

# Programming with Python

## Course Introduction

# MATH26021 Programming with Python

- This is a 10 credit course which introduces algorithms and basic programming concepts using Python as a language
- The course consists of **online lectures, review sessions, and weekly lab classes**
- It is estimated that you will need **at least**
  - **4 hours per week** for self-study of online lectures,
  - **plus further 4 hours per week** for solving exercises!
- **Plan in sufficient self-study time for this course.** This is not an easy course!

And there are no January exams, so all learning will happen during the semester

# General information

- Taught by Professor Stefan Güttel
- All materials and announcements on <http://tiny.cc/mancpy> (linked from BB)
- The teaching times are as follows:

Monday morning      self-study online materials

Monday afternoon      1-hour review session (live coding)

Rest of the week      solve exercises, attend a 2-hour lab class

- You only need to **attend one of the 2-hour online lab classes** each week

Please check your personal timetable

# Self-study materials

These are provided on the course web site <http://tiny.cc/mancpy>

Check out the interactive video lectures marked as [VIDEO]\*

There is an embedded repl.it code editor on the right-hand side

The screenshot shows a web browser window with the address bar displaying `personalpages.manchester.ac.uk/staff/stefan.guettel/py/01-video.php`. The page content includes a video player on the left with a table of contents, a main text area, and a code editor on the right.

**Table of Contents:**

- 00:00 Programming with Python
- 04:05 Contents:
- 04:38 An algorithm
- 07:19 A program
- 09:24 Types of programming languages
- 11:42 Python versions
- 12:37 An example: Hello world
- 17:37 Comments
- 22:30 Input

**Main Text:**

numbers?"

### Converting basic types

We can explicitly tell Python to convert a string to an integer or a real number, and vice versa.

In [14]:

```
x = int(input())
y = float(input())
print("x = ", x)
print("y = ", y)
print("x+y = ", x+y)
z = 'This is a string: ' + str(x+y) + ''
print(z)
```

17  
1.23  
x = 17  
y = 1.23  
x+y = 18.23  
This is a string: "18.23"

We see three conversion functions:

- `int()`, which takes a string and converts it to an

**Code Editor:**

The code editor shows a file named `main.py` with the following code:

```
1 print("hello world")
2
```

Below the code editor is a terminal window with the URL `https://youteach.guettel.repl.run`.

\* If the video doesn't play because you have no access to YouTube, try the alternative link marked as [alt]

# Feedback on code

- If you are new to programming, you will make a lot of mistakes (bugs) in the beginning and you will require feedback on your code writing
- For obvious reasons (viruses, uniformity of help, etc.) **we will NOT give feedback on code via email**
- However, the lab classes provide a great opportunity for getting feedback. There will be many helpers for this purpose, so please make good use of this!
- Further support will be provided in the Blackboard forum
- **Feedback on code will be given in the lab classes and the Blackboard forum, and only there!**

# Blackboard forum

An easy way to discuss exercises and other questions relevant to the course

Forum: MATH26021 Python Forum  
*Forums are made up of individual discussion threads that can be organised around a particular subject. A thread is a conversation within a forum that includes the initial post and all replies to it. When you access a forum, a list of threads appears. [More Help](#)*

Create Thread   Subscribe   Search   Display ▾   Tags ▾

<input type="checkbox"/>	DATE ▾	THREAD	AUTHOR	STATUS	TAGS	UNREAD POSTS	UNREAD REPLIES TO ME	TOTAL POSTS
<input type="checkbox"/>	24/09/20 09:23	<a href="#">How to use this forum</a>	 <b>Stefan Guettel</b>	Published		0	0	1

Thread Actions ▾   Collect   Delete

Thread Actions ▾   Collect   Delete

Displaying 1 to 1 of 1 items   Edit Paging...

← OK

- Before creating a new thread, please check whether someone else has already asked your question

# Blackboard forum: creating a thread and posting code

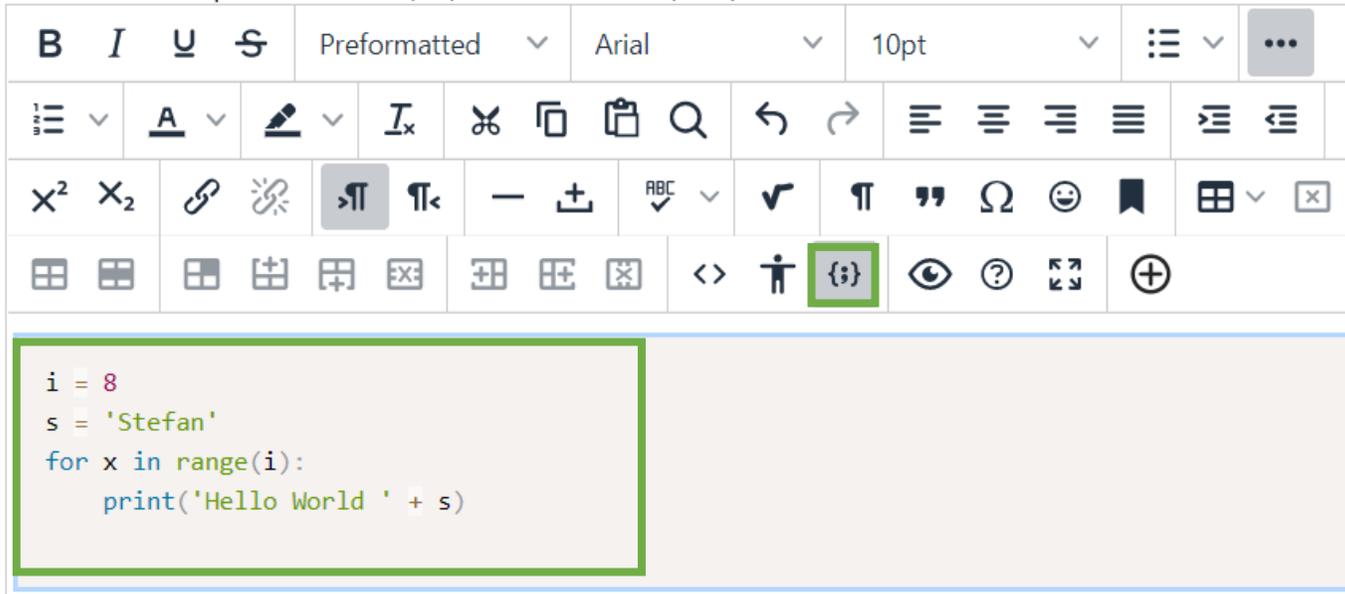
Blackboard forum has a special code environment. **When creating a thread or posting code, please:**

\* Subject

Hello world program: How to improve it?

Message

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).



The screenshot shows the Blackboard forum editor interface. At the top, there is a subject field containing the text "Hello world program: How to improve it?". Below the subject field is a message field. Above the message field is a rich text toolbar with various formatting options like bold, italic, underline, strikethrough, text color, background color, font size, font family, and text alignment. Below the toolbar is a code editor area where a Python code sample is pasted and highlighted with a green box. The code is as follows:

```
i = 8
s = 'Stefan'
for x in range(i):
    print('Hello World ' + s)
```

Generally:

Always choose a meaningful subject (e.g. the number of the exercise)

When posting code:

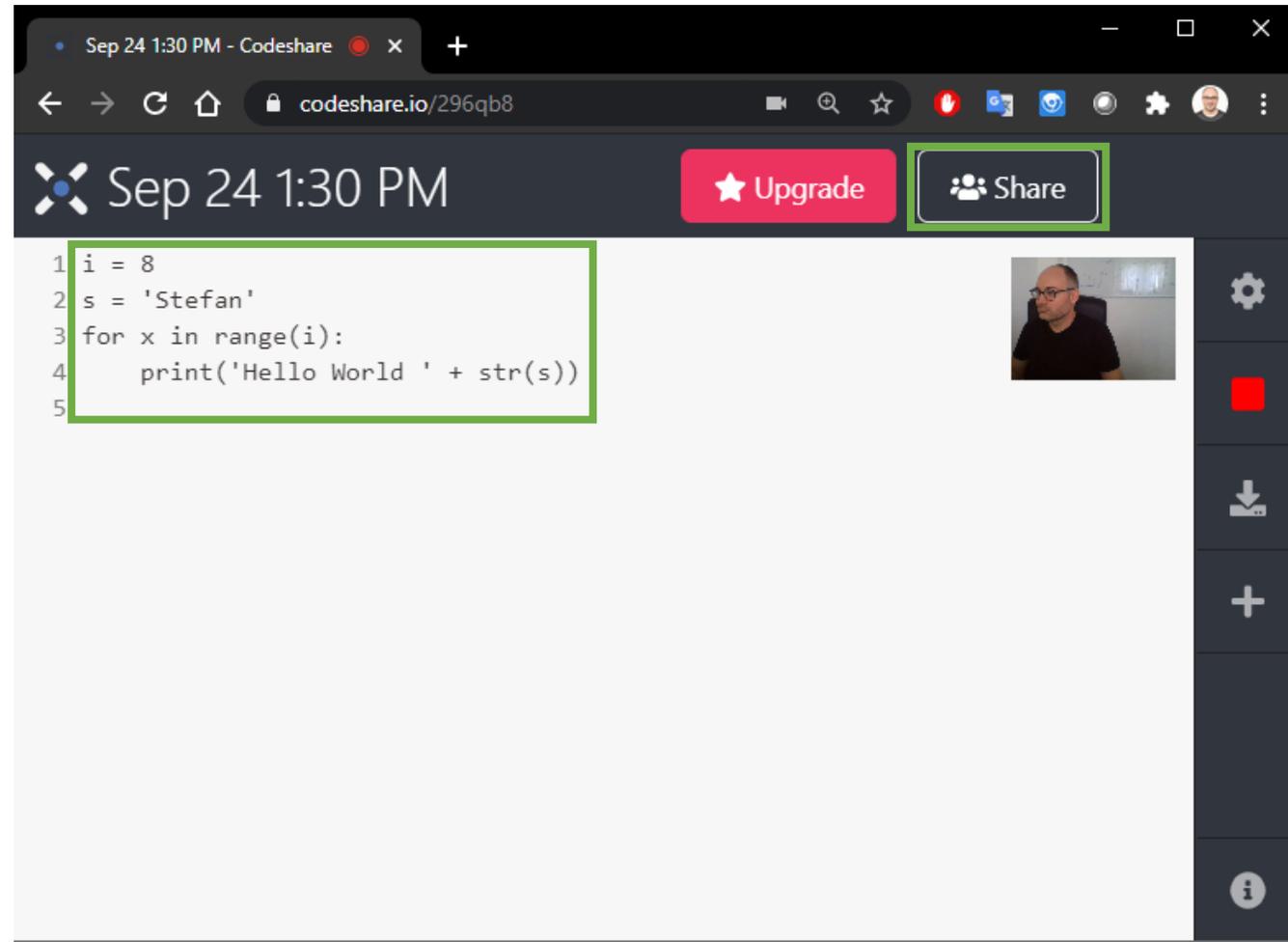
1. Click [Insert/Edit code sample] button
2. Select language 'Python'
3. Paste your code.
4. Click on [Save]

# An alternative way to share code: codeshare.io

If you want to quickly share Python code for an online discussion:

- go to <https://codeshare.io>
- optionally, create an account (recommended)
- click on “Share Code Now”
- enter your Python code into the text field (or copy-and-paste from Spyder)
- click [Share] and post the link on the Blackboard forum or send to discussion partners

Note: one can even do video chats on codeshare, but unfortunately not run the code



The screenshot shows a web browser window with the URL `codeshare.io/296qb8`. The page title is "Sep 24 1:30 PM". The interface includes a navigation bar with a "Share" button highlighted by a green box. Below the navigation bar is a code editor with the following Python code:

```
1 i = 8
2 s = 'Stefan'
3 for x in range(i):
4     print('Hello World ' + str(s))
5
```

The code editor is also highlighted by a green box. To the right of the code editor is a video chat window showing a person's face. The right sidebar contains icons for settings, a red stop button, a download icon, a plus sign, and an information icon.

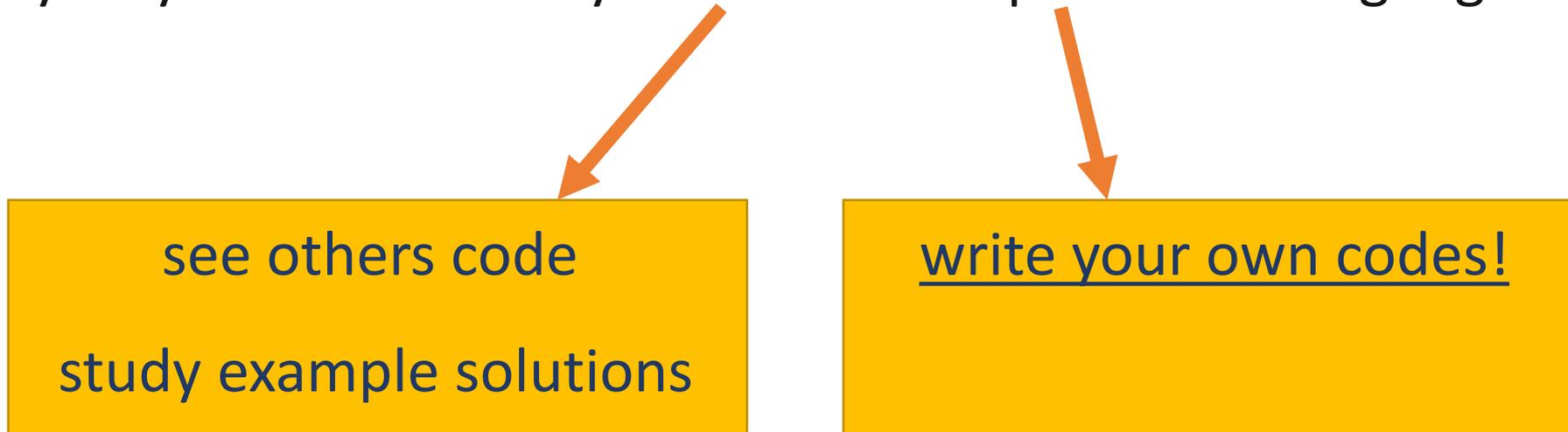
# Assessment

- There will be two mini tests, each counting 15% of the overall mark
- There will be a final coursework project, counting for 70% of the overall mark
- Every week there will be exercises to be completed before and during the lab classes
- **Exercises are very important to actually learn Python and to prepare for the assessments. Make sure to attempt them!**

# So how does one learn how to code?

Think of it like learning a new language

The only way to learn is to try “listen” and “speak” the language



In order to write *Python programs* on our computer, we need the *Spyder IDE* from the *Anaconda distribution*...

# What is Python?

Python is a computer **programming language**. This means, it is a language that you can use to give instructions to your computer.

You can think of your computer as a good friend who follows your instructions, and Python would be the language that you communicate in with him or her.

There are various ways of communication (just as in the “real world”: phone, email, chatting, etc.), some of which are more interactive than others (chatting is more interactive than writing a letter, for example).

Similarly, there are various ways to use the Python language.

# A bit of terminology

**Python:** One (of many other) **programming languages** we will be using. It is the language we will write computer programs in.

**IPython:** A Python **interpreter**. A computer application that provides a convenient and interactive mode for executing Python commands and programs.

**Jupyter:** A **web application** that allows to run IPython in the browser.

**Spyder:** An **integrated development environment (IDE)**. A computer application that includes IPython, a text editor for writing and debugging programs, and more.

**JupyterLab:** An **IDE evolution** of Jupyter notebook.

**PyPlot:** A **module** that provides visualization tools.

**NumPy:** A standard **library** (collection of modules, data types, etc.) that provides numerical arrays and mathematical functions.

**Anaconda:** A Python **distribution**. A single download that conveniently packages all of the above and installs it on your computer.

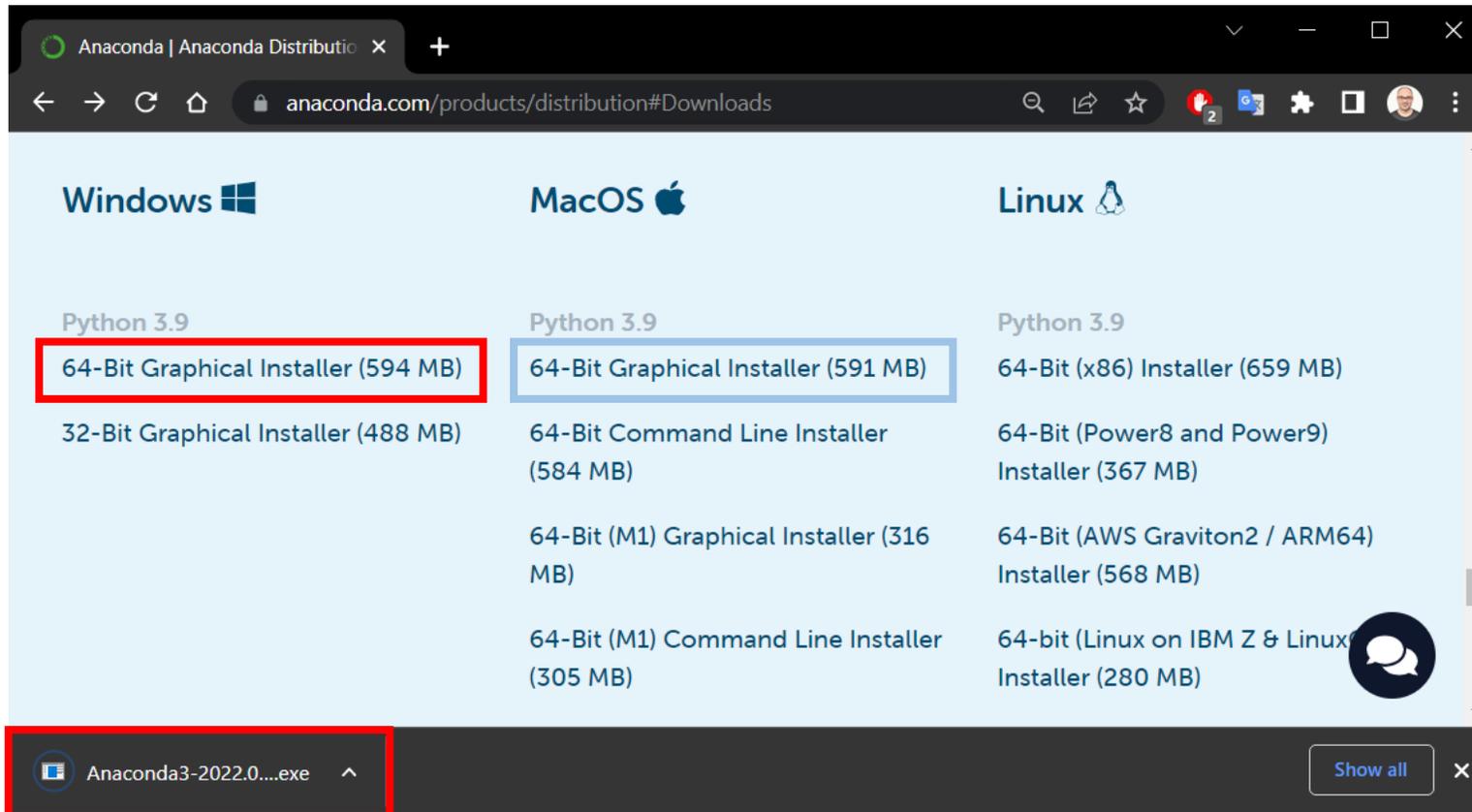
# Install Anaconda

Go to <https://www.anaconda.com/products/distribution#Downloads>

Scroll down, click on “64-Bit Graphical Installer” to download

Double-click on .exe file to start installation

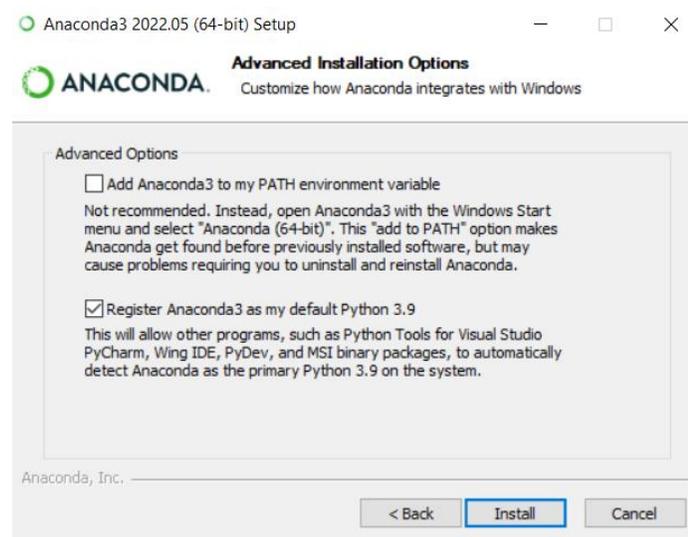
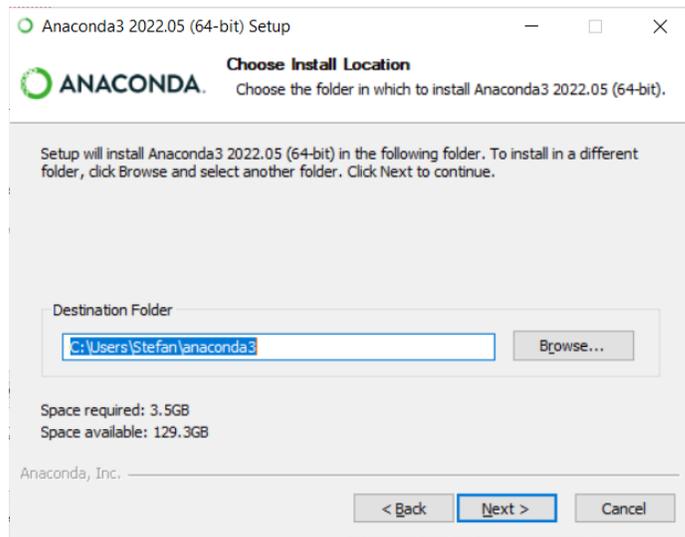
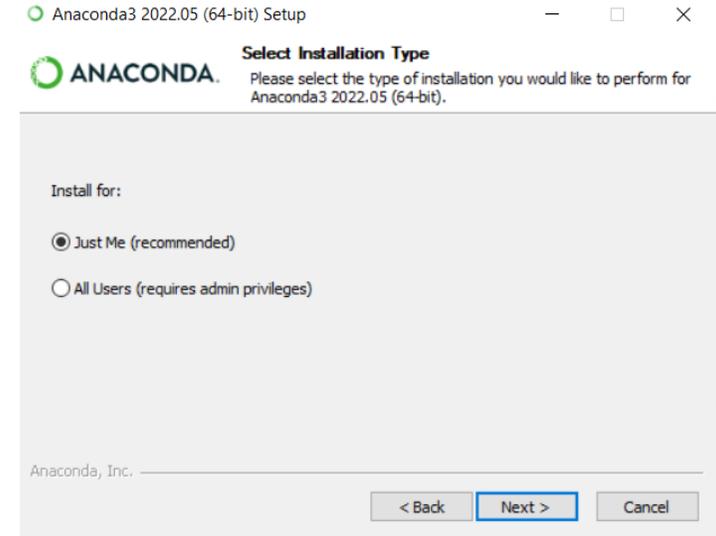
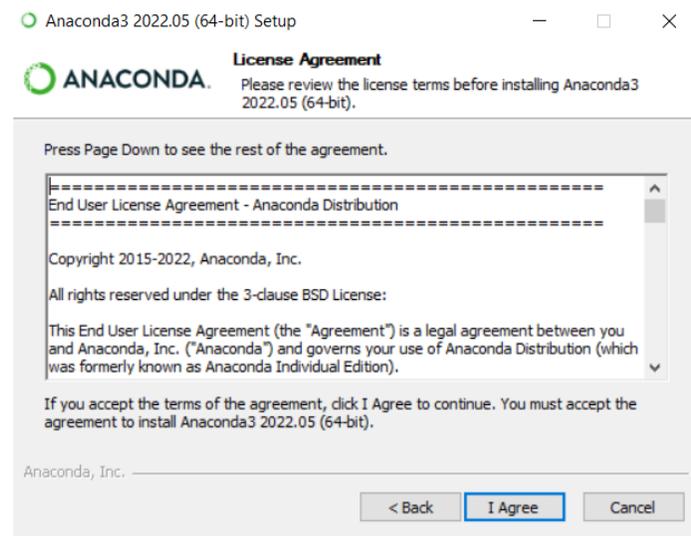
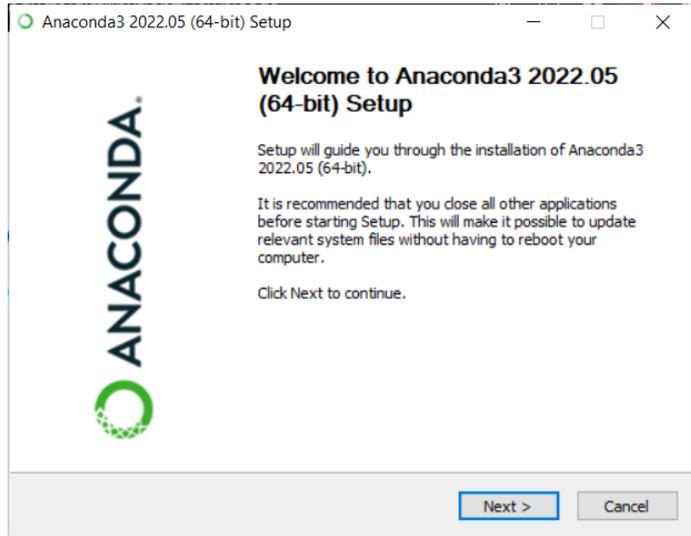
(no need to complete the contact form on the following page)



Installation example shown for a Windows machine  
For Mac, download and install .pkg file instead

# Install Anaconda ctd.

Click through the following pages, leave all the recommended settings as they are.



# Fix the Anaconda installation

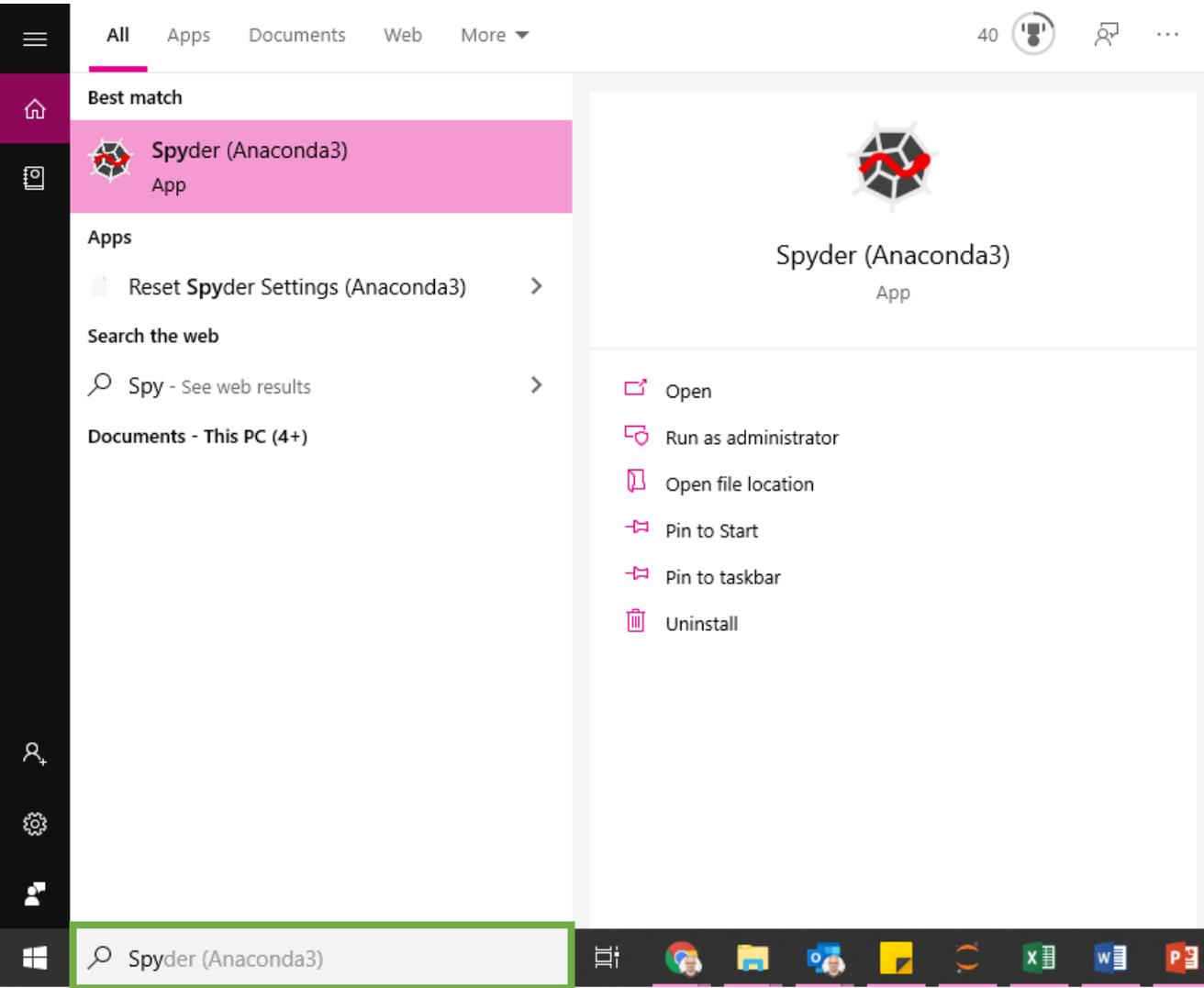
**Unfortunately, Anaconda version 2022.05 comes with a buggy Spyder IDE (v5.1.5)**

[Issue: using Python's `input()` function will just crash Spyder!]

1. To fix this and update Spyder, first make sure Spyder is closed
2. Open the `Anaconda Prompt` by using in the Windows search bar
3. A Command Prompt will open
4. Type `conda update anaconda` – hit [Enter] and WAIT...
5. Type `y` when asked to Proceed, then wait again (can take > 30 minutes)
6. Type `conda remove spyder` – hit [Enter], wait and type `y` to Proceed
7. Type `conda install spyder` – hit [Enter], wait and type `y` to Proceed
8. Once completed, close the console and restart Spyder

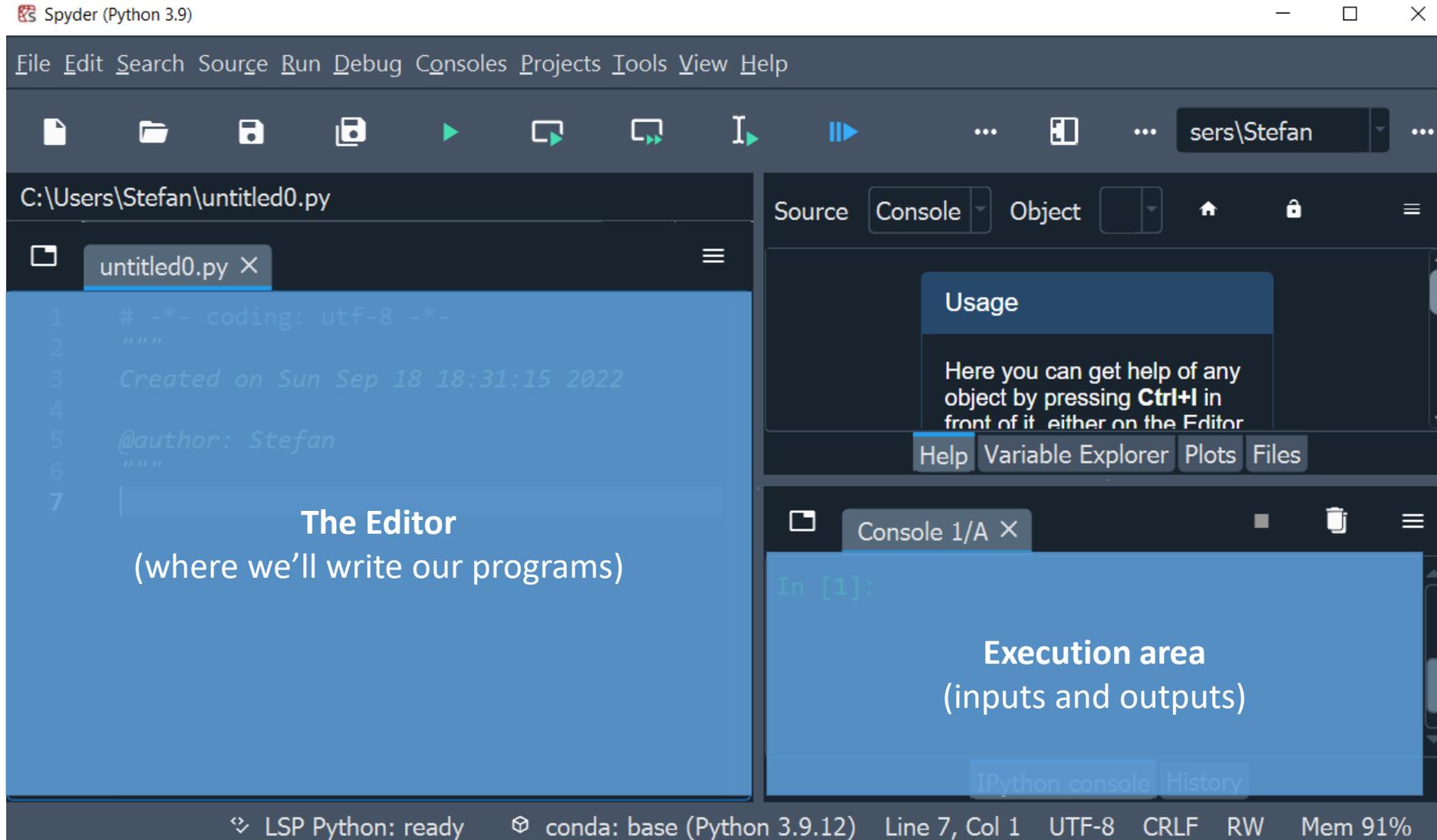
# Start Spyder

After the installation is completed, **type “Spyder”** in the Windows search bar (you might have to accept Firewall access)



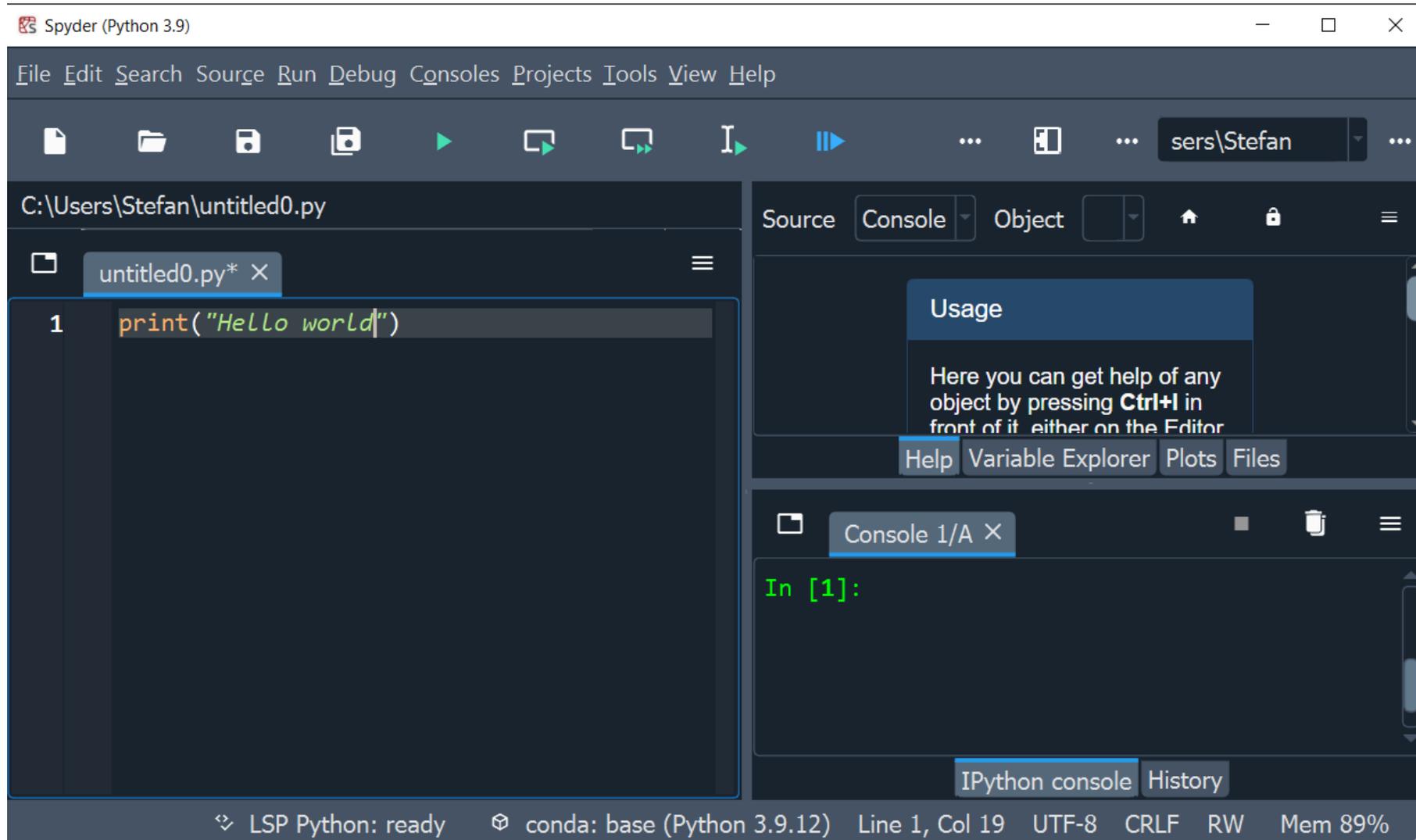
# Start Spyder

You should see a window similar to the one below. Familiarise yourself with the layout.



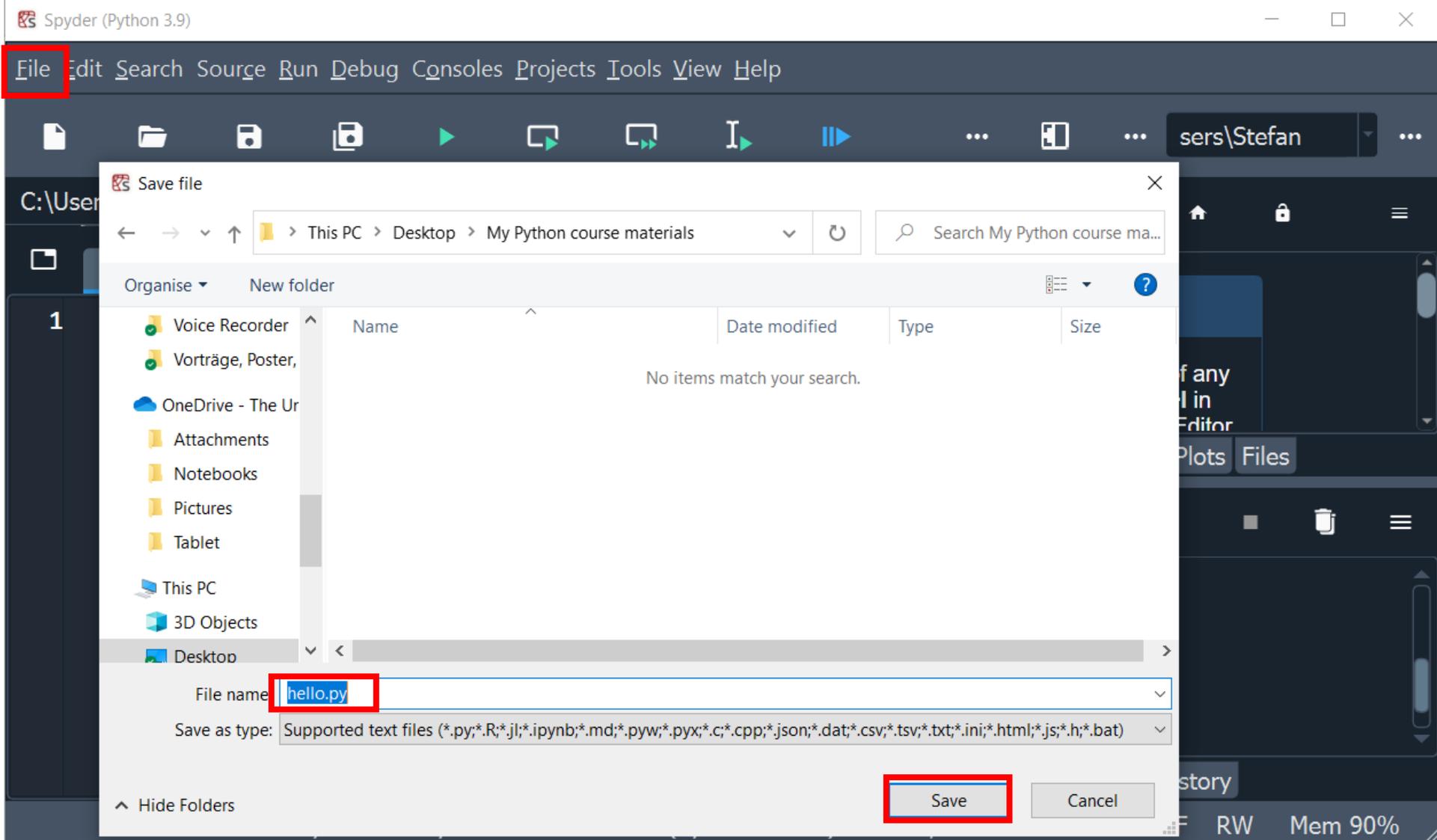
# Your first Python program

Delete the stuff in the editor and type `print("Hello world")`



# Your first Python program ctd.

Go to File and Save as... - create a new folder and save the file as `hello.py`

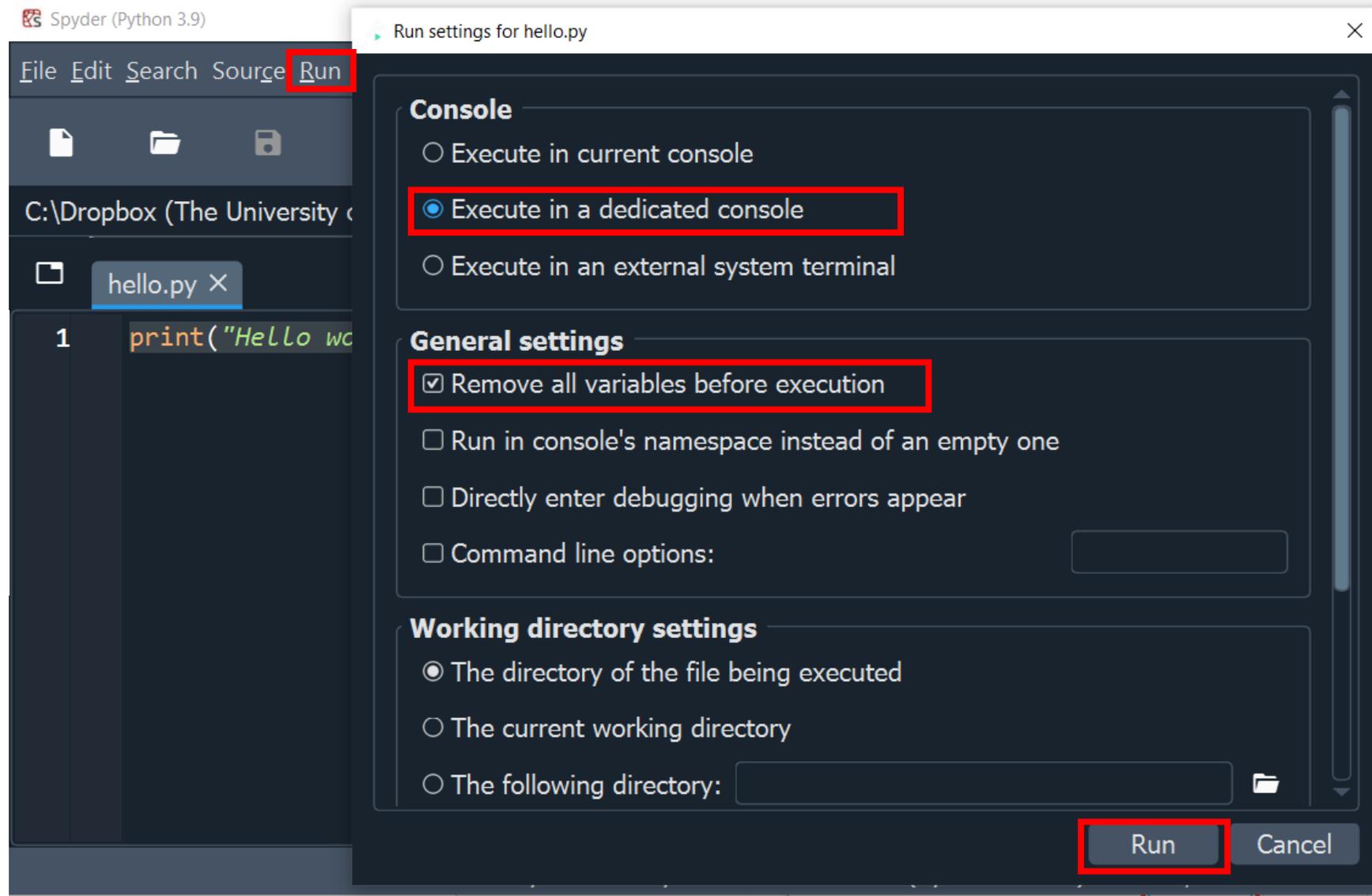


# Your first Python program ctd.

1. Click on Run -> Run (or hit the [F5] key)

2. Select the options

- Execute in a dedicated console
- Remove all variables before execution and click [Run]



# Your first Python program ctd.

Congratulations! Your first program should have printed something to the console.

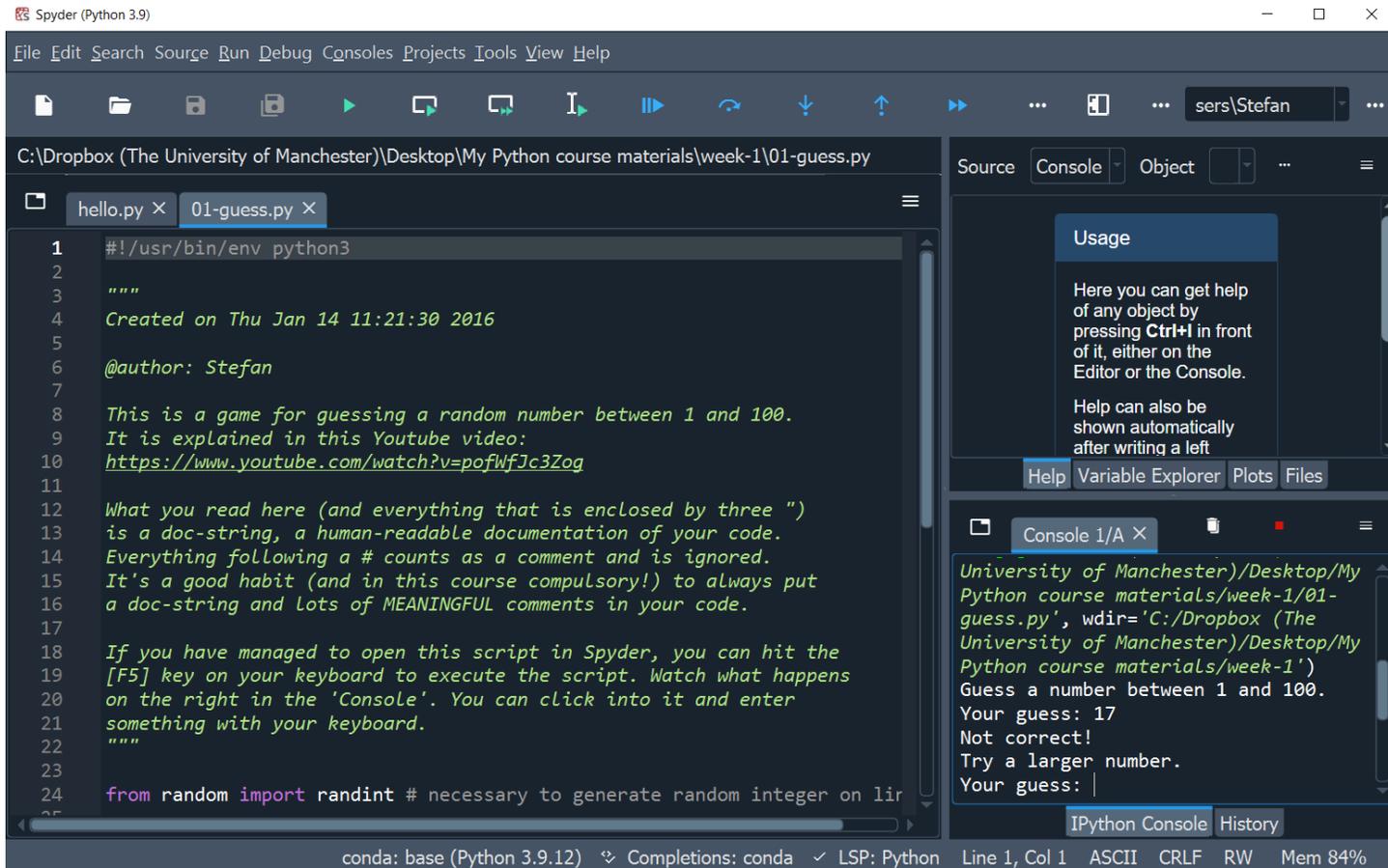
The image shows the Spyder Python IDE interface. The main editor window displays a single line of Python code: `print("Hello world")`. The file path is `C:\Dropbox (The University of Manchester)\Desktop\My Python course materials\hello.py`. The console window on the right shows the output of the program: `wdir='C:/Dropbox (The University of Manchester)/Desktop/My Python course materials'` and `Hello world`. The console prompt is `In [2]: |`. The status bar at the bottom indicates `LSP Python: ready`, `conda: base (Python 3.9.12)`, `Line 1, Col 19`, `UTF-8`, `CRLF`, `RW`, and `Mem 85%`. A green arrow points from the console output back to the code in the editor.

# A more interesting program: number guessing

Download the 01-guess.py file from the course website

Ideally, move it to a new “week-1” folder for all your codes (be organised from the start!)

Open the file in Spyder, run it, play with it, read the code, understand it? Modify?



The screenshot shows the Spyder Python IDE interface. The main editor window displays a Python script named '01-guess.py' with the following content:

```
1 #!/usr/bin/env python3
2
3 """
4 Created on Thu Jan 14 11:21:30 2016
5
6 @author: Stefan
7
8 This is a game for guessing a random number between 1 and 100.
9 It is explained in this Youtube video:
10 https://www.youtube.com/watch?v=pofWfJc3Zog
11
12 What you read here (and everything that is enclosed by three ")
13 is a doc-string, a human-readable documentation of your code.
14 Everything following a # counts as a comment and is ignored.
15 It's a good habit (and in this course compulsory!) to always put
16 a doc-string and lots of MEANINGFUL comments in your code.
17
18 If you have managed to open this script in Spyder, you can hit the
19 [F5] key on your keyboard to execute the script. Watch what happens
20 on the right in the 'Console'. You can click into it and enter
21 something with your keyboard.
22 """
23
24 from random import randint # necessary to generate random integer on li
```

The console window shows the output of the script:

```
University of Manchester)/Desktop/My
Python course materials/week-1/01-
guess.py', wdir='C:/Dropbox (The
University of Manchester)/Desktop/My
Python course materials/week-1')
Guess a number between 1 and 100.
Your guess: 17
Not correct!
Try a larger number.
Your guess: |
```

Don't be afraid to modify the code.  
It's likely you will break it!  
But your computer won't break.  
Try fix the code again.  
And again.  
Until it works.  
That's the only way to become a programmer 😊

Refer to:

<https://www.youtube.com/watch?v=pofWfJc3Zog>



## Wrap up

- Overall format of the course (self-study lectures, review sessions, lab classes)
- Importance of scheduling your learning, keeping up with the material
- Making use of lab classes and the Blackboard forum
- Anaconda installation and first assignment: guessing game
- Other tools: [codeshare.io](https://codeshare.io) and [repl.it](https://repl.it)