

## Introduction to Eclipse and PyDev

We will be using Eclipse PyDev plugin in this course to create and run Python programs. Eclipse is a platform that can be used for application development tasks using different languages, including Java and Python. A separate handout describes how to install Eclipse and its Python plugin - PyDev. This handout will take you through the process of creating and running a Python program.

### Specifying the Eclipse workspace

When you start Eclipse, you will be asked to specify the directory in which to store your projects. By default, on Windows, it creates a folder called *eclipse-workspace* in your personal folder. Use this directory or another of your choice. To switch the workspace to a different folder, after starting Eclipse, go to **File > Switch Workspace** and select the desired folder.

### Creating a new PyDev project

To create and run a Python program in Eclipse, you must form a **project** for it:

1. Create a new PyDev Project by selecting **File > New > Project > PyDev > PyDev Project**. This will open the **PyDev Project** dialog window (see Figure 1)
2. In the dialog window:
  - Enter a project name of your choice into the Project name field.
3. In the **PyDev Project** dialog box (Figure 1), make sure the right interpreter is selected (To make sure python version 3 is selected, click on [Configure the interpreter not listed](#) link verifying that the selected interpreter's location should be within **anaconda3** installation, i.e. mention **anaconda3** in the path (Figure 2) ). On a Mac, the default location of the anaconda installation is `/Users/<username>/opt/anaconda3/bin`. Click on **Finish**.

A project folder will be added to your workspace, which you can see in the window pane on the left side of the Eclipse Editor window (Figure 3).

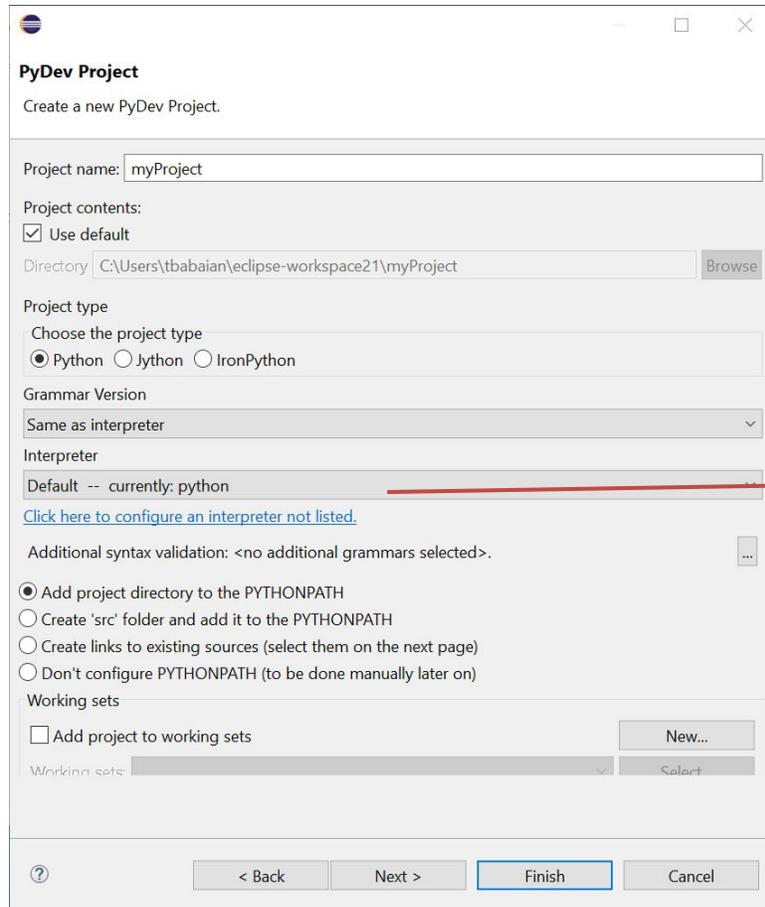


Figure 1. PyDev Project window. Select python (shown underlined in red)

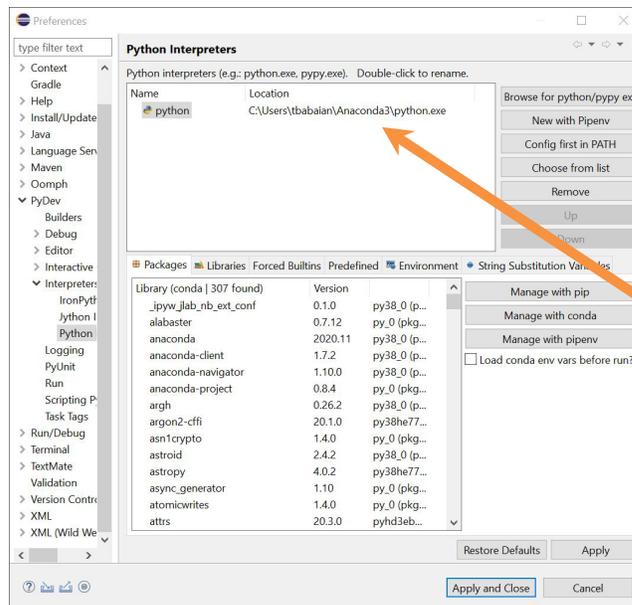


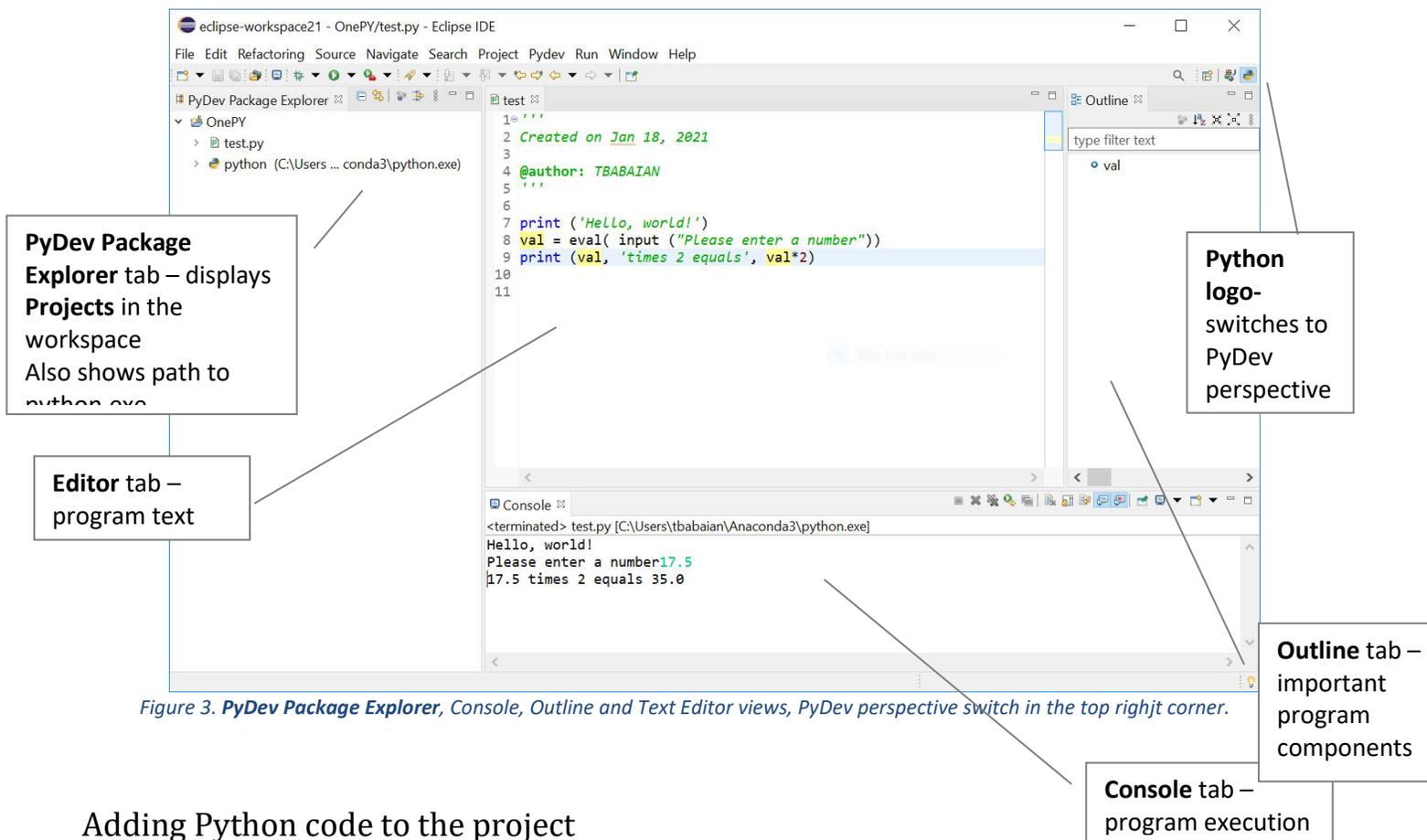
Figure 2. Preferences. Finding and renaming python interpreters (python under anaconda3 path)

## Eclipse Perspectives and Views

The Eclipse workbench provides several *views* and *perspectives* to access information regarding a project. A view is a window that displays specific kinds of information regarding tasks. Each perspective is a collection of views designed for a certain kind of project. We are going to be using the PyDev perspective. Later in the course, we will also be using the Debug perspective. We will make use of the following views (see Figure 1):

- PyDev Package Explorer** – provides access to project files,
- Console** – displays the textual input/output from a running Java program (i.e., program interaction)
- Outline** – displays variables and functions associated with the file that is being edited.

The **Java** perspective is the default. To switch to **PyDev**, if it isn't already selected, choose **Window > Perspective > Open Perspective > Other > PyDev**. Then, select the **PyDev Package Explorer** view by choosing **Window > Show View > PyDev Package Explorer** to see and access the contents of each project directory. The **PyDev Package Explorer** view is displayed on the left side of the Eclipse window. Select the **Console** view in the same manner, and notice that it usually appears at the bottom of the window.



## Adding Python code to the project

To add a Python file to a project, first, make sure the project is selected in the Navigator view by clicking on it. Make sure you have the PyDev perspective selected by clicking on the python logo in the top right corner. You can add python code using either one of two ways:

- Select **File> New> PyDev Module** and specify the name of the file (without an extension)

- Select **File > New > File >** and specifying a file name with extension `.py`.

The editor will open the newly created file. Add the following line to create a program that prints the phrase "Hello, world!":

```
print ("Hello, world!")
```

Save by pressing Ctrl-s or by clicking the Save button. Eclipse will also periodically save your file for you. The Eclipse editor contains several useful features, including syntax highlighting, parenthesis matching, automatic display of help and much more. One of the most useful is the **Refactoring**.

## Compilation and Syntax errors

Eclipse runs the interpreter each time a program is saved. If the interpreter detects syntax errors in the program, the error messages will appear in the **Problems** view pane on the bottom of the window. Each line containing an error will also be marked in the editor pane by a red circle with an X inside it. To experience syntax error detection, add an extra space to the front of the line you entered before, and save the file. Place the cursor on the displayed error mark and read the error description that appears.

## Running the program

When a program is free of syntax errors, you can execute it by clicking the Run button (a green circle with a white arrow inside). Alternatively, right-click the `.py` file in the Navigator pane or the Editor window and select **Run As > Python Run**.

Notice the **Console** tab appearing on the bottom window. This view shows the output from running the program and also allows the user to enter input data as necessary. At this time, the Console should display the following string:

```
Hello, world!
```

## Add an existing program to a project

Here's how to create a project for an existing program, such as one that has been posted to the course website:

1. Select **File > New > PyDev Project**.
2. In the next window, specify the name of the project and click on the **Finish** button.
3. Add existing files to the new project in one of two ways:
  - Download the files directly into the project folder that was just created in your workspace. Then press F5 on that folder in the Navigator tab to refresh the project's content. The files should now show up in the **PyDev Package Explorer** pane.
  - Download the files somewhere else on your computer, such as the desktop. Then drag and drop them into the project folder in the **PyDev Package Explorer** tab. You will be asked to select how files should be imported into the project. If you choose to *Copy files*, then copies of the files will be created in your current project folder. If you choose to *Link to files*, then links will be created to wherever you downloaded the files.

## Running in Interpreter Command-Line Mode – PyDev Console

You can simulate the way python works in a command-line form, where each line of code is evaluated immediately, by executing the PyDev Console. This is a very useful tool, when you are experimenting, trying to learn about a new function, try it out, see how it works.

To access the PyDev Console, go to the Console tab and click on an icon with a yellow plus sign. Select PyDev Console, as shown in Figure 4.

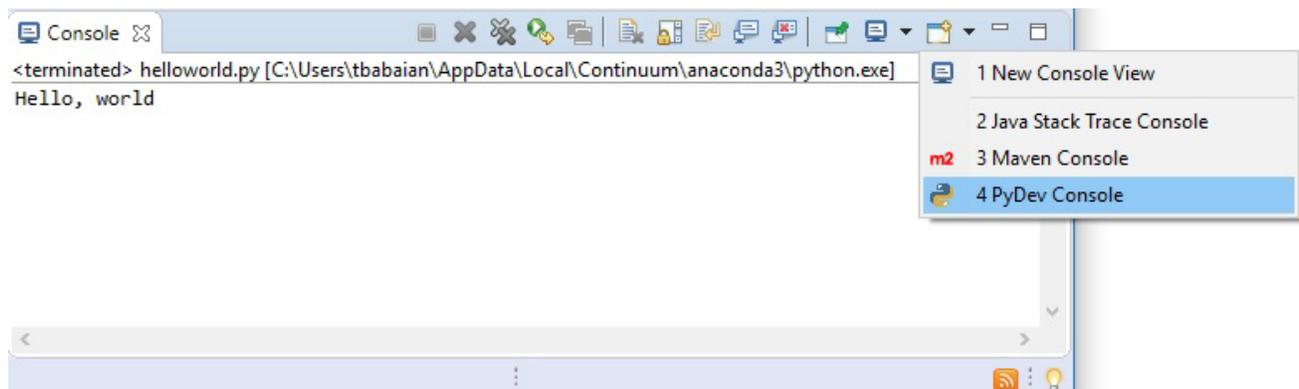


Figure 4. Console tab; bringing up the PyDev Console

In the popup window that appears, choose Python Console, and then select the right interpreter. That should result in the PyDev Console becoming available to you; the second line that shows up specifies the reference to the python interpreter used— verify that it has **anaconda3** in the path. As shown in Figure 5, you can type code in this console and execute it one line at a time.

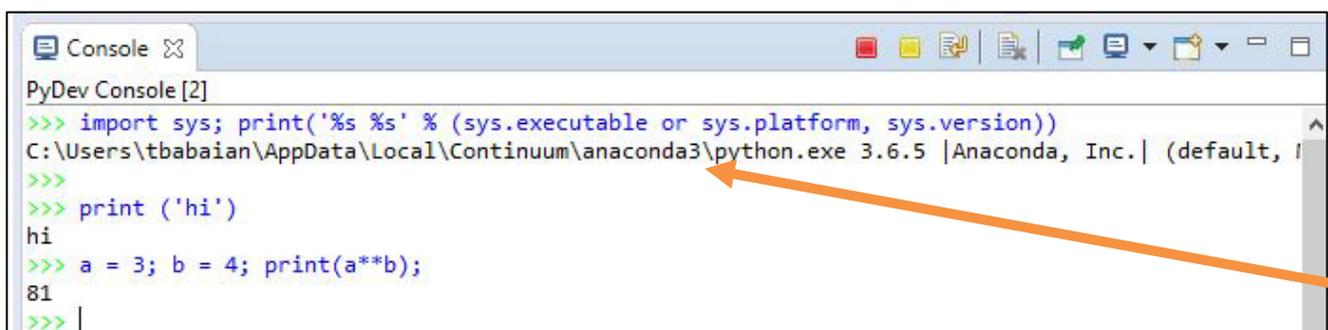


Figure 5. PyDev Console in action.

You can switch between the PyDev Console and the regular Console tab, which shows output of the programs, by selecting one or the other from the drop-down list that looks like a screen button on the Console tab menu.

## Customizing the editor and color scheme as shown in class.

Eclipse and PyDev allow customization of the environment, including the appearance of code (fonts and colors). I have changed the default settings, and if you wish to follow the schema used in class, set it up as shown in the next figure by selecting Window → Preferences → PyDev → Editor and choosing the colors for different lexical elements as shown in Figure 6.

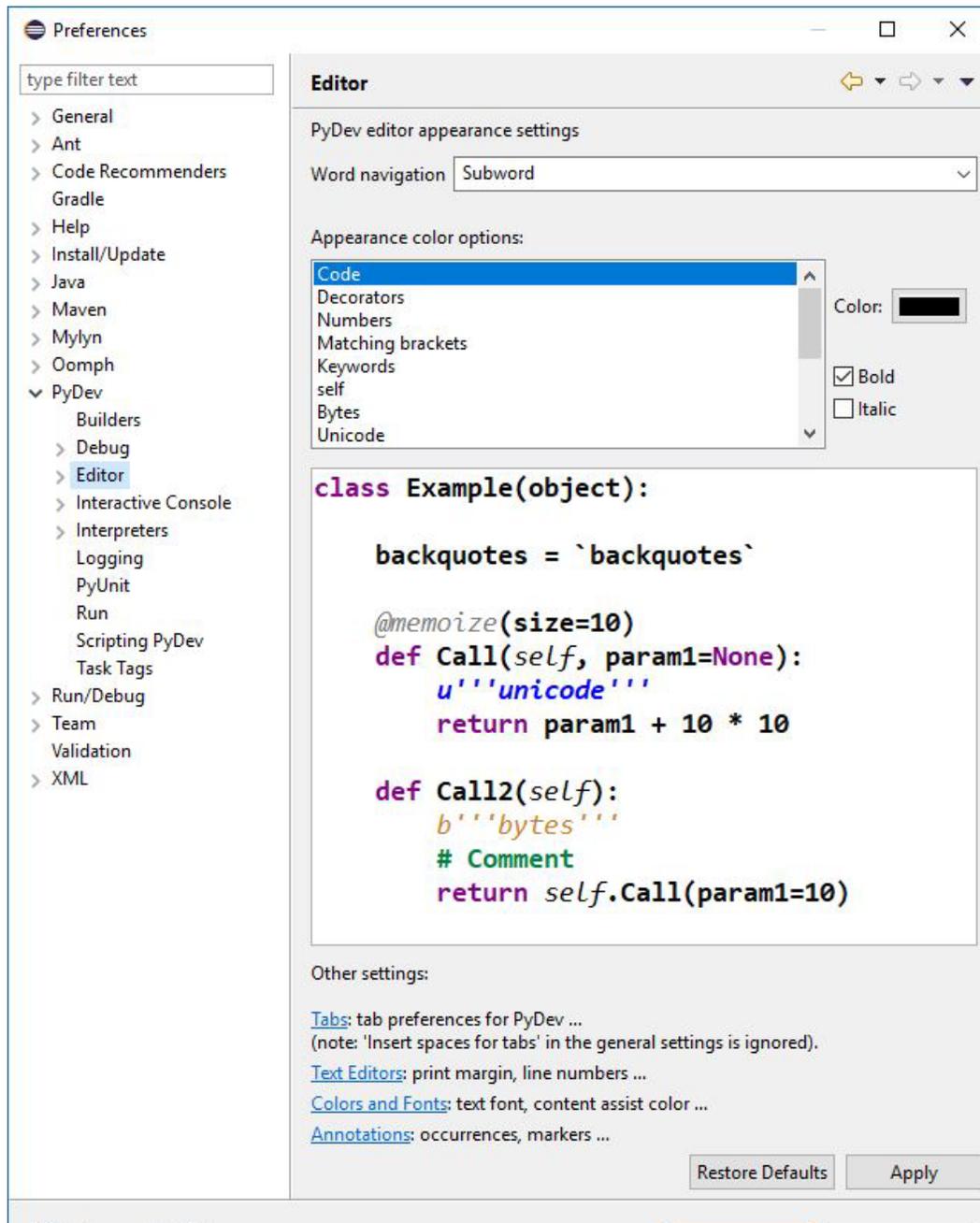


Figure 6. Changing colors for syntax highlighting.