



python<sup>TM</sup>

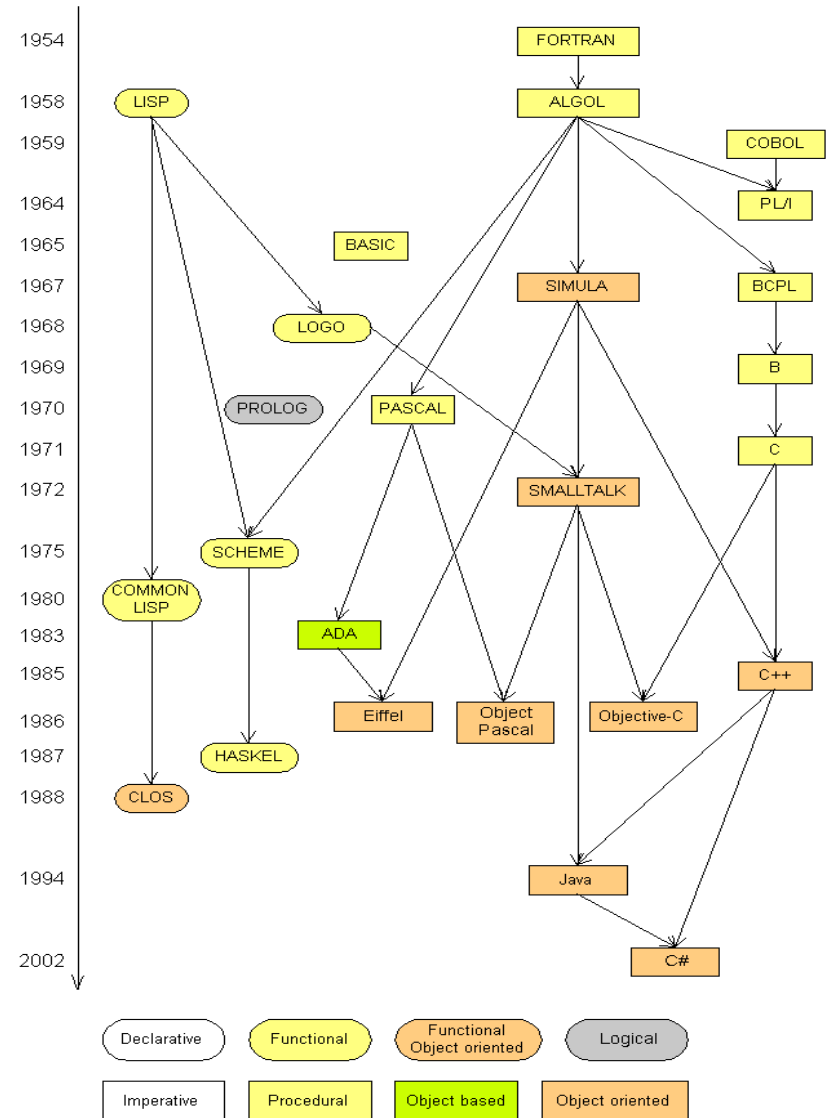
# Introduction

# Overview

- History
- Significance
- Installing & Running Python
- Simple script examples

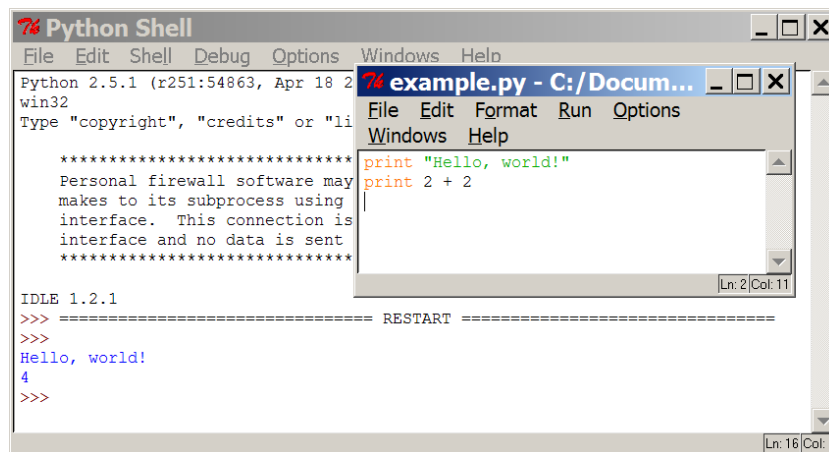
# Languages

- **Some influential ones:**
  - FORTRAN
    - science / engineering
  - COBOL
    - business data
  - LISP
    - logic and AI
  - BASIC
    - a simple language



# Programming basics

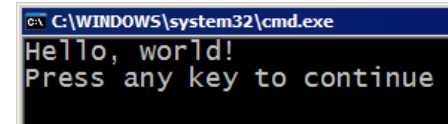
- **code or source code:** The sequence of instructions in a program.
- **syntax:** The set of legal structures and commands that can be used in a particular programming language.
- **output:** The messages printed to the user by a program.
- **console:** The text box onto which output is printed.
  - Some source code editors pop up the console as an external window, and others contain their own console window.



The image shows two overlapping windows. The background window is the 'Python Shell' (IDLE 1.2.1). It displays a message about firewall software and a prompt. The foreground window is a code editor titled 'example.py - C:/Docum...'. It contains the following code:

```
print "Hello, world!"
print 2 + 2
```

The code editor's status bar shows 'Ln: 2 | Col: 11'. The Python Shell's status bar shows 'Ln: 16 | Col: 4'.



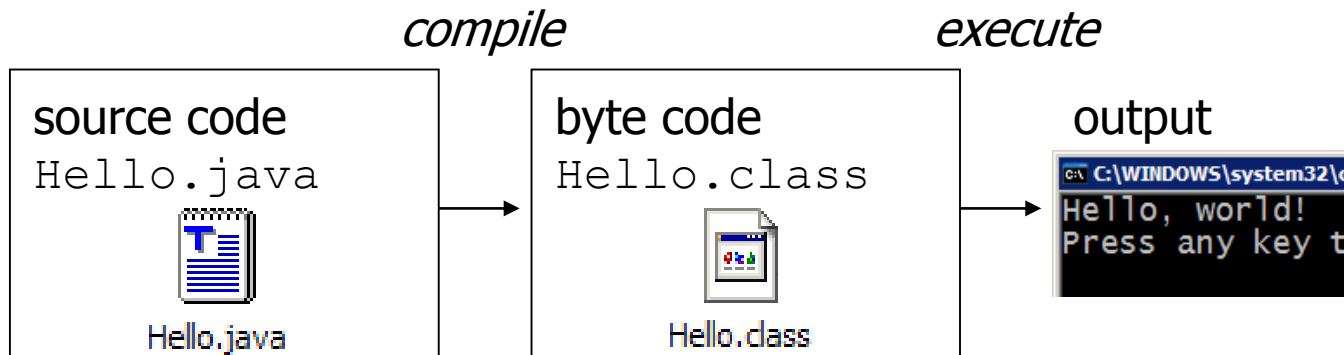
The image shows a Windows command prompt window titled 'C:\WINDOWS\system32\cmd.exe'. It displays the output of the Python program:

```
Hello, world!
Press any key to continue
```

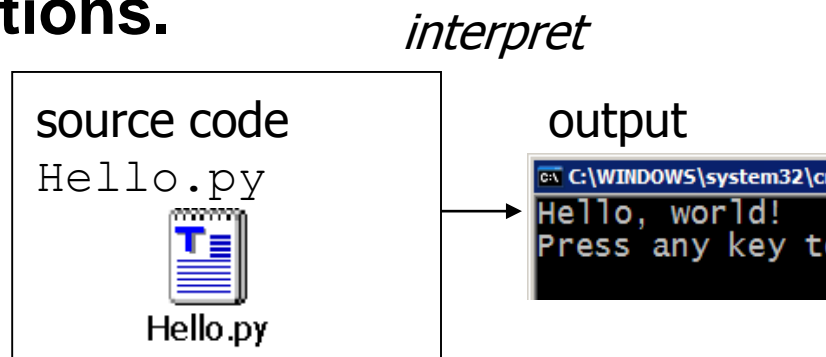


# Compiling and interpreting

- Many languages require you to *compile* (translate) your program into a form that the machine understands.



- Python is instead directly *interpreted* into machine instructions.

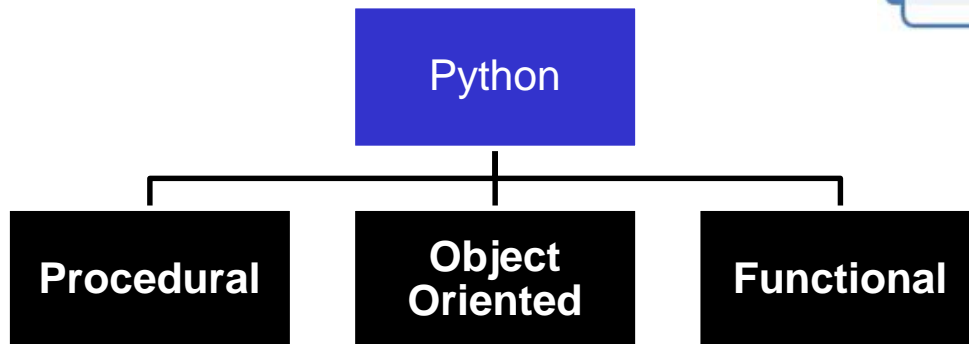
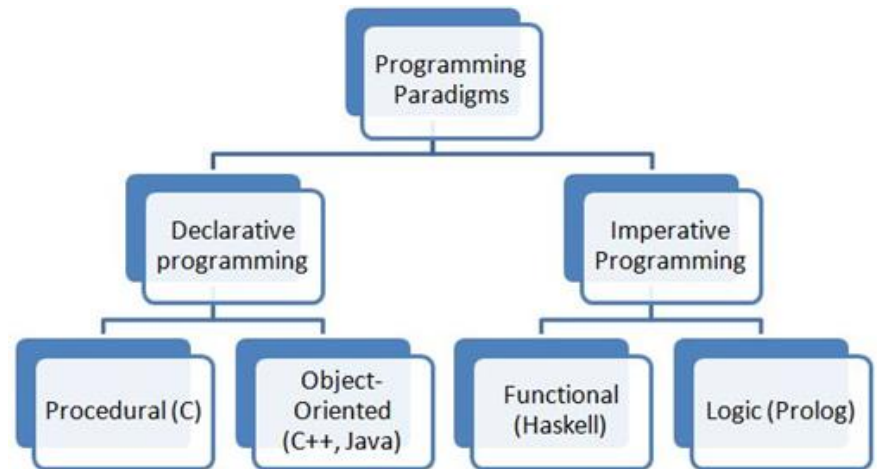


# Brief History of Python

- Invented in the Netherlands, early 90s by Guido van Rossum ([video](#))
- Named after Monty Python (British surreal comedy troupe)
- Open sourced from the beginning, man-aged by [Python Software Foundation](#)
- Considered a scripting language, but is much more
- Scalable, object oriented and functional from the beginning
- Used by Google from the beginning



# Programming Paradim



# Python's Benevolent Dictator For Life

“Python is an experiment in how much freedom program-mers need. Too much freedom and nobody can read another's code; too little and expressive-ness is endangered.”

- [Guido van Rossum](#)



Guido van Rossum






# Python's place in the Market


- TIOBE has been collecting data on programming language “popularity” for many years
- Counts results for a query like "<language> programming" on popular search engines

Apr 2020	Apr 2019	Change	Programming Language	Ratings	Change
1	1		Java	16.73%	+1.69%
2	2		C	16.72%	+2.64%
3	4	⬆	Python	9.31%	+1.15%
4	3	⬇	C++	6.78%	-2.06%
5	6	⬆	C#	4.74%	+1.23%
6	5	⬇	Visual Basic	4.72%	-1.07%
7	7		JavaScript	2.38%	-0.12%
8	9	⬆	PHP	2.37%	+0.13%
9	8	⬇	SQL	2.17%	-0.10%
10	16	⬆	R	1.54%	+0.35%
11	19	⬆	Swift	1.52%	+0.54%
12	18	⬆	Go	1.36%	+0.35%
13	13		Ruby	1.25%	-0.02%
14	10	⬇	Assembly language	1.16%	-0.55%
15	22	⬆	PL/SQL	1.05%	+0.26%
16	14	⬇	Perl	0.97%	-0.30%
17	11	⬇	Objective-C	0.94%	-0.57%
18	12	⬇	MATLAB	0.93%	-0.36%
19	17	⬇	Classic Visual Basic	0.83%	-0.23%
20	27	⬆	Scratch	0.77%	+0.28%

# http://python.org/

[Python](#)[PSF](#)[Docs](#)[PyPI](#)[Jobs](#)[Community](#)



[GO](#)[Socialize](#)[Sign In](#)

[About](#)[Downloads](#)[Documentation](#)[Community](#)[Success Stories](#)[News](#)[Events](#)

```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

### Functions Defined

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. [More about defining functions in Python 3](#)

[1](#)[2](#)[3](#)[4](#)[5](#)

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)

## Get Started

Whether you're new to programming or an experienced developer, it's easy to learn and use Python.

## Download

Python source code and installers are available for download for all versions! Not sure which version to use? [Click here](#)

## Docs

Documentation for Python's standard library, along with tutorials and guides, are available online.

## Jobs

Looking for work or have a Python related position that you're trying to hire for? Our **relaunched** [community job board](#) is the



# http://docs.python.org/

Python » English » 3.8.2 » Documentation »

## Download

Download these documents

## Docs by version

[Python 3.9 \(in development\)](#)  
[Python 3.8 \(stable\)](#)  
[Python 3.7 \(stable\)](#)  
[Python 3.6 \(security-fixes\)](#)  
[Python 3.5 \(security-fixes\)](#)  
[Python 2.7 \(EOL\)](#)  
[All versions](#)

## Other resources

[PEP Index](#)  
[Beginner's Guide](#)  
[Book List](#)  
[Audio/Visual Talks](#)  
[Python Developer's Guide](#)

## Python 3.8.2 documentation

Welcome! This is the documentation for Python 3.8.2.

### Parts of the documentation:

#### [What's new in Python 3.8?](#)

*or all "What's new" documents since 2.0*

#### [Tutorial](#)

*start here*

#### [Library Reference](#)

*keep this under your pillow*

#### [Language Reference](#)

*describes syntax and language elements*

#### [Python Setup and Usage](#)

*how to use Python on different platforms*

#### [Python HOWTOs](#)

*in-depth documents on specific topics*

#### [Installing Python Modules](#)

*installing from the Python Package Index & other sources*

#### [Distributing Python Modules](#)

*publishing modules for installation by others*

#### [Extending and Embedding](#)

*tutorial for C/C++ programmers*

#### [Python/C API](#)

*reference for C/C++ programmers*

#### [FAQs](#)

*frequently asked questions (with answers!)*

### Indices and tables:

#### [Global Module Index](#)

*quick access to all modules*

#### [General Index](#)

*all functions, classes, terms*

#### [Glossary](#)

*the most important terms explained*

#### [Search page](#)

*search this documentation*

#### [Complete Table of Contents](#)

*lists all sections and subsections*

### Meta information:

#### [Reporting bugs](#)

#### [History and License of Python](#)



# The Python tutorial is good!

Previous topic

[Changelog](#)

Next topic

[1. Whetting Your Appetite](#)

## This Page

[Report a Bug](#)

[Show Source](#)

## Quick search

Go

Enter search terms or a module,  
class or function name.

## The Python Tutorial

Python is an easy to learn, powerful programming language. It has efficient high-level data structures, a flexible, interpreted nature, make it an ideal language for scripting and rapid application development in many

The Python interpreter and the extensive standard library are freely available in source or binary form on many platforms. There are also distributions of and pointers to many free third party Python modules, programs and tools, and additional

The Python interpreter is easily extended with new functions and data types implemented in C or C++ modules.

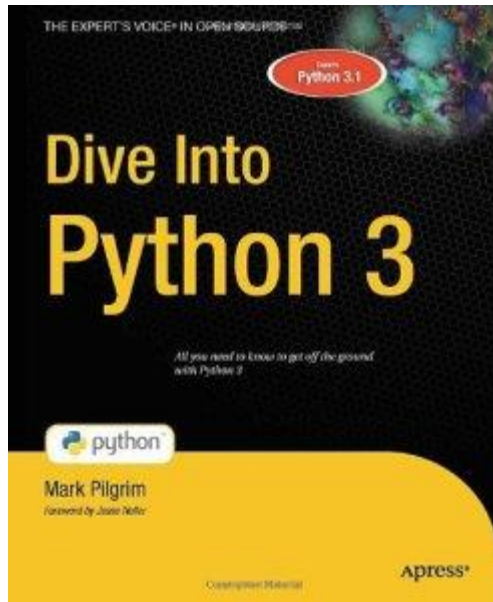
This tutorial introduces the reader informally to the basic concepts and features of the Python language and, in many ways, provides a concise, precise definition of the Python language. It can be read off-line as well.

For a description of standard objects and modules, see [The Python Standard Library](#). [The Python Language Reference](#), [the Python Interpreter](#) and [Python/C API Reference Manual](#). There are also several books covering

This tutorial does not attempt to be comprehensive and cover every single feature, or even every corner of the language. It is written in a simple, informal style. After reading it, you will be able to read and write Python modules and programs, and

The [Glossary](#) is also worth going through.

- [1. Whetting Your Appetite](#)
- [2. Using the Python Interpreter](#)
  - [2.1. Invoking the Interpreter](#)
    - [2.1.1. Argument Passing](#)
    - [2.1.2. Interactive Mode](#)
  - [2.2. The Interpreter and Its Environment](#)
    - [2.2.1. Source Code Encoding](#)
- [3. An Informal Introduction to Python](#)



# Running Python

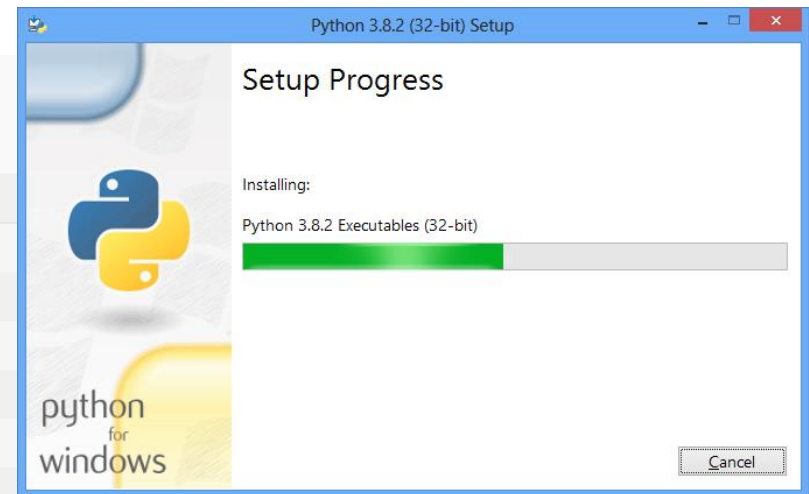
<http://histo.ucsf.edu/BMS270/diveintopython3-r802.pdf>

# Installing Python 3

- Visit [python.org/download/](https://python.org/download/) and download the last Python 3 (ver. Python 3.8.2) Windows installer

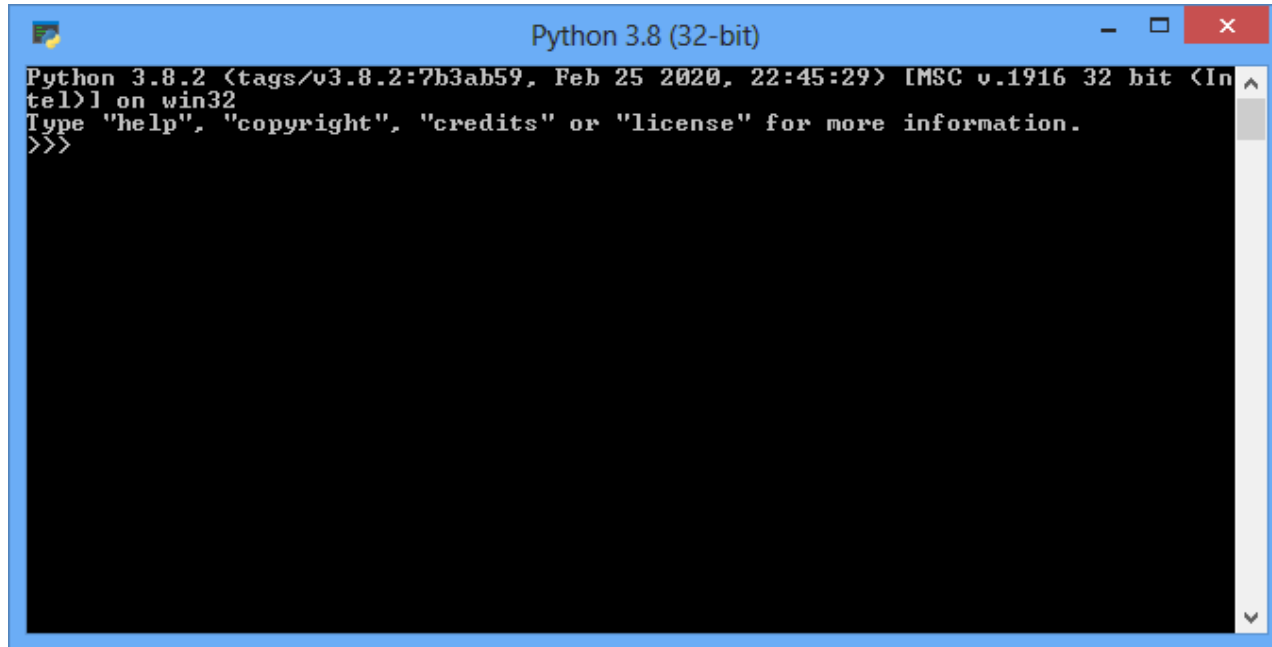
## Files

Version	Operating System	Description			
<a href="#">Gzipped source tarball</a>	Source release				
<a href="#">XZ compressed source tarball</a>	Source release				
<a href="#">macOS 64-bit installer</a>	Mac OS X	for OS X 10.9 and later			
<a href="#">Windows help file</a>	Windows				
<a href="#">Windows x86-64 embeddable zip file</a>	Windows	for AMD64/EM64T/x64			
<a href="#">Windows x86-64 executable installer</a>	Windows	for AMD64/EM64T/x64			
<a href="#">Windows x86-64 web-based installer</a>	Windows	for AMD64/EM64T/x64	2586cdad1a363d1a8abb5fc102b2d418	1363760	<a href="#">SIG</a>
<a href="#">Windows x86 embeddable zip file</a>	Windows		1b1f0fc5ee8601f160cfad5b560e3a7	7147713	<a href="#">SIG</a>
<a href="#">Windows x86 executable installer</a>	Windows		6f0ba59c7dbeba7bb0ee21682fe39748	26481424	<a href="#">SIG</a>
<a href="#">Windows x86 web-based installer</a>	Windows		04d97979534f4bd33752c183fc4ce680	1325416	<a href="#">SIG</a>



# Python command line

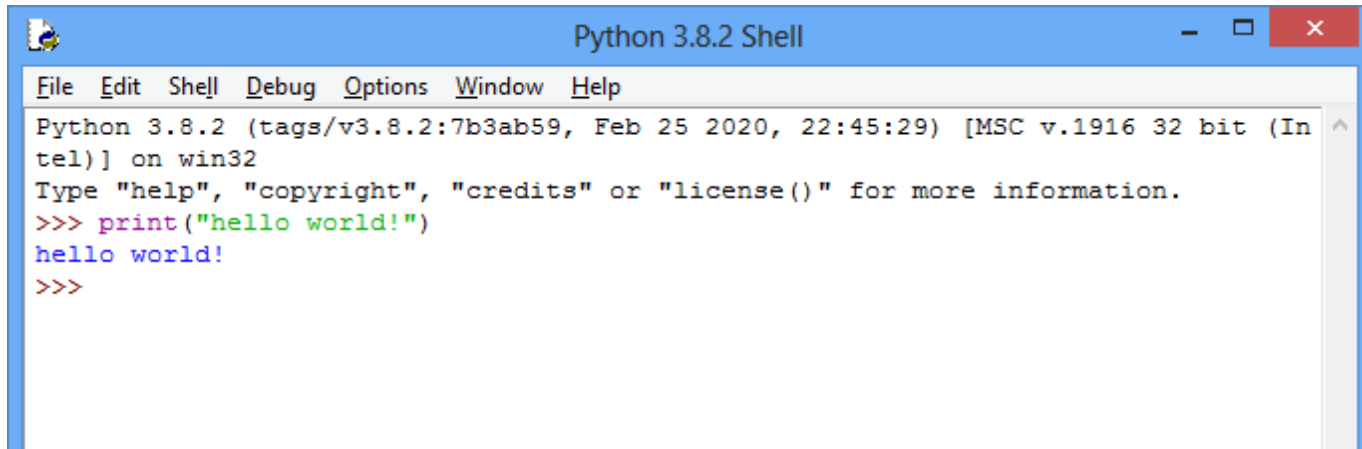
In Start menu, there should be a new item called  
Python 3.8

A screenshot of a Windows command prompt window titled "Python 3.8 (32-bit)". The window has a blue title bar with standard Windows window controls (minimize, maximize, close). The main area is black with white text. The text displayed is: "Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32", followed by "Type 'help', 'copyright', 'credits' or 'license' for more information.", and then the prompt ">>>" on a new line.

```
Python 3.8 (32-bit)
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

# IPython

- An enhanced interactive Python shell
- Included in your python installation as python shell



```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("hello world!")
hello world!
>>>
```



# Hello World program

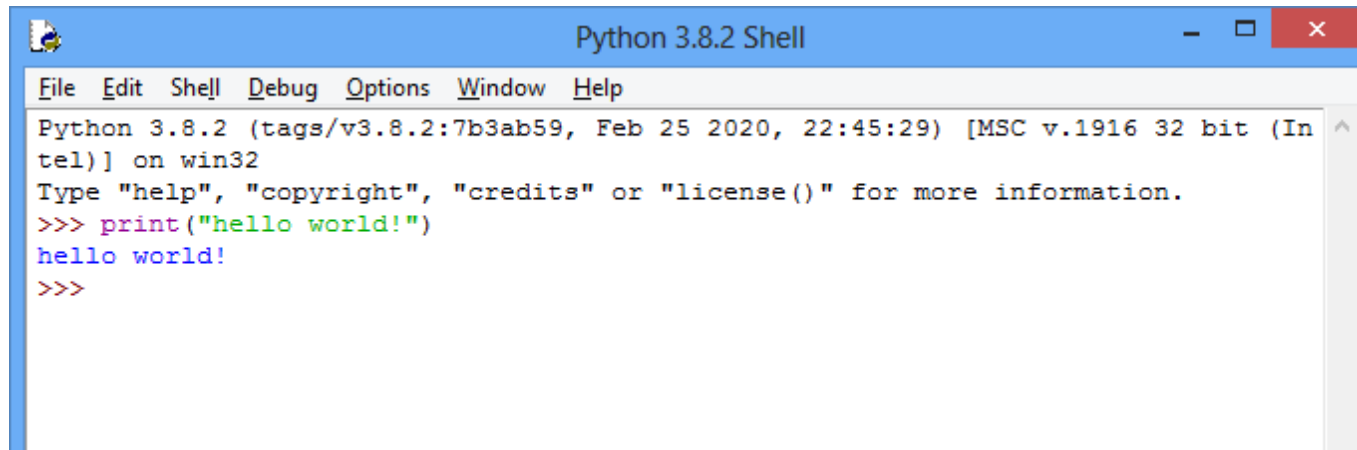
- Entered interactively in Python prompt:

Python prompt waits for input:

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

- Or in IPython prompt:

```
In [1]: print ( " Hello world " )  
Hello world
```



The screenshot shows a Windows-style application window titled "Python 3.8.2 Shell". The window has a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the following text:  
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>> print("hello world!")  
hello world!  
>>>



# All in one [ ANACONDA ]

- **Anaconda is a completely free Python distribution (i.e. set of packages) for scientific purposes, as stated in their website. It contains more than 125 Python packages for science, mathematics, engineering and data analysis.**

Available at <https://www.anaconda.com/distribution/>



Products ▾

Pricing

Solutions ▾

Resources ▾

Blog

Company ▾

Get Started



Individual Edition

## Your data science toolkit

With over 20 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.

Download



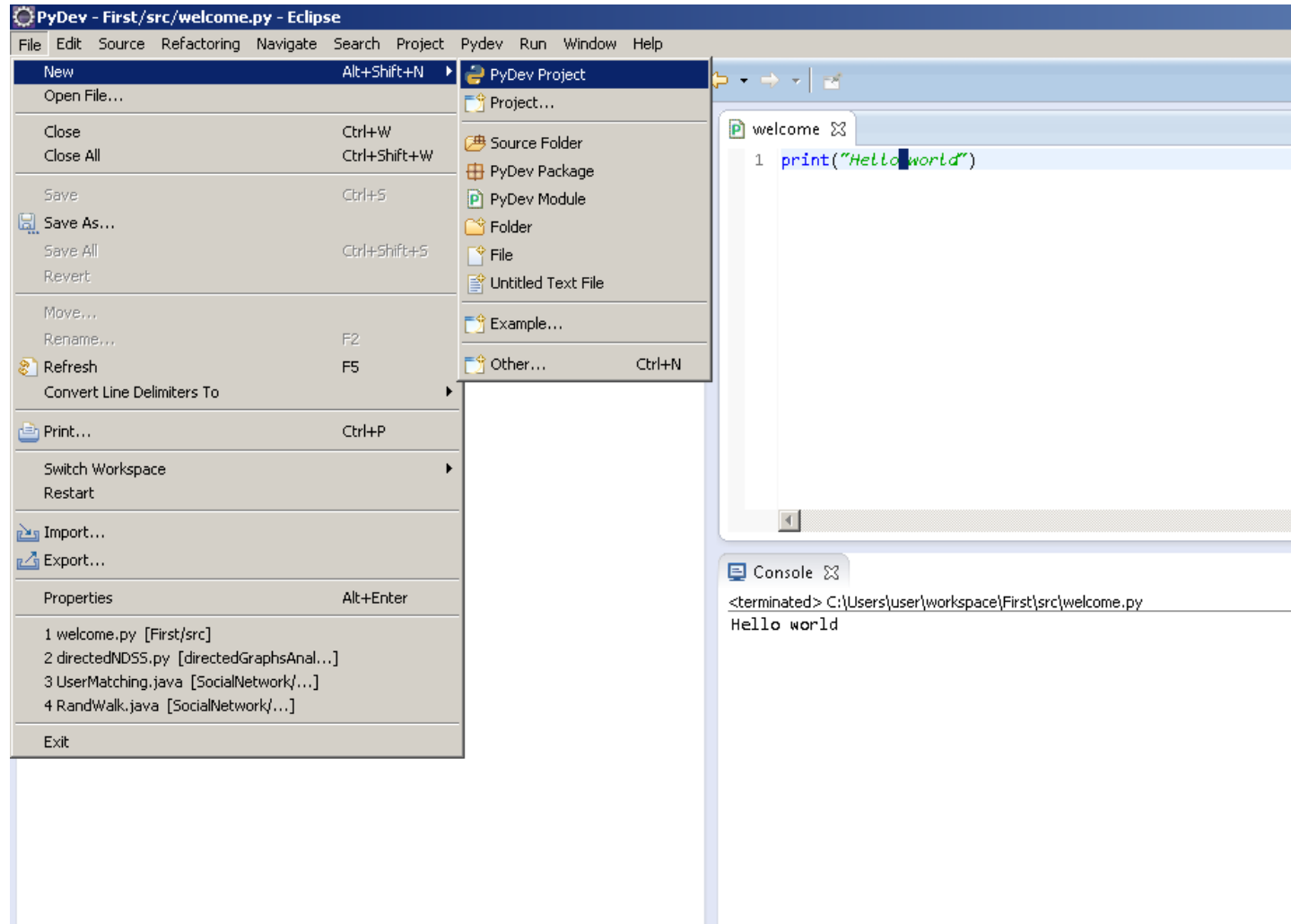
# Installing e Plugin to Eclipse

- PyDev Plugin is used to let [eclipse](#) support python software development
- You must install python before installing the plugin
- More details about installing PyDev are in the following link
- Simply open eclipse > help>marketplace>search for PyDev . Install the plugin
  - [http://www.pydev.org/manual\\_101\\_install.html](http://www.pydev.org/manual_101_install.html)
  - In the [page](#), start reading the installation instruction from **Installing with the update site**
- **Note: after installing pyDev you need to restart eclipse**



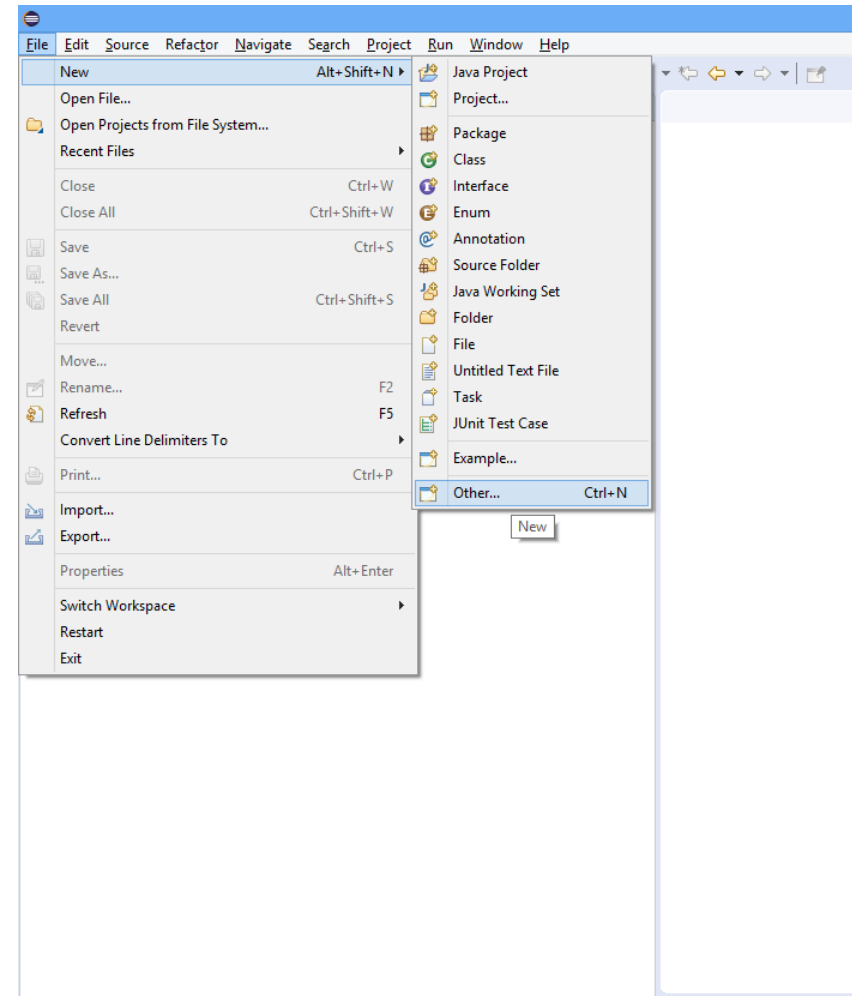
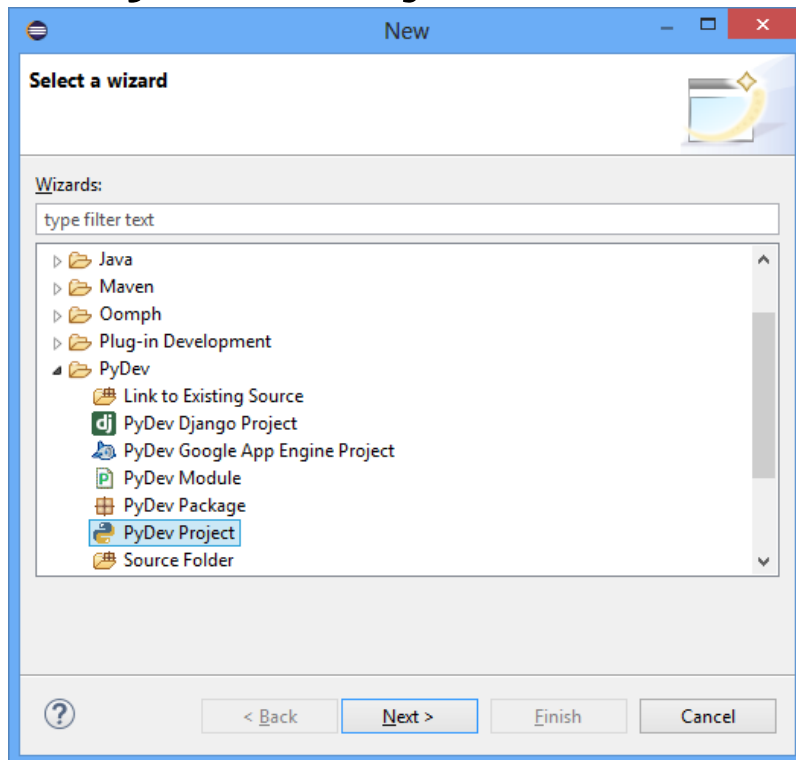
# Using eclipse to write python

## 1. Open eclipse and Create a new pyDev project



# Create Python Project

- Open eclipse > file > new > other ...
- Choose PyDev > PyDev Project



# Create Python Project

- Write the project name
- Choose grammar version 3
- Choose the src option to create project with src folder

**PyDev Project**  
❌ Project interpreter not specified

Project name:

Project contents:  
☒ Use default  
Directory:

Project type  
Choose the project type  
☒ Python ☐ Jython ☐ IronPython

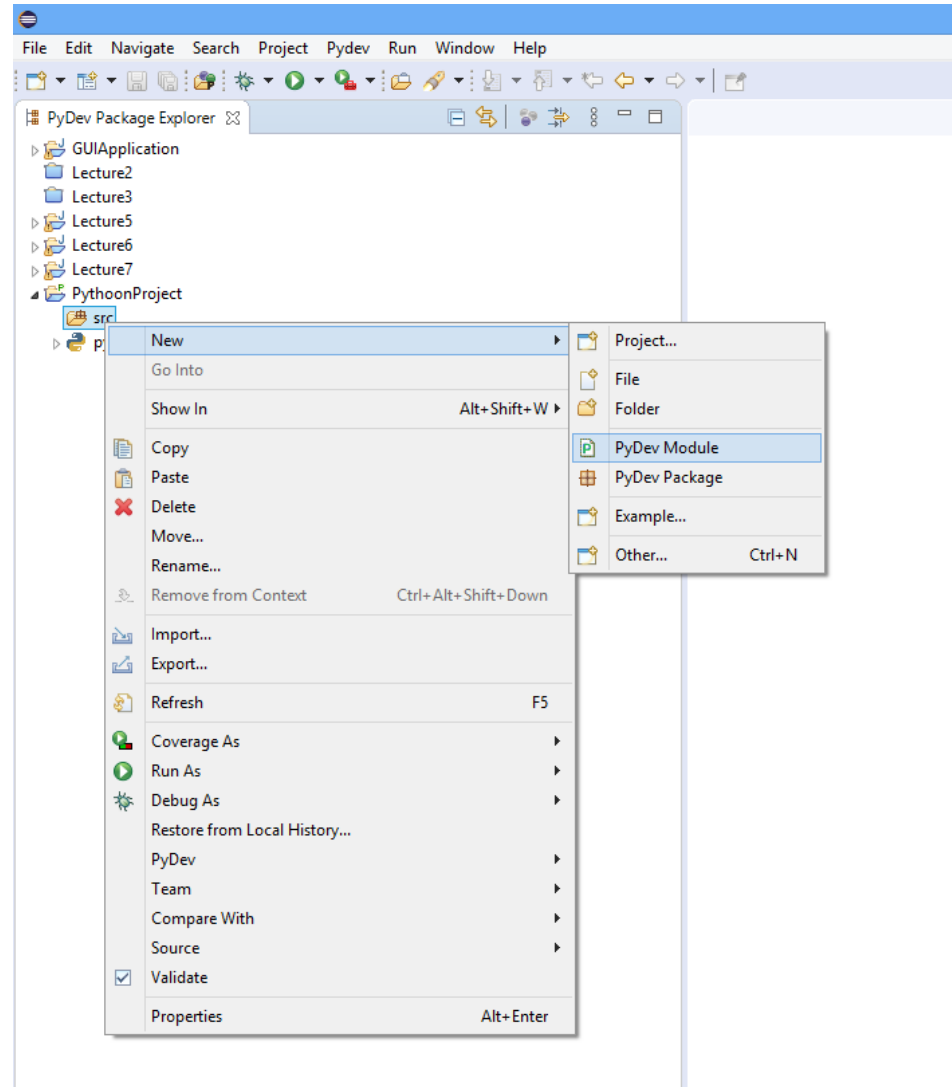
Grammar Version  
Same as interpreter  
2.5  
2.6  
2.7  
3.5  
3.6  
3.7  
3.8  
Same as interpreter

☒ Add project directory to the PYTHONPATH  
☐ Create 'src' folder and add it to the PYTHONPATH  
☐ Create links to existing sources (select them on the next page)  
☐ Don't configure PYTHONPATH (to be done manually later on)

Working sets  
☐ Add project to working sets   
Working sets:

# Add python file to Project

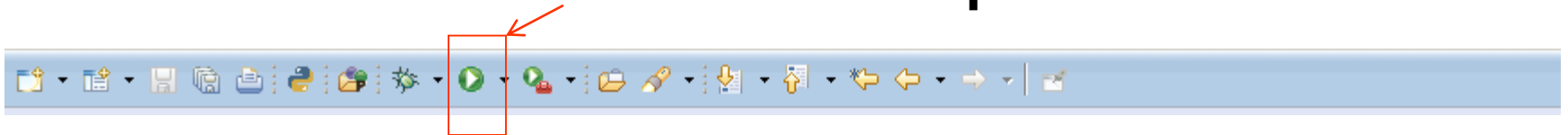
- Python code will be saved at a file with **py** extension
- To create one file in your project
  - Do right click at the src folder
  - Choose file



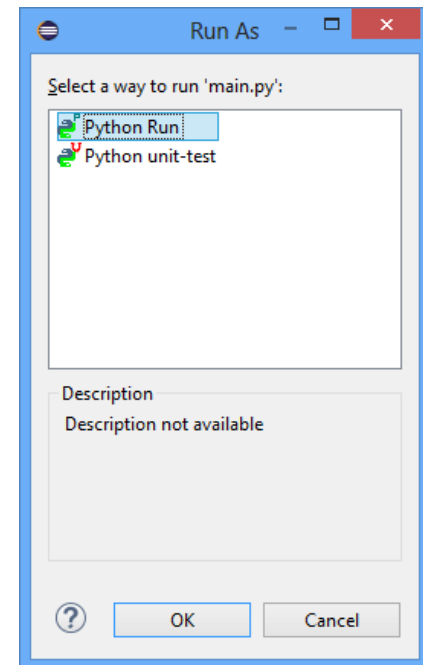
- Write a simple Hello world output

```
print("Hello world")
```

- Then click on run button on eclipse

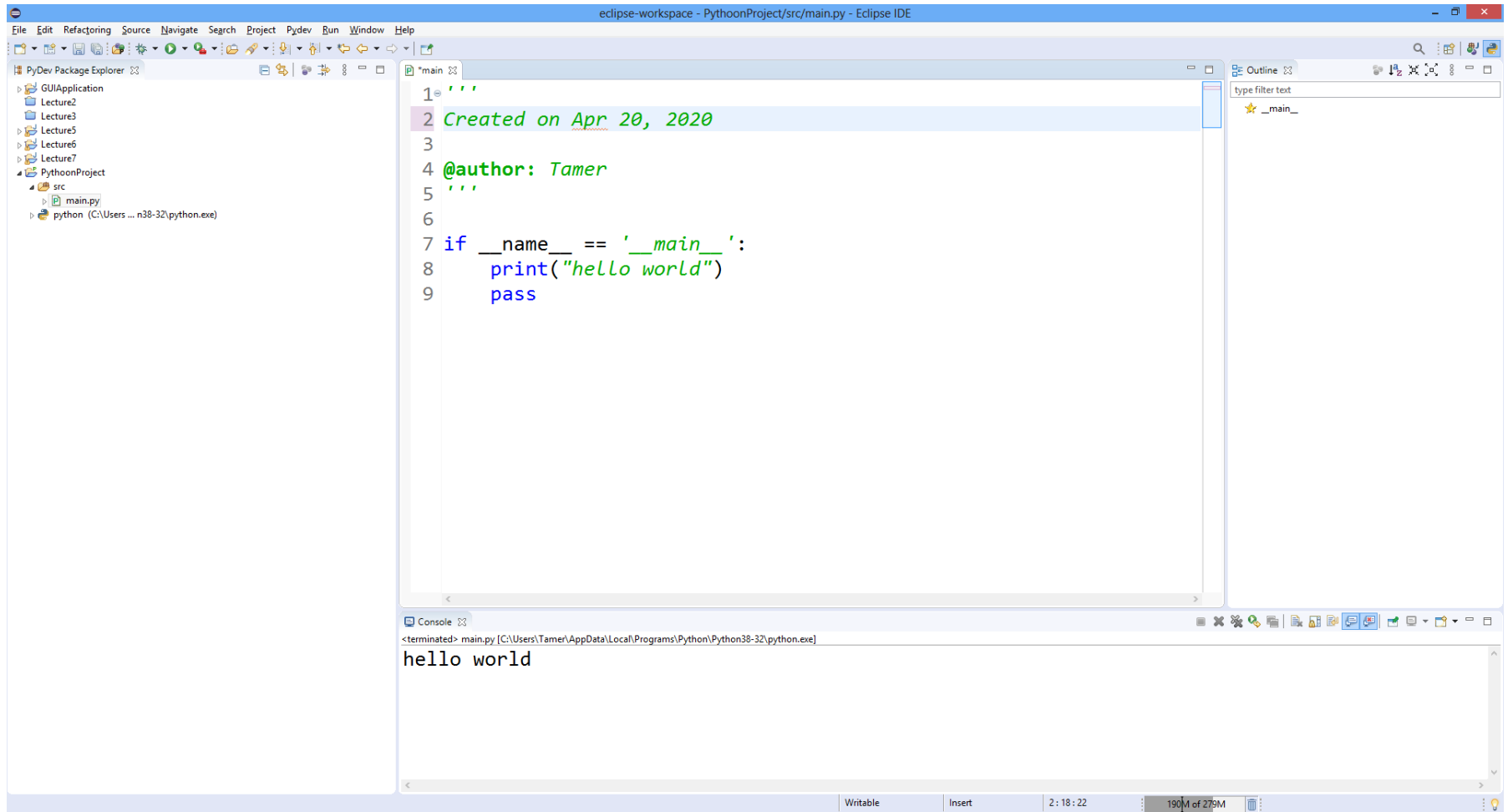


- Eclipse will ask you to run as choose Python run





# Start writing python



The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Refactoring, Source, Navigate, Search, Project, Pydev, Run, Window, and Help. The left sidebar shows the Project Explorer with a tree view of the project structure: GUIApplication, Lecture2, Lecture3, Lecture5, Lecture6, Lecture7, and PythoonProject. The PythoonProject folder is expanded, showing a src folder containing main.py. The main editor window displays the code for main.py, which is a simple 'hello world' program. The code is as follows:

```
1 '''  
2 Created on Apr 20, 2020  
3  
4 @author: Tamer  
5 '''  
6  
7 if __name__ == '__main__':  
8     print("hello world")  
9     pass
```

The right sidebar shows the Outline view with a search bar and a list of symbols, including \_\_main\_\_.

The bottom console window shows the output of the program, which is 'hello world'. The console also displays the path to the Python interpreter: C:\Users\Tamer\AppData\Local\Programs\Python\Python38-32\python.exe.

