University of Technical

Installing Pyzo

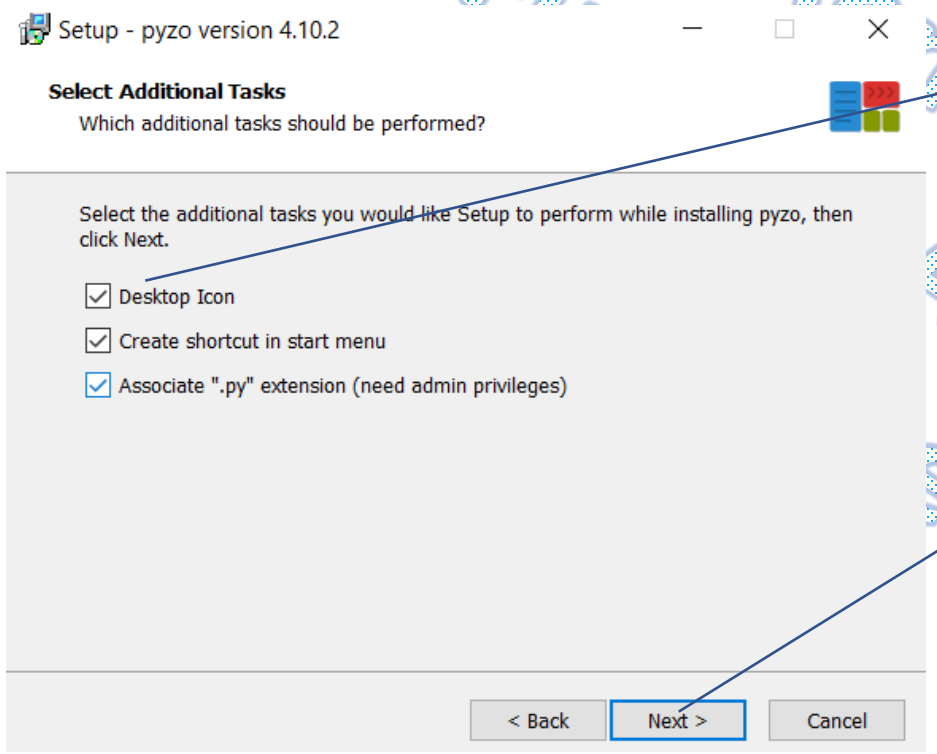
- Download form: https://pyzo.org/start.html



Choose installation folder

Click Next

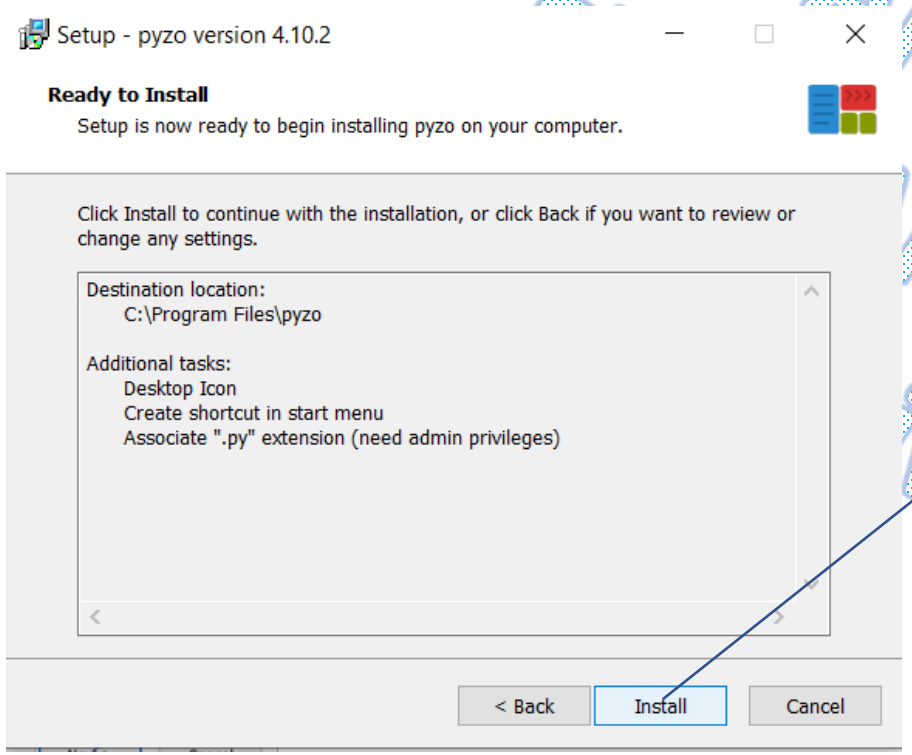
Al-Furqan Ahmed Hilli Technical



Check or uncheck

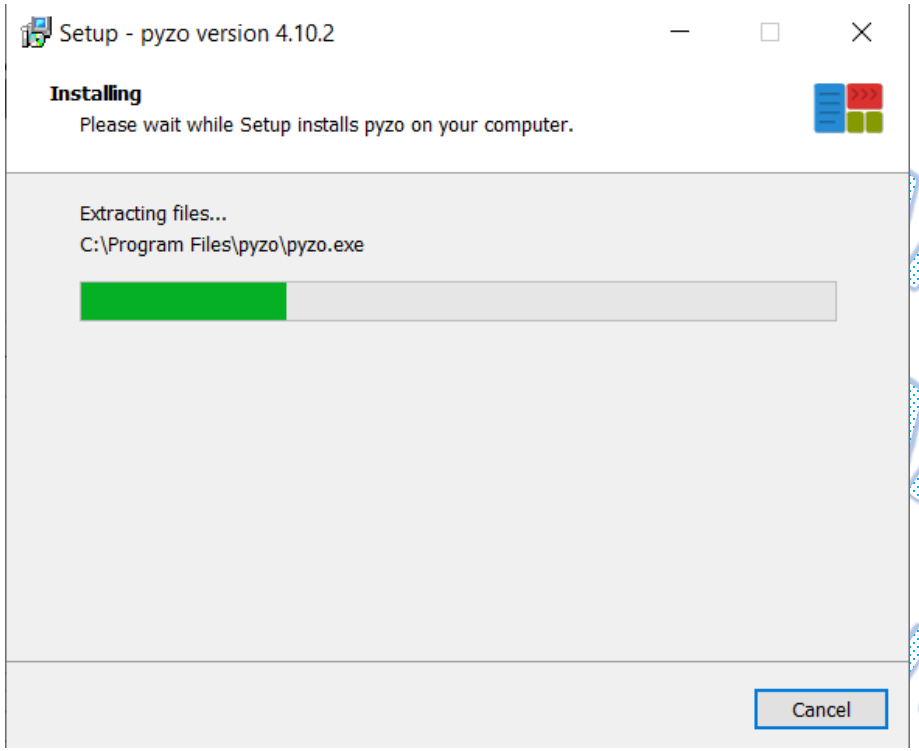
Click Next

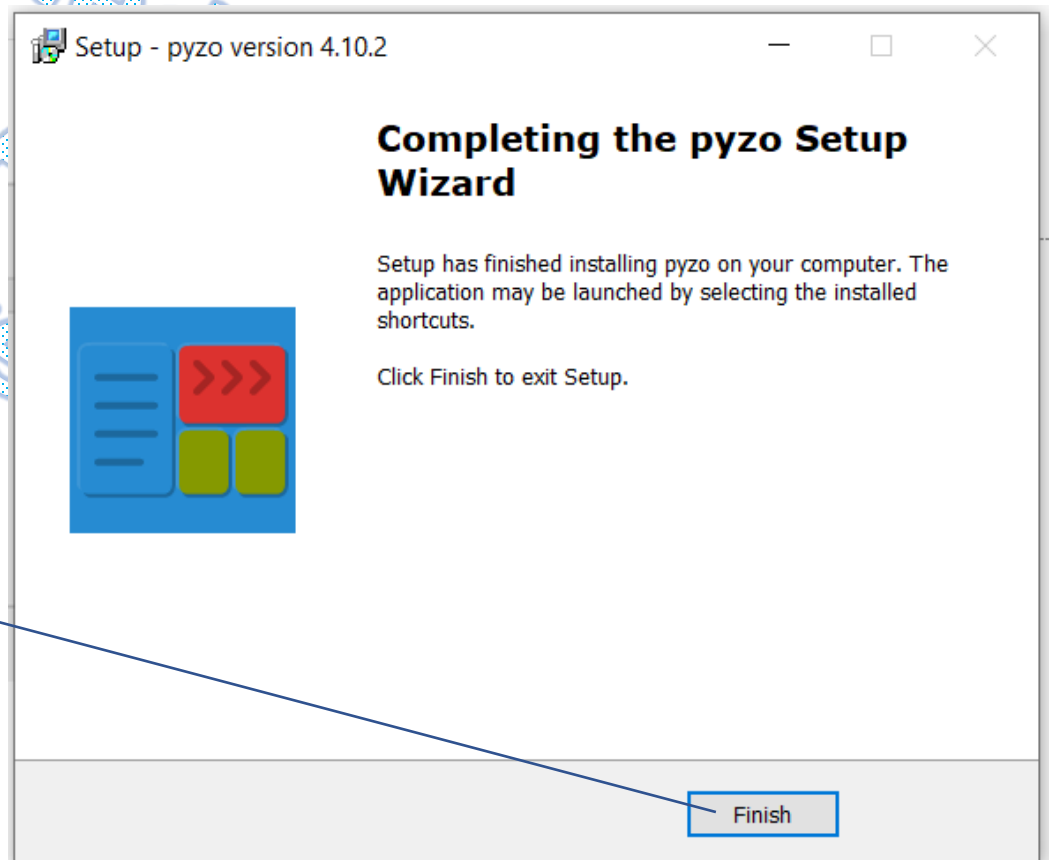
Al-Fu... Ahmed Al H... Awsat Technical ...ersity



Click Install

Al-Ahmed Al-Awsat Technical University

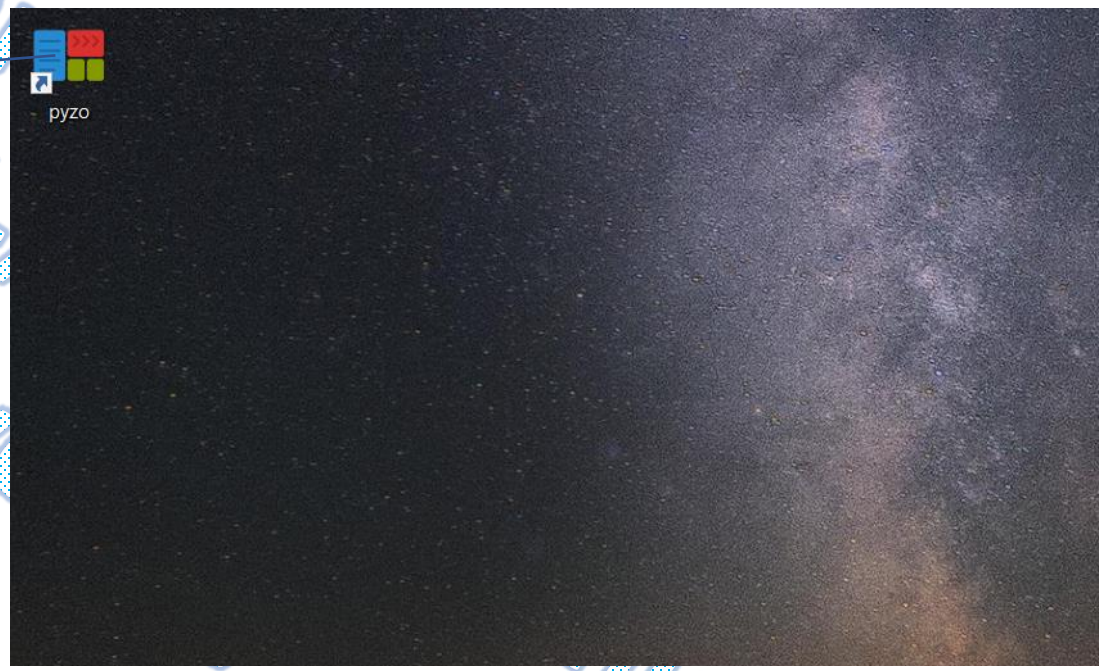


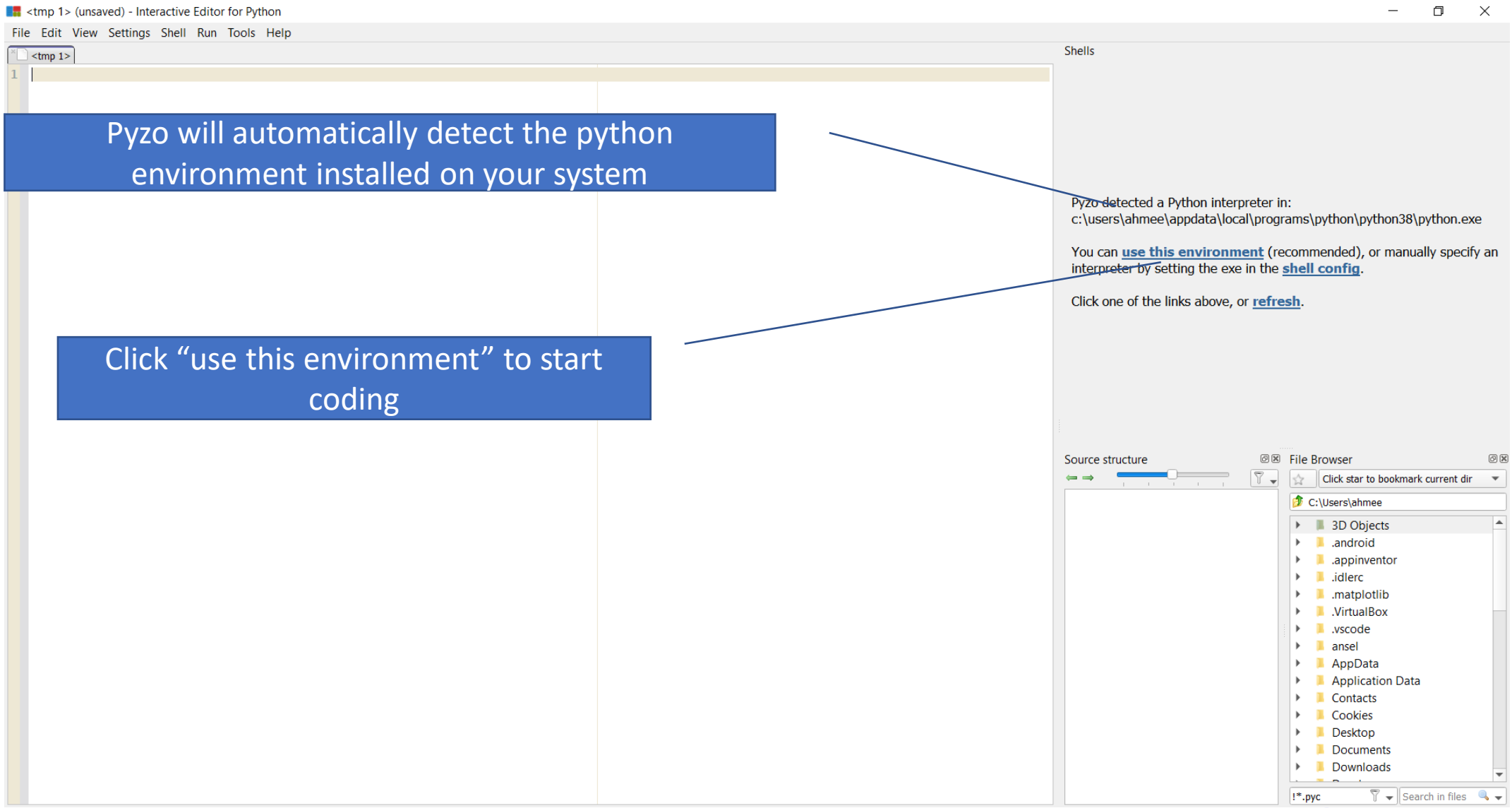


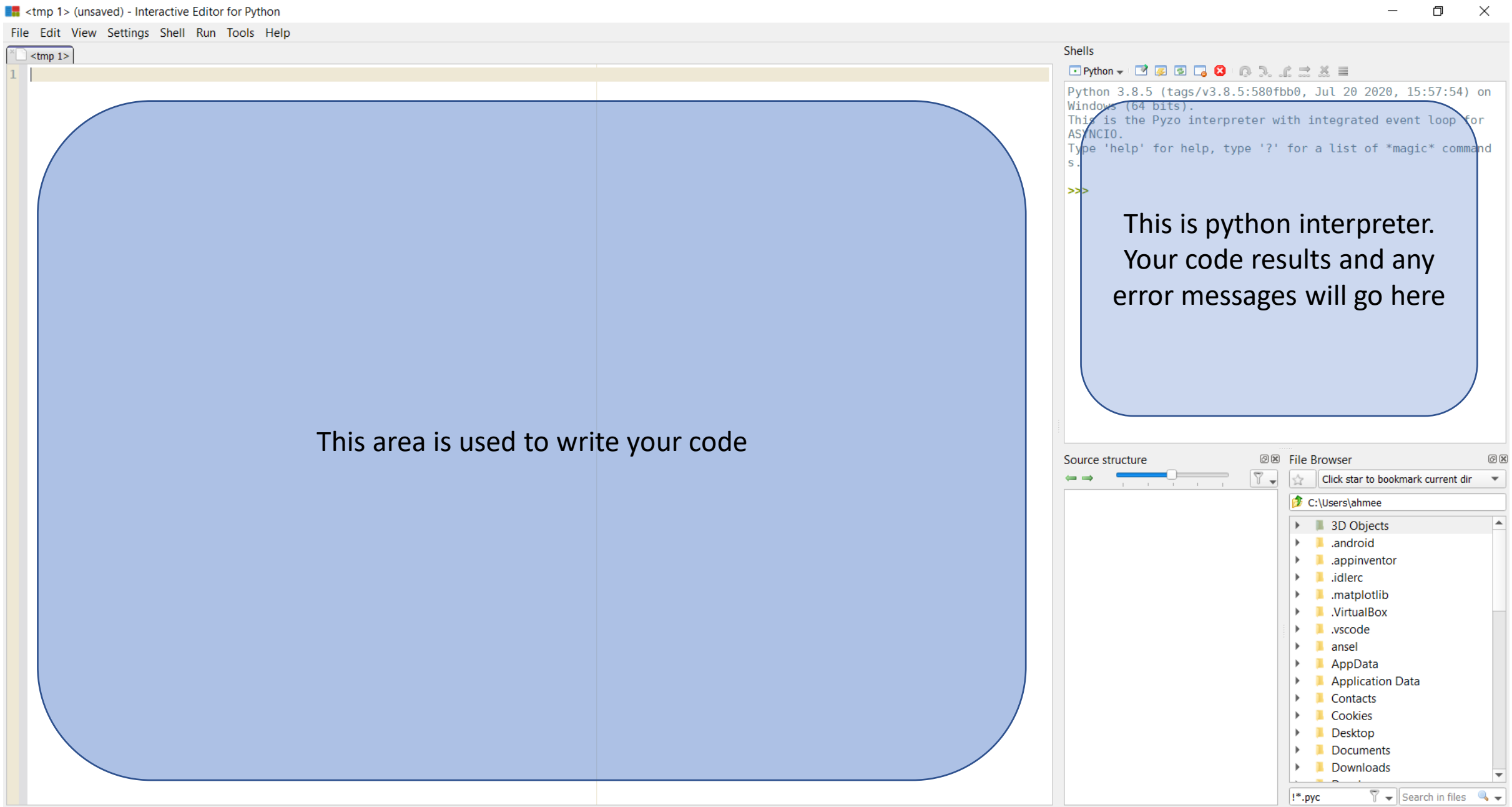
Click Finish

Al-Furat Al-
Ahl
Univ
nical

Double click
to start




<tmp 1> (unsaved) - Interactive Editor for Python
File Edit View Settings Shell Run Tools Help
<tmp 1>
1
Pyzo will automatically detect the python environment installed on your system
Click “use this environment” to start coding
Shells
Pyzo detected a Python interpreter in:
c:\users\ahmee\appdata\local\programs\python\python38\python.exe
You can use this environment (recommended), or manually specify an interpreter by setting the exe in the shell config.
Click one of the links above, or refresh.
Source structure
File Browser
Click star to bookmark current dir
C:\Users\ahmee
3D Objects
.android
.appinventor
.idlerc
.matplotlib
.VirtualBox
.vscode
ansel
AppData
Application Data
Contacts
Cookies
Desktop
Documents
Downloads
!*.pyc
Search in files


<tmp 1> (unsaved) - Interactive Editor for Python
File Edit View Settings Shell Run Tools Help
<tmp 1>
1
This area is used to write your code
Shells
Python
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) on Windows (64 bits).
This is the Pyzo interpreter with integrated event loop for ASYNCIO.
Type 'help' for help, type '?' for a list of *magic* commands.
>>>
This is python interpreter. Your code results and any error messages will go here
Source structure
File Browser
Click star to bookmark current dir
C:\Users\ahmee
3D Objects
.android
.appinventor
.idlerc
.matplotlib
.VirtualBox
.vscode
ansel
AppData
Application Data
Contacts
Cookies
Desktop
Documents
Downloads
!*.pyc
Search in files

Simple Example

```
#
# This code prints Hello World!! on python console
#
#


print("Hello World!!")
```

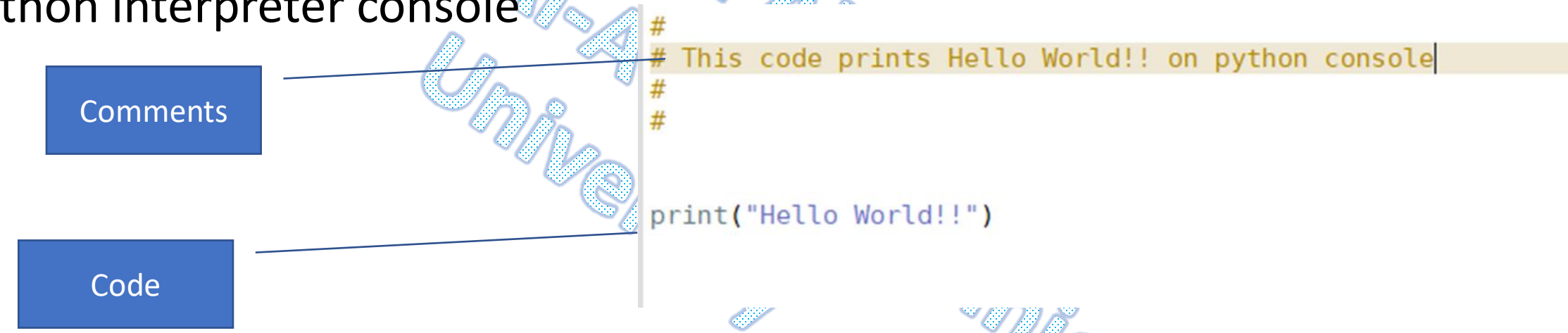
- On Pyzo, go to File->New
- Type the following code:

  print('Hello World!!')

- To run the code, press 'F5' on your keyboard
- To save your code, go to File->save or you can press 'ctrl+s', then enter the name of the file code, and press 'save'

Al-Furat Al-Awsat Technical University
Ali Hilli

- What is ‘print(‘Hello World!!’)’ do?
  print is a build in function that takes text as input, and prints it on Python interpreter console



Al-Furat Al-Awsat Technical University

Comments

- In python, any line that starts with ‘#’ are ignored by the interpreter
- These lines usually called “comments”
- Comments are important part of your code, because it increases the readability of your code
- Also, comments sometimes are used to stop running line or lines of code in debugging

Al-Furat Al-Awsat Technical University
Ahmed Al-Hilli

- For writing multiple lines comment, put your comments lines between ‘’’ ‘’’
-

```
#
# This code prints Hello World!! on python console
#
#

'''
All the lines here will be ignored by python interpreter
This is useful for writing multiple lines comments
'''

print("Hello World!!")
```

Ahmed Al-Fuwati Al-Awsat Technical University

White Spaces

- In python, white spaces matter. Unlike most programming language, white spaces are important in Python

```
#
# This code prints Hello World!! on python console
#
#

'''
All the lines here will be ignored by python interpreter
This is useful for writing multiple lines comments
'''


    print("Hello World!!")
```

```
>>> (executing file "hello.py")
  File "C:\Users\ahmee\python slides code\hello.py", line 12
    print("Hello World!!")
    ^
IndentationError: unexpected indent
```

- Indent and white spaces are used in python to define blocks of code (more on this comes later)

Ahmed Al Ha
Al-Furat Al-Awsat Technical
University

Variables in Python

- Variable: is a place in memory to hold data
- You can only hold one thing in one variable
- You can always get the stored data inside the variable by its name
- The name of the variable should start with letter or underscore
- Variable names are case sensitive (variable ‘A’ and variable ‘a’ are different)

Ahmed Al-Hilli
Al-Furat Al-Awsat Technical University

Addition Example

```
#
# This code Adds two numbers
#
#

number1 = 10
number2 = 20
_result = number1+number2

print(_result)
```

The name of the first variable to hold the first number

The name of the second variable to hold the second number

A variable to store the result of adding

Ahmed Al Hiti at Al-Awsat Technical University

Basic Math Operations

```
# This code illistrates the basic arithmatic operations in python

num1 = 10
num2 = 3

sum = num1 + num2 # the sum
sub = num1 - num2 # subtraction
div = num1/num2 # division
reminder = num1 % num2 # reminder
square = num1**2 # square of num1

print(sum,sub,div,reminder,square)
```

```
>>> (executing file "basic arithmatic operations.py")
13 7 3.3333333333333335 1 100
```

Variables Type

- There are many types
- Here, we discuss only four variable types
  - int: a whole number like (7,6,-5,0)
  - float: a number with fraction (3.2, -3.09, 2.0)
  - String: text (‘Ahmed’, ‘Hello world!!’)
  - Boolean: a variable that is either True or False

- The utility function ‘type(var)’
  This function return the variable type ‘var’

Ahmed Al Hilli
Al-Furat Al-Awsat Technical University

```python
# This file illistrates different variable types

a_int = 7
a_float = 3.4
a_string = 'Hi, this variable is string'
a_bool = True

print(a_int,type(a_int))
print(a_float,type(a_float))
print(a_string,type(a_string))
print(a bool,type(a bool))
```

```
>>> (executing file "variable types.py")
7 <class 'int'>
3.4 <class 'float'>
Hi, this variable is string <class 'str'>
True <class 'bool'>
```

Al Hilli
Awsat Technical
University

Back to Basic Operations example

```
# This code illistrates the basic arithmatic operations in python

num1 = 10
num2 = 3

sum = num1 + num2 # the sum
sub = num1 - num2 # subtraction
div = num1/num2 # division
reminder = num1 % num2 # reminder
square = num1**2 # square of num1

print(sum,sub,div,reminder,square)
```

- It is better to ask the user to input the numbers rather than changing them in the code!!
- How to accept input from user?
  We can use the function ‘input()’, which returns string represents user’s input

```python
# Basic math operations with user's input

num1 = input("Enter the first number: ")
num2 = input("Enter the second number: ")

sum = num1+num2
sub = num1 - num2

div = num1/num2 # division
reminder = num1 % num2 # reminder
square = num1**2 # square of num1

print(sum,sub,div,reminder,square)
```

```
>>> (executing file "basic arithmatic operations with user input.py"
)
Enter the first number: 10
Enter the second number: 2
Traceback (most recent call last):
  File "C:\Users\ahmee\python slides code\basic arithmatic operation
s with user input.py", line 7, in <module>
    sub = num1 - num2
TypeError: unsupported operand type(s) for -: 'str' and 'str'
```

Why??

Variable recasting

- Changing variable type to another type
- Functions like int(), str(), and float() can recast a variable to its type

```
x = int(1)   # x will be 1
y = int(2.8) # y will be 2
z = int("3") # z will be 3
```

```
x = float(1)     # x will be 1.0
y = float(2.8)   # y will be 2.8
z = float("3")   # z will be 3.0
w = float("4.2") # w will be 4.2
```

```
x = str("s1") # x will be 's1'
y = str(2)    # y will be '2'
z = str(3.0)  # z will be '3.0'
```

Ahmed Al Hilli
Al-Furat Al-Awsat Technical University

```python
# Basic math operations with user's input
'''
num1 = input("Enter the first number: ")
num2 = input("Enter the second number: ")
'''
num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))

sum = num1+num2
sub = num1 - num2

div = num1/num2 # division
reminder = num1 % num2 # reminder
square = num1**2 # square of num1

print(sum,sub,div,reminder,square)
```

ed Al Hilli
wsat Technical
sit

```
>>> (executing file "basic arithmatic operations with user input.py")
Enter the first number: 10
Enter the second number: 3
13 7 3.3333333333333335 1 100
```

Thank you for listening

Questions?

Ahmed Al Hilli
Al-Furat Al-Awsat Technical University

Day 2

Conditional Statements (control statements)

Ahmed Al Hilli
Al-Furat Al-Awsat Technical University

Overview of what will cover in this lecture

- Bool variables
- Conditional statements
- Strings and Lists

Ahmed Al Hilli
Al-Furat Al-Awsat Technical University

Bool Variables

- Bool variables are variables that can only have two possible values: True or False
- It helps in testing some conditional statements
  ex: 3>5 (false)
  so in python, if you write the following code:

```
>>> my_bool_var = 3>5

>>> my_bool_var
False
```

- As shown in the example, we use “>” to test a statement
- These kind of operands are called “Relational Operators”

Ahmed Al Hilli
Al-Furat Al-Awsat Technical University

Relational Operators

Al-Fuad Ahmed

- Let ‘a’ and ‘b’ be two variables with assigned values (a,b can be int, float, or even str), or two constants (numbers, or text)

```
''' Relational Operators

a > b : Ture if a is bigger than b, false otherwise
a < b : Ture if a is less than b, false otherwise
a >= b : Ture if a is bigger than or equal to b, false
a <= b : Ture if a is less than or equal to b, false o
a == b : Ture if a equals b, false otherwise
a != b : Ture if a is not equal b, false otherwise

'''
```

```
bool1 = 12 > 13
bool2 = 12 == 12
bool3 = 12 != 12
bool4 = 3 >= 0
bool5 = 2 <= 0
bool6 = 2 < 4

print(bool1, bool2, bool3 , bool4, bool5, bool6)
```

```
>>> (executing file "<tmp 1>")
False True False True False True

>>>
```

Al Hilli
Technical

Composite Bool Expression

- You can connect two or more bool expressions to form more complicated bool statement
- How to connect two bool expressions? and, or, not

Expression 1	Expression 2	And	Or
Ture	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

Expression	Not
True	False
False	True

```
''' and, or, not use in python
'''


e1 = True
e2 = True
e3 = False
e4 = False



print(e1 and e2)
print(e1 or e2)
print(e1 and e4)
print(not e3)
```

```
>>> (executing file "<tmp 2>")
True
True
False
True

>>>
```

hmed Al Hilli
l-Awsat Technical
iversity

```python
# program to create complex bool expression

num = int(input('Enter a number: '))

bool_test = (num > 10) and (num < 20)

print('Test if the number you\'ve entered between 10 and 20:',bool_test)
```

```
Enter a number: 10
Test if the number you've entered between 10 and 20 False
```

Istanbul Technical University

Control Statement (if-then-else)

- Syntax :

```
if exp:
        code block
else:
        code block
```

- Note the indent!!!

Ahmed Al Hilli
Al-Furat Al-Awsat Technical University

```
# program to test if statement

num = int(input('Enter a number:'))

if (num > 10):
    print('the number is bigger than 10!')
else:
    print('the number is smaller or equal to 10!!')

print('The End')
```

Note the indent

```
>>> (executing file "if_statement.py")
Enter a number:2
the number is smaller or equal to 10!!
The End
```

Awsat Technical

How to check if the number is 10 or less?

- We can use the following:

```
if (exp1):
        code block
elif (exp2):
        code block
else:
        code block
```

Ahmed Al Hilli
Al-Furat Al-Awsat Technical University

```python
# program to test if statement

num = int(input('Enter a number:'))

if (num > 10):
    print('the number is bigger than 10!')
elif num == 10:
    print('the number is equal to 10!!')
else:
    print('the number is less than 10')

print('The End')
```

```
>>> (executing file "if_elif_else_statement.py")
Enter a number:10
the number is equal to 10!!
The End
```

Dr. Ahmed Al Hilli Technical

Back to variables: String

- String: set of characters (i.e., text)
- You can refer to individual character by using ‘str[index]’, where ‘index’ is an integer that refers to the position of the character
- Indexing the last character, set ‘index’ to -1
- Slicing: get set of characters in order
  ex: str1[0:10] will get all characters from index 0 to index 9

```
>>> str1 = 'Welcome to Introduction to Python'

>>> str1[0]
'W'

>>> str1[-1]
'n'

>>> str1[0:10]
'Welcome to'
```

Index in python start from 0

Slicing will not include the last index

Ahmed Al-Hilli
Al-Furat Al-Awsat Technical University

Some useful functions for strings

```
my_str = 'The quick brown fox jumps over the lazy dog'

str_length = len(my_str) # get the number of characters inside the string
o_number = my_str.count('o')    # count the occurance of the letter o
str2 = my_str.capitalize()  # capatalize the first letter in the string
str3 = my_str.upper()   # change all character to upper letter
str4 = my_str.lower()   # change all character to lower case

print(str_length)
print(o_number)
print(str2)
print(str3)
print(str4)
```

```
>>> (executing file "str_useful_functions.py")
43
4
The quick brown fox jumps over the lazy dog
THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
the quick brown fox jumps over the lazy dog
```

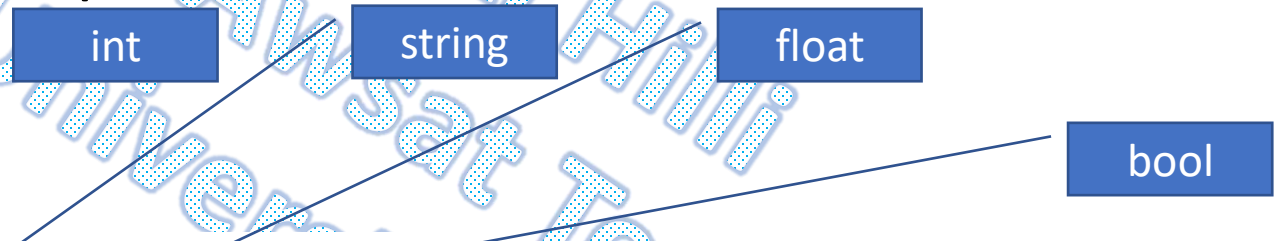
```python
# some useful functions for string

my_str = 'The quick brown fox jumps over the lazy dog'

str1 = my_str.replace('o','4')  # replace specific character with another one
str_all = my_str.split()    # split the string into set of strings accoriding to a specific character
                            # while space by default

print(str1)
print(str_all)


print(type(str1))

print(type(str_all))
```

```
>>> (executing file "str_useful_functions_2.py")
The quick br4wn f4x jumps 4ver the lazy d4g
['The', 'quick', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog']
<class 'str'>
<class 'list'>
```

The type of str_all is list???

Lists

- Lists in python is a sequence of objects
- Objects can have different types
- You can define lists with square brackets




```
>>> my_list = [1,'Hello',3.2,True]

>>> my_list
[1, 'Hello', 3.2, True]
```

Al-Furat Al-Awsat Technical University
Ahmed Al Hilli

- You can use indexing to return specific item in lists (like strings)

```
>>> my_list[0]
1

>>> my_list[3]
True
```

- Slicing also works on lists

```
>>> my_list[0:2]
[1, 'Hello']
```

Ahmed Al Hilli
Al-Furat Al-Awsat Technical University

- The most important function in lists is ‘append()’, which adds an item to the end of the list

```
>>> my_list
[1, 'Hello', 3.2, True]

>>> my_list.append(11223)

>>> my_list
[1, 'Hello', 3.2, True, 11223]
```

The appended item

Al-Furat Al-Awsat Technical University
Ahmed Al Hussain

Thank you for listening

Questions?

Al-Furat Al-Awsat Technical University
Ahmed Al Hilli

Day 3

Loops and list comprehension in Python

Ahmed Al Hilly
Al-Furat Al-Awsat Technical University

For loop

Al-Furat Al-Awsat Technical University
Ahmed Al Hilli

- Loops are used when we want to repeat the same action multiple times
- Syntax of ‘for’ loop:
  for index in sequence:
      block of code

- Sequence can be list, string, or **range**

- Range: is a special type that used mainly with for loops.
- The syntax:
  for index in range(10):
      block of code

```
>>> for i in range(10):
...     print(i)
...
...
0
1
2
3
4
5
6
7
8
9
```

Not including 10

Index i starts from 0

Al-Furat Al Abrine University

Al-Furat Al-Awsat Technical University
Ahmed Al Hilli

- More detailed syntax for range function:
  range(start, end, inc)

  start: starting index
  end: ending index (no included)
  inc: the increment

```
>>> for i in range(2,10,2):
...     print(i)
...
...
2
4
6
8
```

```
>>> for i in range(10,0,-2):
...     print(i)
...
...
10
8
6
4
2
```

- EX: write a python code that accepts an integer from the user, and find the sum of all integers that is less than or equal to the entered number?
- Sol:
  - First, our code should accept a number form user. (use input function)
  - Check if the number is greater than zero
  - Create a variable (int) to hold the final result
  - Use for loop to sum the number (Note that this is only one way to solve the problem. There are many other better approaches to solve this kind of problems)

Ahmed Al Hilli
Al-Furat Al-Awsat Technical University

```
# This code prints the sum of all natural number that are less than or equal to
num = int(input('Please enter a positive number:'))

if num<=0:
    print('The number you\'ve entere                sitive!!')
    quit()

my_sum = 0
for i in range(num+1):
    my_sum = my_sum + i

print('The sum of all integers between 1 and '+str(num)+' is '+str(my_sum))
```

Why num+1??

Casting int to str

```
>>> (executing file "sum_of_int.py")
Please enter a positive number:5
The sum of all integers between 0 and 5 is 15
```

While loop

- For loops are useful for repeating an action a **specific** number of times

- But, what if you don’t know exactly that specific number?
- For example, if you want to read lines from a text file, but you don’t know the number of lines in that file?
- In these situations, while loops are useful

Ahmed Al Hilli
Al-Furat Al-Awsat Technical University

- While loop syntax:
  while(condition):
      code

- Condition is a Boolean statement. If condition is true, the code will get executed, otherwise not

Ahmed Al Hilli
Al-Furat Al-Awsat Technical University

```python
# This code prints the sum of all natural number that are less than or equal to specific number

num = int(input('Please enter a positive number:'))

cond = True
my_sum = 0
my_index = 0
while (cond):
    if num<=0:
        print('you\'ve entered a negative number!!!')
        break

    my_index = my_index + 1
    cond = my_index < num
    my_sum = my_sum + my_index


print('The sum is: '+str(my_sum))
```

ersity Technical

```
>>> (executing file "sum_of_int_while_loop.py")
Please enter a positive number:5
The sum is: 15
```

Some useful controlling statements with loops

- Break: Used to exit a loop
- Continue: used to skip the current iteration, and go to the next iteration

Al-Furat Al-Awsat Technical University
Ahmed Al Hilli

```python
# This code prints the sum of all even natural number
# that are less than or equal to specific number

num = int(input('Please enter a positive number:'))

if num<=0:
    print('yoour number is negative!!')
    quit()

my_sum = 0
for i in range(num+1):
    if i/2 != int(i/2):
        continue

    my_sum = my_sum + i

print('The sum is: '+str(my_sum))
```

```
>>> (executing file "sum_of_even_int.py")
Please enter a positive number:5
The sum is: 6
```

Hilli Technical

List Comprehension

Al-Furat Al-Awsat Technical University
Ahmed Al Hilli

- List Comprehension: an elegant way to create lists in Python
- Syntax:
  [expression for item in list if condition]

Examples

```
>>> my_list = [x for x in range(100)]

>>> my_list
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25,
26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49,
50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73,
74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97,
98, 99]
```

Al-Furat Ahliyya University Technical

```
>>> my_list = [x**2+3*x+5 for x in range(100)]

>>> my_list
[5, 9, 15, 23, 33, 45, 59, 75, 93, 113, 135, 159, 185, 213, 243, 275, 309, 345, 383, 423, 465,
509, 555, 603, 653, 705, 759, 815, 873, 933, 995, 1059, 1125, 1193, 1263, 1335, 1409, 1485, 156
3, 1643, 1725, 1809, 1895, 1983, 2073, 2165, 2259, 2355, 2453, 2553, 2655, 2759, 2865, 2973, 30
83, 3195, 3309, 3425, 3543, 3663, 3785, 3909, 4035, 4163, 4293, 4425, 4559, 4695, 4833, 4973, 5
115, 5259, 5405, 5553, 5703, 5855, 6009, 6165, 6323, 6483, 6645, 6809, 6975, 7143, 7313, 7485,
7659, 7835, 8013, 8193, 8375, 8559, 8745, 8933, 9123, 9315, 9509, 9705, 9903, 10103]
```

Al Hilli
Awsat Technical
University

```
>>> my_list = [x**2+3*x+5 for x in range(100) if x/2 == int(x/2)]

>>> my_list
[5, 15, 33, 59, 93, 135, 185, 243, 309, 383, 465, 555, 653, 759, 873, 995, 1125, 1263, 1409, 15
63, 1725, 1895, 2073, 2259, 2453, 2655, 2865, 3083, 3309, 3543, 3785, 4035, 4293, 4559, 4833, 5
115, 5405, 5703, 6009, 6323, 6645, 6975, 7313, 7659, 8013, 8375, 8745, 9123, 9509, 9903]
```

Al Hilli
Al-Awsat Technical
University

To loop or not to loop

- Which one is better? Loops or list comprehension
- List comprehension is usually faster than loops
- Memory requirement for loops is usually less than list comprehension requirement
- Use list comprehensive if you have enough memory, otherwise use loops

Ahmed Al Hilli
Al-Furat Al-Awsat Technical University

```python
# This code prints the sum of all even natural number
# that are less than or equal to specific number

import datetime



num = 5000000
if num<=0:
    print('yoour number is negative!!')
    quit()

t0 = datetime.datetime.now()
my_sum = 0
for i in range(num+1):
    if i/2 != int(i/2):
        continue

    my_sum = my_sum + i

t_exe_for = datetime.datetime.now() - t0
print('The sum is: '+str(my_sum))
print('The execution time using loops is: '+str(t_exe_for))

# Using list comprehension
t0 = datetime.datetime.now()
lst = sum([x for x in range(num+1) if x/2==int(x/2)])
t_exe_list = datetime.datetime.now() - t0

print('The sum is: '+str(my_sum))
print('The execution time using list comprehension is: '+str(t_exe_list))
```

```
>>> (executing file "sum_of_even_int_loop_vs_lists.py")
The sum is: 6250002500000
The execution time using loops is: 0:00:01.683084
The sum is: 6250002500000
The execution time using list comprehension is: 0:00:01.274595
```

ical

Thank you for listening

Questions?

Al-Furat Al-Awsat Technical University
Ahmed Al Hilli