INTRODUCTION TO PYTHON

DENNIS TROLLE & ANDERS NIELSEN

AIM OF THIS COURSE

This is an online course, which you may follow in your own pace, and seeks to:

- Introduce the basic concepts of the Python language
- Provide you with enough knowledge of Python to use it as an alternative to other data-processing alternatives – e.g. Excel
- Through examples get hands-on experience with Python, and lay a foundation that enables you to conduct further data analysis

LEARNING GOALS

After the course, the participant is expected to be able to:

- Prepare and import data into Python (e.g. pandas dataframes).
- Conduct basic processing of data.
- Sort, slice and select in data for analysis.
- Apply basic data visualizations.
- Apply basic statistical analysis.
- Export data to record format (e.g. text or excel).

WHAT IS PYTHON?

- Python is a high-level general-purpose programming language
- It is a scripting language (it is interpreted – as opposed to a compiled language like Fortran), and Python is similar to f.x. perl and matlab
- It was originally created by Guido van Rossum (first released in 1991)
- Python is open source and community driven
- Python is platform independent (runs on Windows, Mac, Linux)
- Python comes with “batteries included” - a standard distribution includes many different libraries for analysis, plotting etc.

WHY USE PYTHON?

- As a scientist, consultant or manager, you deal with data/observations and you often need a tool to help with data manipulation, analysis and plotting
  – **Python is a powerful, open source (and free) option**

- Basically, you use Python to **“get the job done”**

- Python is “easy to learn”, as it was developed with emphasis on code readability, and therefore used at many universities when introducing programming

- Good documentation – massive online support

- It is very versatile and include f.x. many common data manipulations, scientific computing and machine learning, statistics, many plotting features, web frameworks, data base connectivity, interfaces to GIS, text and image processing

WORKING WITH PYTHON

Can be done:

- Interactively:
  - Through a **terminal** (Python shell)
  - Through an **IDE** (Integrated Development Environment), f.x.:
    - Spyder
    - IPython, Jupyter
    - + many more

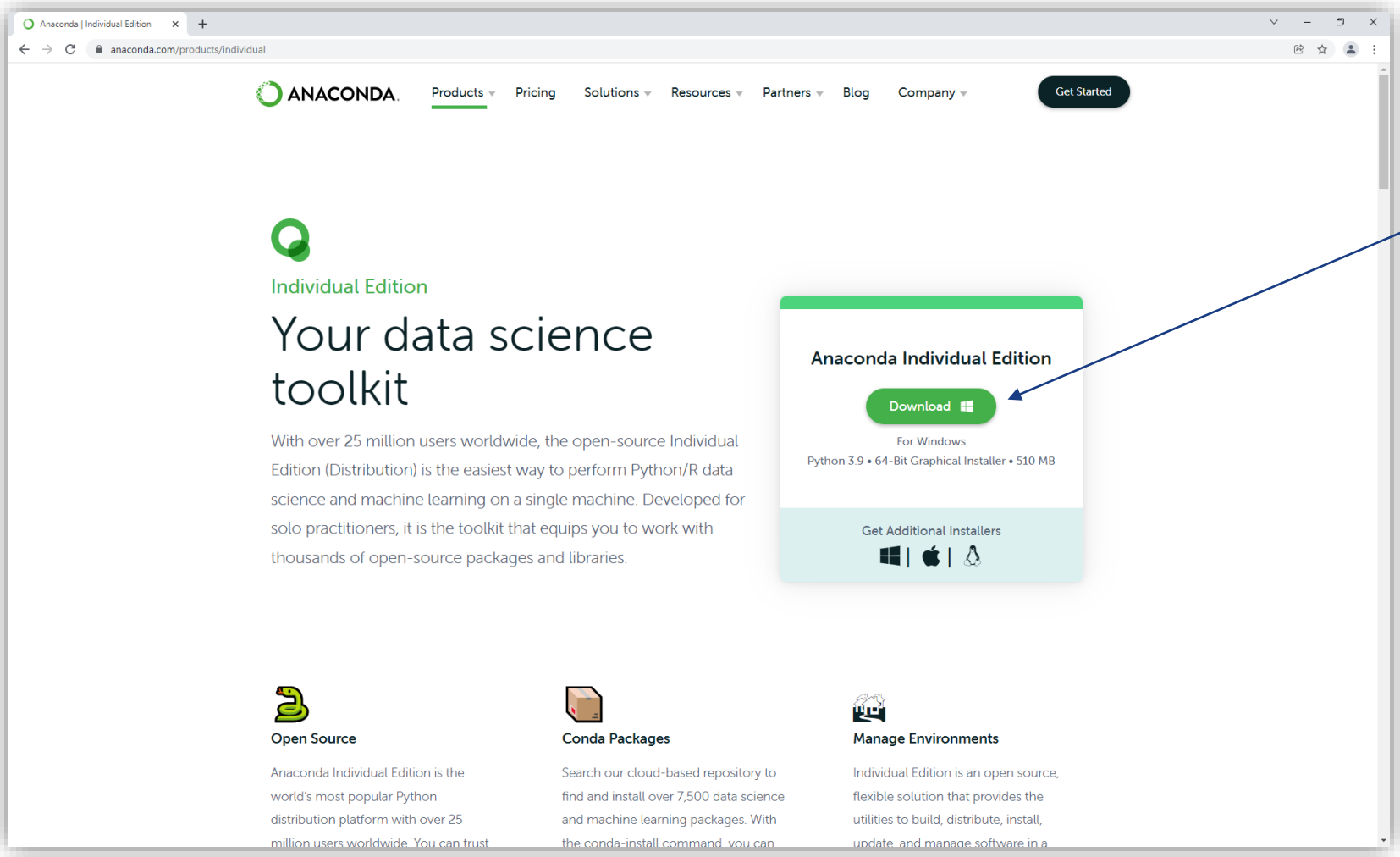
- As scripts (*.py):
  - Run from a **terminal**
  - Developed and run via an **IDE** with a GUI (Graphical User Interfase)

- **In this course**, we will use the "**Spyder IDE**", which comes with the Anaconda distribution (with Python 3)
- Spyder is an open source cross-platform IDE for Python

CODE SAMPLES FROM SPYDER

```
x = 34 - 23       # A comment.
y = "Hello"       # Another one.
z = 3.45
if z == 3.45 or y == "Hello":
    x = x + 1
    y = y + " World"   # String concatenate.
print(x)
print(y)
```

INSTALLATION

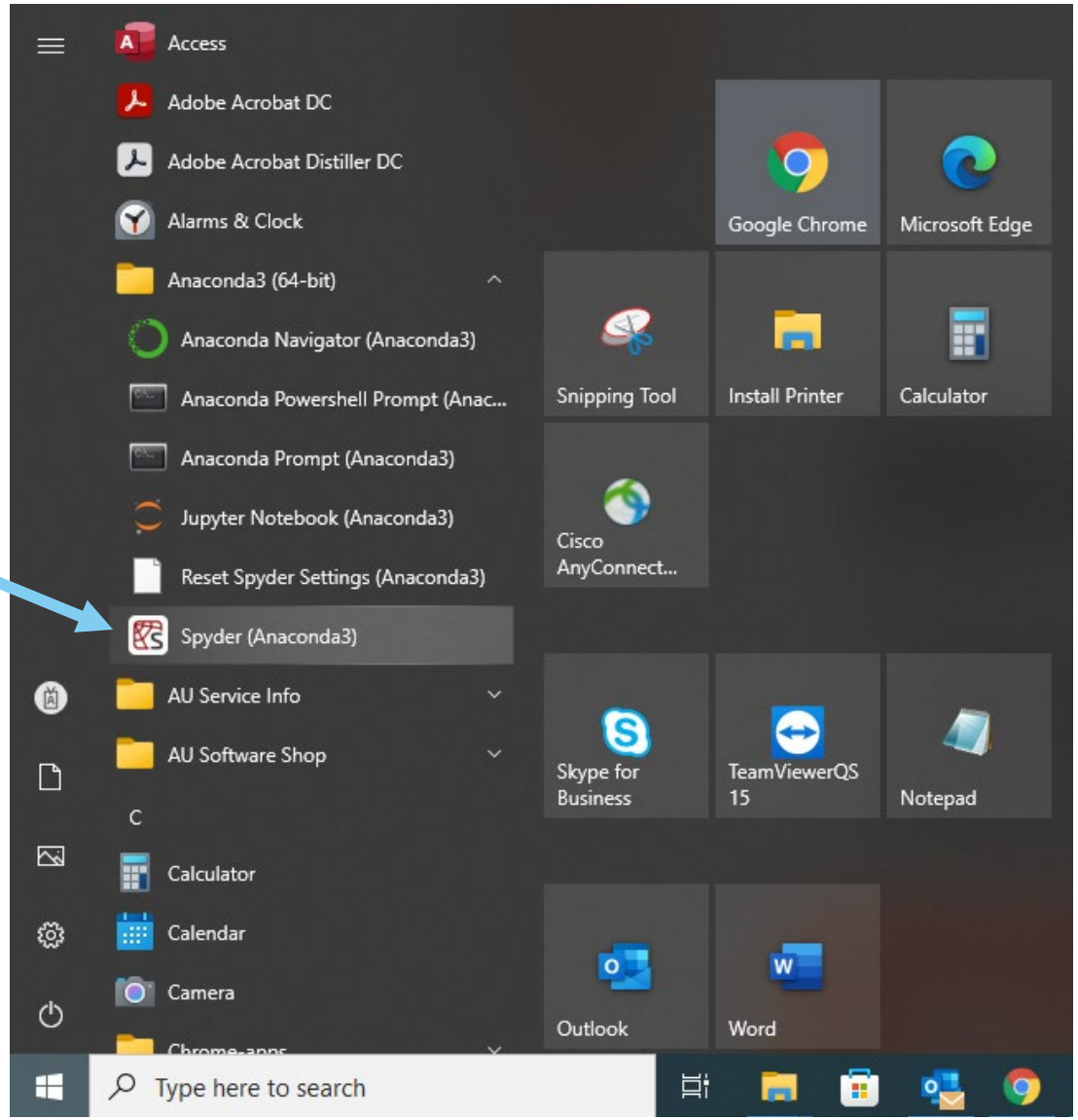


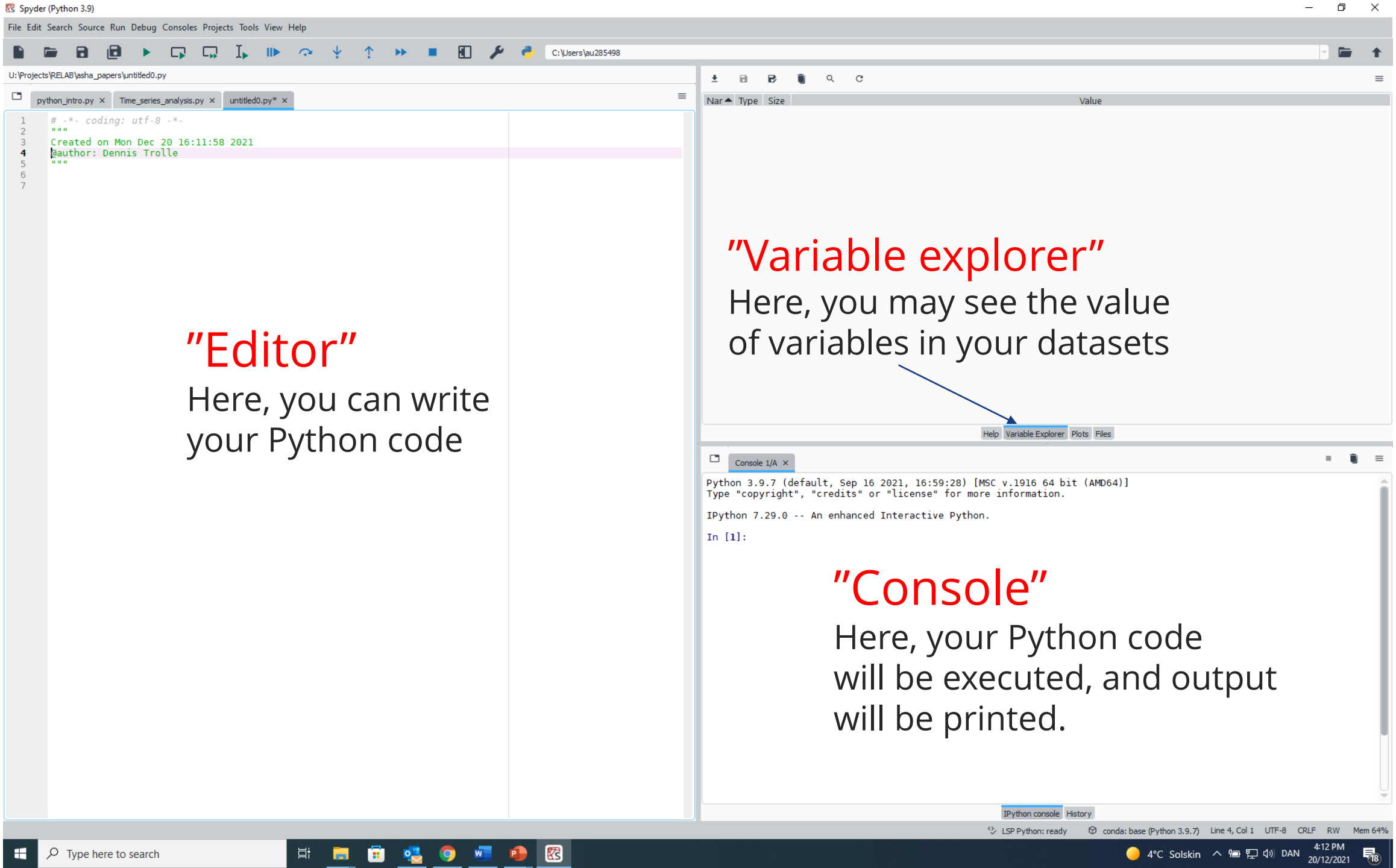
Download and install for your chosen platform (e.g. Windows)

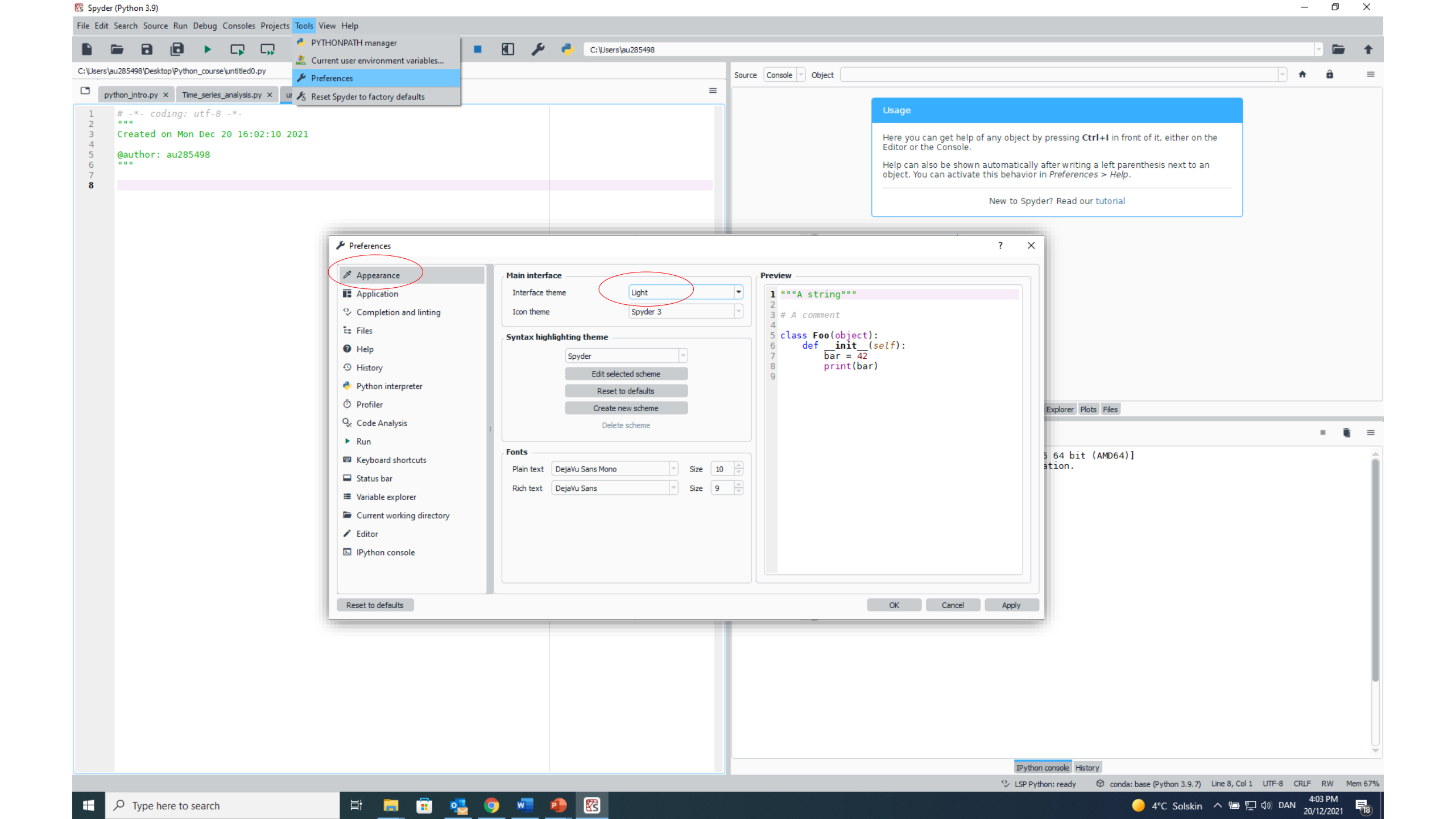
Use free version – no need to sign up

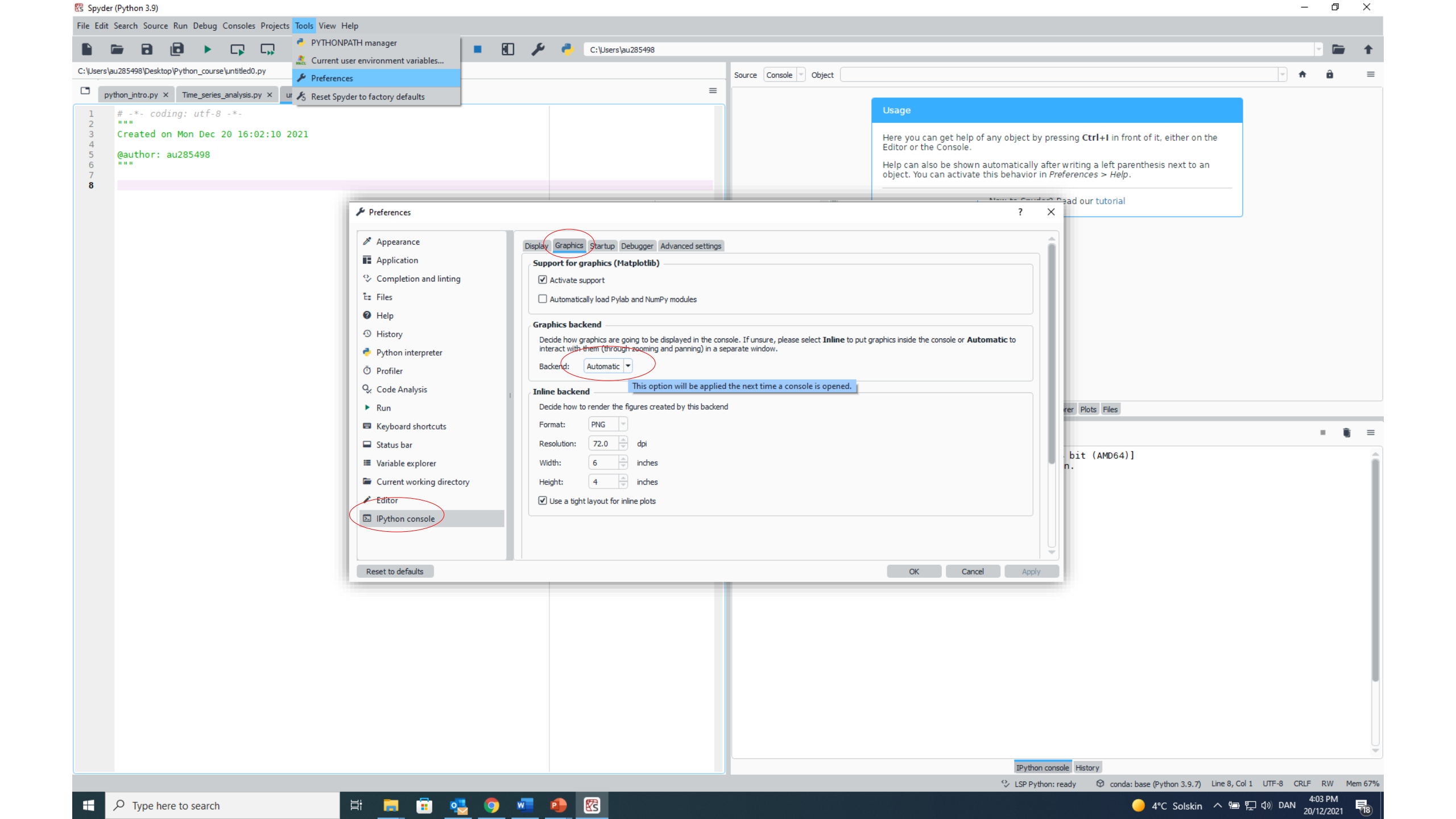
https://www.anaconda.com/products/individual

LAUNCH SPYDER WHEN INSTALLED




Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\au285498
U:\Projects\RELAB\asha_papers\untitled0.py
python_intro.py
Time_series_analysis.py
untitled0.py*
-*- coding: utf-8 -*-
"""
Created on Mon Dec 20 16:11:58 2021
@author: Dennis Trolle
"""
"Editor"
Here, you can write your Python code
Nar
Type
Size
Value
"Variable explorer"
Here, you may see the value of variables in your datasets
Help
Variable Explorer
Plots
Files
Console 1/A
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.29.0 -- An enhanced Interactive Python.
In [1]:
"Console"
Here, your Python code will be executed, and output will be printed.
IPython console
History
LSP Python: ready
conda: base (Python 3.9.7)
Line 4, Col 1
UTF-8
CRLF
RW
Mem 64%
Type here to search
4°C Solskin
DAN
4:12 PM
20/12/2021


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
PYTHONPATH manager
Current user environment variables...
Preferences
Reset Spyder to factory defaults
C:\Users\au285498
C:\Users\au285498\Desktop\Python_course\untitled0.py
python_intro.py
Time_series_analysis.py
-*- coding: utf-8 -*-
"""
Created on Mon Dec 20 16:02:10 2021
@author: au285498
"""
Source Console Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Preferences
Appearance
Application
Completion and linting
Files
Help
History
Python interpreter
Profiler
Code Analysis
Run
Keyboard shortcuts
Status bar
Variable explorer
Current working directory
Editor
IPython console
Main interface
Interface theme
Light
Icon theme
Spyder 3
Syntax highlighting theme
Spyder
Edit selected scheme
Reset to defaults
Create new scheme
Delete scheme
Fonts
Plain text
DejaVu Sans Mono
Size
10
Rich text
DejaVu Sans
Size
9
Preview
"""A string"""
# A comment
class Foo(object):
def __init__(self):
bar = 42
print(bar)
Reset to defaults
OK
Cancel
Apply
Explorer Plots Files
IPython console History
LSP Python: ready
conda: base (Python 3.9.7)
Line 8, Col 1
UTF-8
CRLF
RW
Mem 67%
Type here to search
4°C Solskin
DAN
4:03 PM
20/12/2021


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
PYTHONPATH manager
Current user environment variables...
Preferences
Reset Spyder to factory defaults
C:\Users\au285498
C:\Users\au285498\Desktop\Python_course\untitled0.py
python_intro.py
Time_series_analysis.py
-*- coding: utf-8 -*-
"""
Created on Mon Dec 20 16:02:10 2021
@author: au285498
"""
Source Console Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
Read our tutorial
Preferences
Appearance
Application
Completion and linting
Files
Help
History
Python interpreter
Profiler
Code Analysis
Run
Keyboard shortcuts
Status bar
Variable explorer
Current working directory
Editor
IPython console
Display
Graphics
Startup
Debugger
Advanced settings
Support for graphics (Matplotlib)
Activate support
Automatically load Pylab and NumPy modules
Graphics backend
Decide how graphics are going to be displayed in the console. If unsure, please select Inline to put graphics inside the console or Automatic to interact with them (through zooming and panning) in a separate window.
Backend: Automatic
This option will be applied the next time a console is opened.
Inline backend
Decide how to render the figures created by this backend
Format: PNG
Resolution: 72.0 dpi
Width: 6 inches
Height: 4 inches
Use a tight layout for inline plots
Reset to defaults
OK
Cancel
Apply
Plots
Files
IPython console
History
LSP Python: ready
conda: base (Python 3.9.7)
Line 8, Col 1
UTF-8
CRLF
RW
Mem 67%
Type here to search
4°C Solskin
DAN
4:03 PM
20/12/2021

Spyder (Python 3.9)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\au285498

U:\Projects\RELAB\asha_papers\untitled0.py

python_intro.py × Time_series_analysis.py × untitled0.py* ×

```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Mon Dec 20 16:11:58 2021
4  @author: Dennis Trolle
5  """
6
7  print("I love Python")
```

Try to write:
print(”I love Python”)

Then highlight/select the line and click 

Note:
Python is ”case-sensitive” and knows the difference between small and capital letters
(remember to write ”print” with small letters)

Nar | Type | Size | Value

Help | Variable Explorer | Plots | Files

Console 1/A ×

```
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.29.0 -- An enhanced Interactive Python.

In [1]: print('I love Python')
I love Python

In [2]:
```

IPython console | History

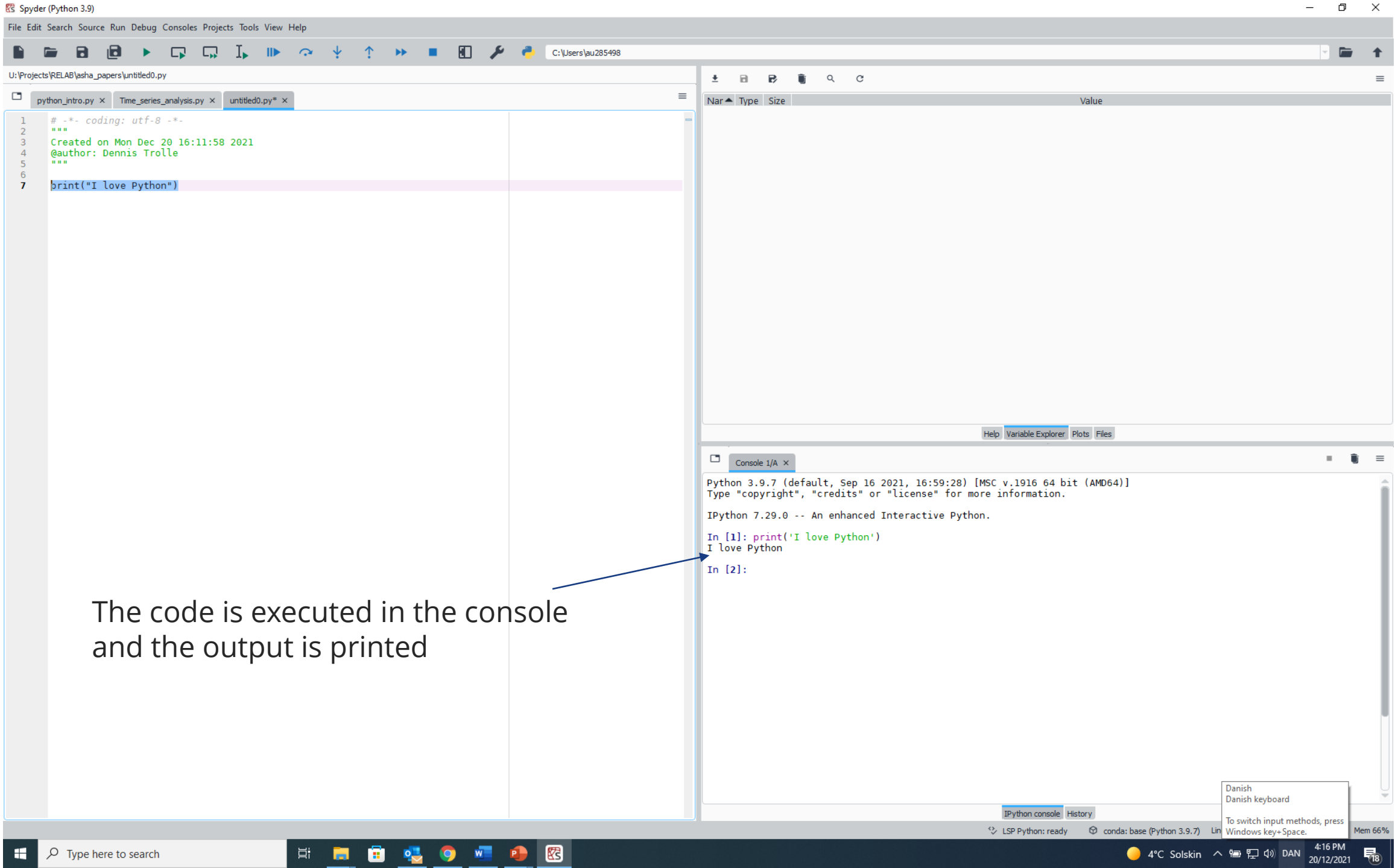
LSP Python: ready | conda: base (Python 3.9.7) | Mem 66%

Danish
Danish keyboard

To switch input methods, press Windows key+Space.

Type here to search

4°C Solskin | DAN | 4:16 PM 20/12/2021



The code is executed in the console and the output is printed

UNDERSTANDING PYTHON CODE

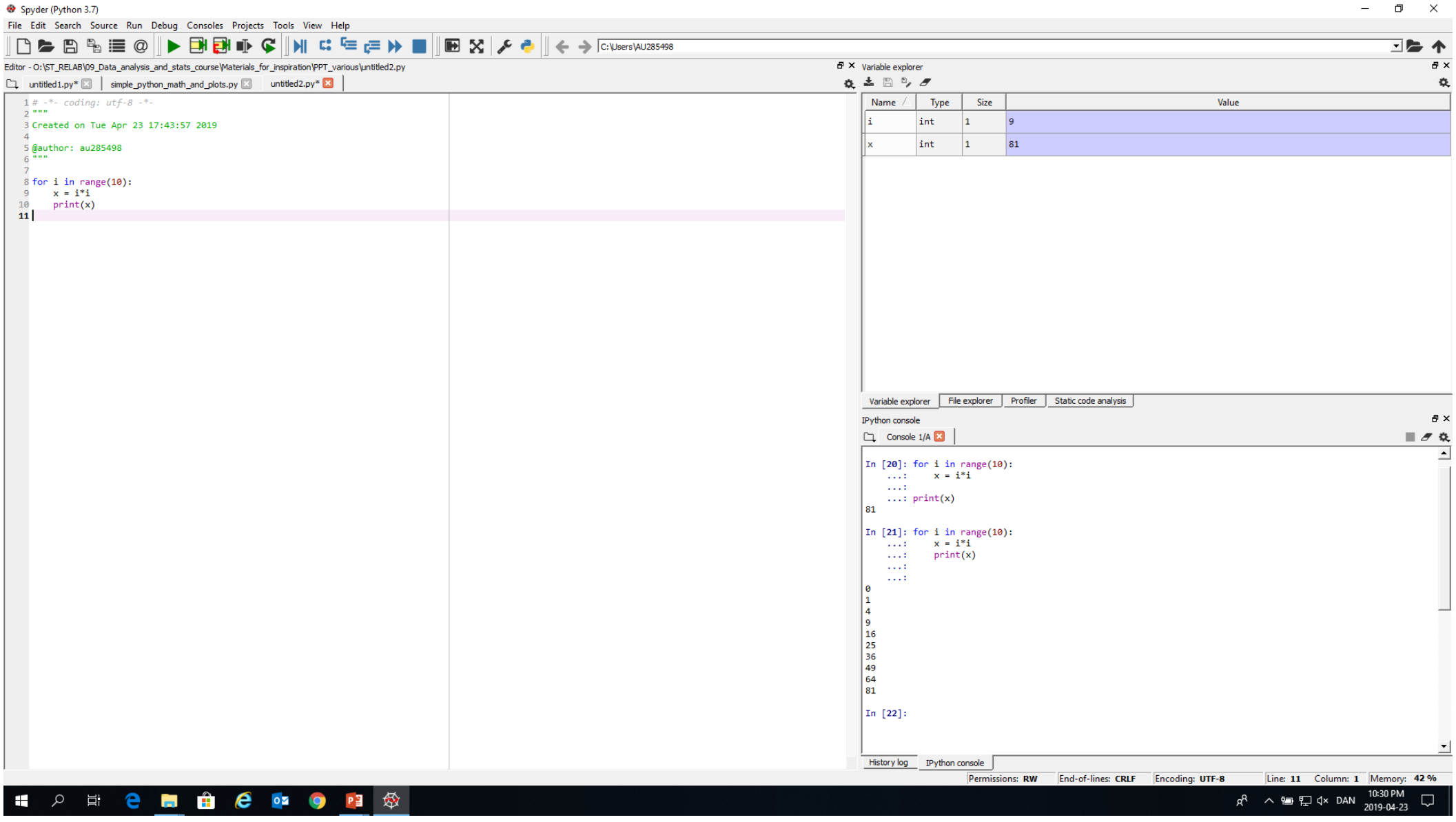
DENNIS TROLLE & ANDERS NIELSEN

1 SLIDE ON PYTHON CODE BASICS

- Indentation matters to the meaning of the code
  - Block structure indicated by indentation
- The first assignment to a variable creates it
  - Python is a dynamically-typed language: no type declarations are needed. *Names* do not have types, *objects* do, and Python will figure out the type when a variable is assigned a value the first time
- Assignment uses `=` and comparison uses `==`
- For numbers `+ - * / %` are as expected
  - Use of **`+`** for string concatenation
  - Use of **`%`** for string formatting
- Logical operators are words (`and`, `or`, `not`), *not* symbols (&&, ||, !)
- The basic printing command is `print()`

INDENTATION MATTERS

TRY IT...



BASIC DATA TYPES

Integers (default for numbers):

```
z = 5 / 2  # Answer = 2 if z is integer (integer division)
```

Floats:

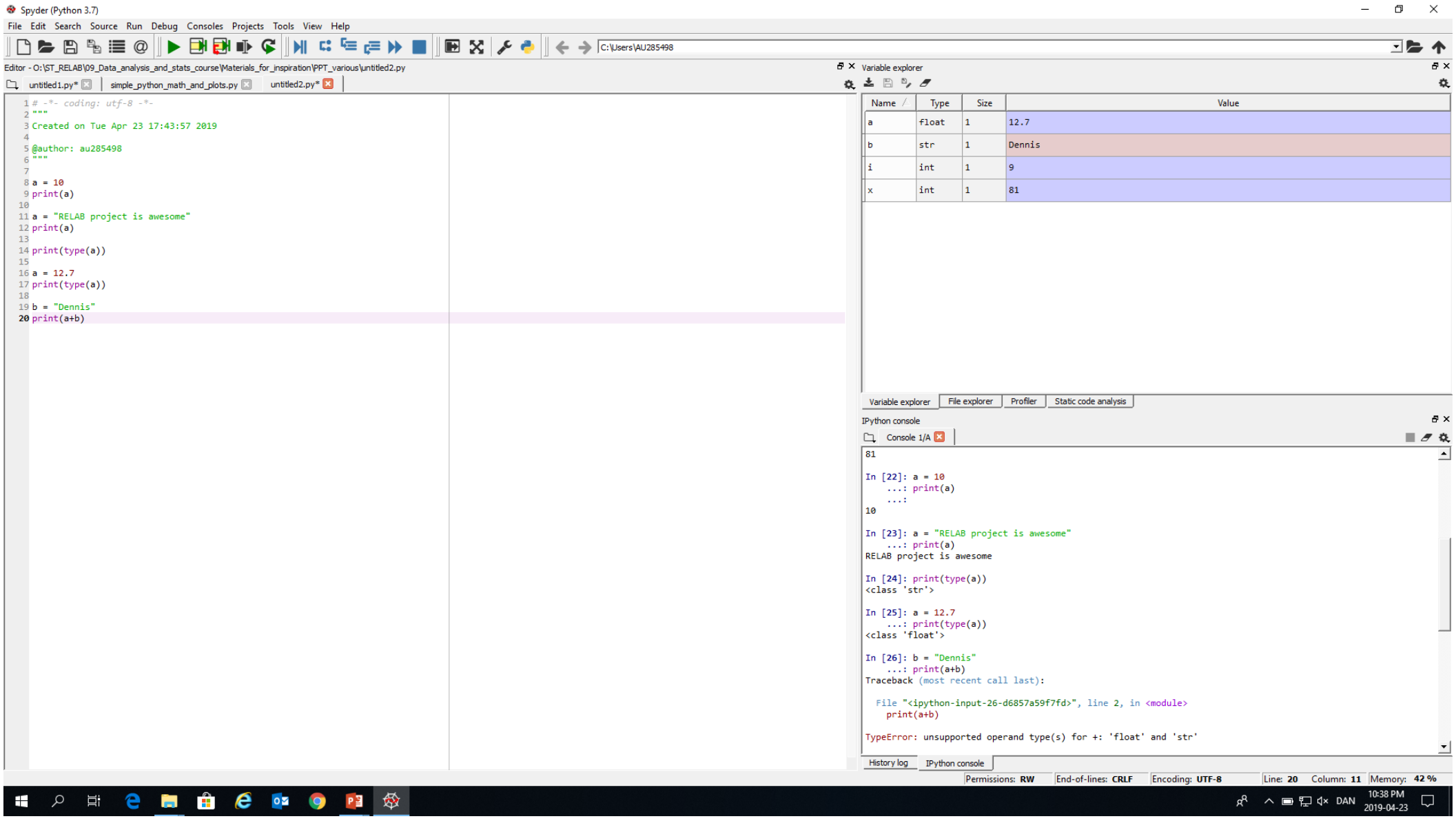
```
x = 3.456
```

Strings:

- Can use "..." or '...' to specify, "foo" == 'foo'
- Unmatched can occur within the string "John's" or 'John said "foo!".'
- Use triple double-quotes for multi-line strings or strings than contain both ' and " inside of them: """a'b"c"""

DATA TYPES

TRY IT...



PYTHON DATA CONTAINERS
(DATA STRUCTURES)

Lists (mutable sets of strings)

- `var = [] # create list using square brackets`
- `var = ['one', 2, 'three', 'banana']`

Tuples (immutable sets)

- `var = ('one', 2, 'three', 'banana')`

Dictionaries (associative arrays or 'hashes')

- Dictionaries are sets of key & value pairs
- `var = {} # create dictionary using use curly brackets`
- `var = {'lat': 40.20547, 'lon': -74.76322}`
- `var['lat'] = 40.2054`

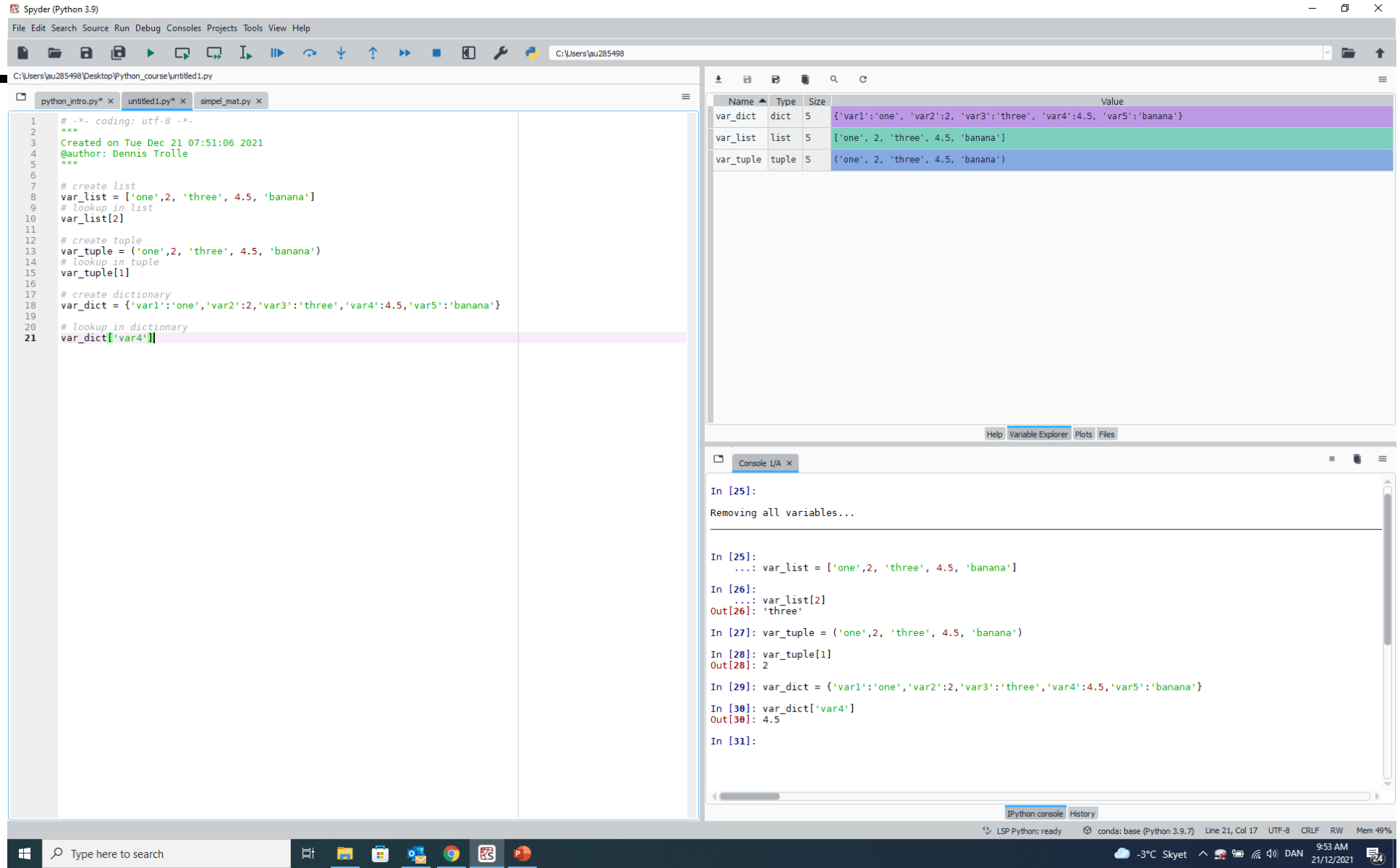
Pandas DataFrame (two-dimensional arrays – like a spreadsheet with column and row labels)

- makes manipulating data easy, from selecting or replacing columns and indices to reshaping your data
- `df = pandas.DataFrame(columns=['X', 'Y', 'Z'])`

Each data structure has its own set of methods

DATA CONTAINERS

TRY IT...



PYTHON CONTROL FLOW STATEMENTS

A program's control flow is the order in which the program's code executes. The control flow of a Python program is regulated by conditional statements, loops, and function calls.

https://docs.python.org/3/tutorial/controlflow

EXAMPLE OF CONTROL FLOW STATEMENTS

Conditionals

- if name == “Kurt Cobain”:
  <do_something>
- else:
  <do_something_else>

Loops

- for item in list:
  <do_something>
- while n < 1000:
  <do_something>

Example of if statement

```
a = 250
b = 38
if b > a:
   print("b is greater than a")
elif a == b:
   print("a and b are equal")
else:
   print("a is greater than b")
```

PYTHON FUNCTIONS

- Good for code re-use

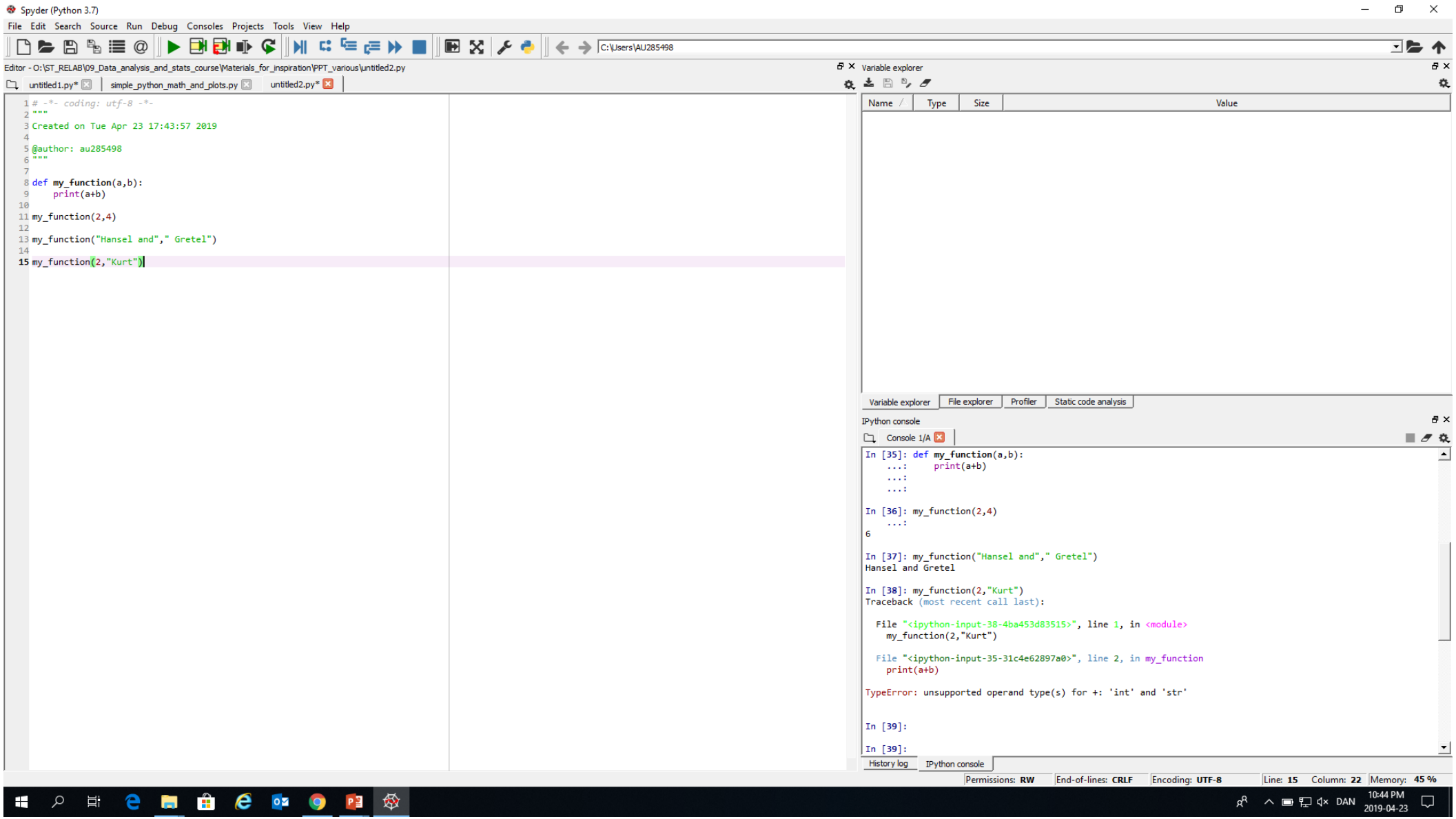
- General structure:

```
def <name>(parameters):
    <block_of_python_code>
```

- Note: indentation of code block
- Parameters can be empty – but you must include ()
- Remember the :

FUNCTIONS

TRY IT...



LOGICAL EXPRESSIONS

Many logical expressions use *relational operators*:

Operator	Meaning	Example	Result
==	equals	`1 + 1 == 2`	`True`
!=	does not equal	`3.2 != 2.5`	`True`
<	less than	`10 < 5`	`False`
>	greater than	`10 > 5`	`True`
<=	less than or equal to	`126 <= 100`	`False`
>=	greater than or equal to	`5.0 >= 5.0`	`True`

Logical expressions can be combined with *logical operators:*

Operator	Example	Result
`and`	`9 != 6 and 2 < 3`	`True`
`or`	`2 == 3 or -1 < 5`	`True`
`not`	`not 7 > 0`	`False`

MODULES AND LIBRARIES

MODULES, PACKAGES AND LIBRARIES

- **Modules** are additional pieces of code that further extend Python's functionality
- A module typically has a specific function, f.x.:
  - additional math functions, plot, databases, network…
- Python comes with many useful modules
- A Python **package** is a collection of modules. A Python **library** is a collection of packages.
- Python libraries, such as SciPy, can therefore include multiple packages and modules. Oftentimes, developers create Python libraries to share reusable code with the community.

Modules are accessed using import

- import pandas # imports pandas
- Import pandas as pd # imports pandas as pd
- Subsets of modules can be imported, e.g. import pandas.plot

Modules are then addressed by modulename.function()

- pd.read_excel(filename)

PYTHON LIBRARIES FOR DATA SCIENCE

NumPy:

- introduces objects for multidimensional arrays and matrices, as well as functions that allow to easily perform advanced mathematical and statistical operations on those objects

- provides vectorization of mathematical operations on arrays and matrices which significantly improves the performance

- many other python libraries are built on NumPy

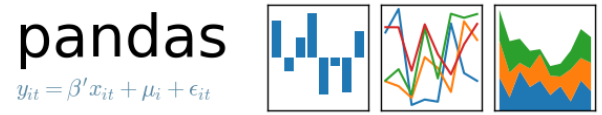
Link: http://www.numpy.org/

PYTHON LIBRARIES FOR DATA SCIENCE

SciPy:

- collection of algorithms for linear algebra, differential equations, numerical integration, optimization, statistics and more
- part of SciPy Stack
- built on NumPy

Link: https://www.scipy.org/scipylib/



PYTHON LIBRARIES FOR DATA SCIENCE

Pandas:

- adds data structures and tools designed to work with table-like data (similar to Series and Data Frames in R)
- provides tools for data manipulation: reshaping, merging, sorting, slicing, aggregation etc.
- allows handling missing data

Link: http://pandas.pydata.org/

PYTHON LIBRARIES FOR DATA SCIENCE

matplotlib:

- python 2D plotting library which produces publication quality figures in a variety of hardcopy formats
- a set of functionalities similar to those of MATLAB
- line plots, scatter plots, barcharts, histograms, pie charts etc.
- relatively low-level; some effort needed to create advanced visualization

Link: https://matplotlib.org/

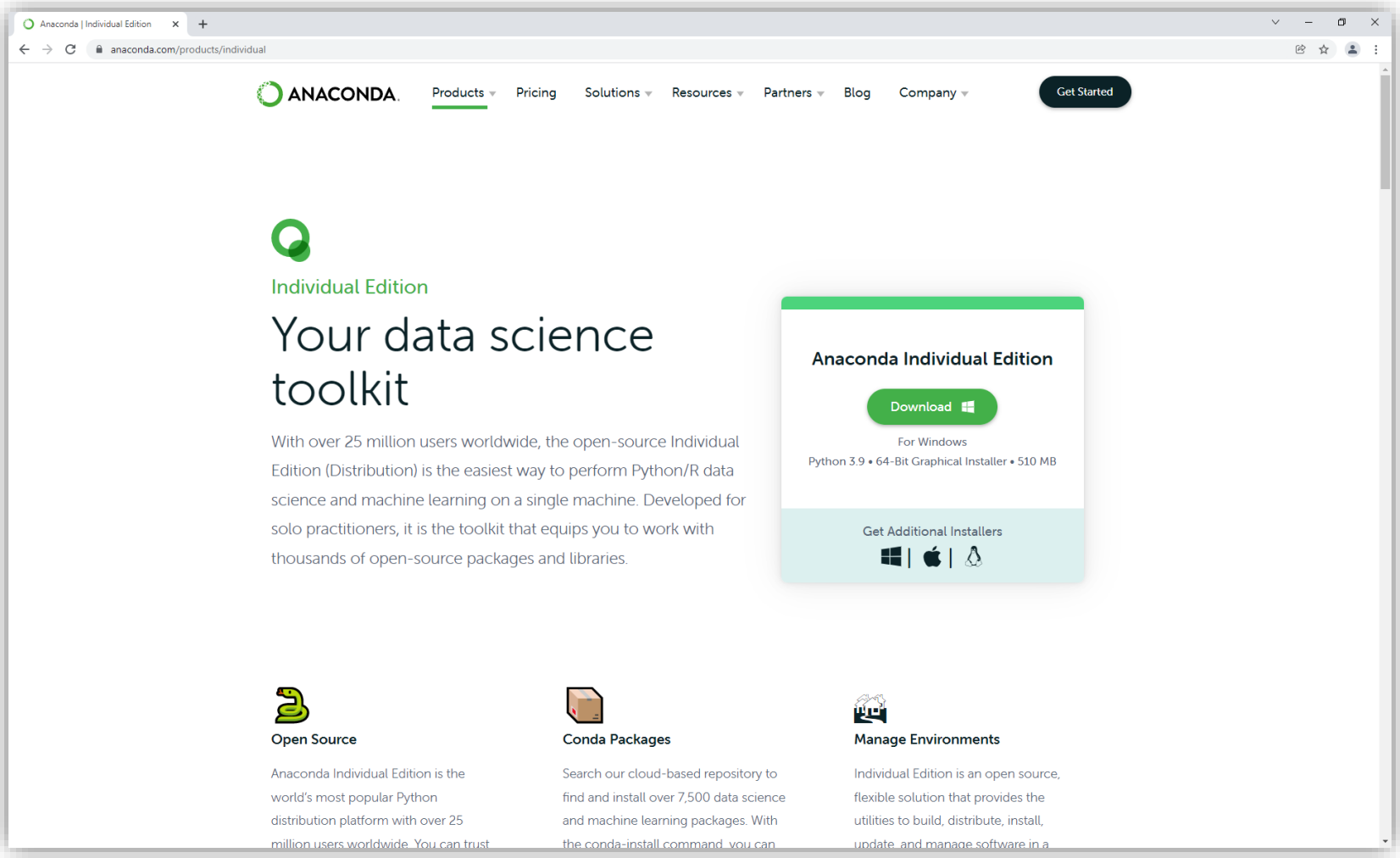
PYTHON LIBRARIES FOR DATA SCIENCE

Seaborn:

- based on matplotlib
- provides high level interface for drawing attractive statistical graphics
- Similar (in style) to the popular ggplot2 library in R

Link: https://seaborn.pydata.org/

THE ANACONDA INSTALLATION INCLUDES THESE LIBRARIES BY DEFAULT



https://www.anaconda.com/products/individual