Pycharm

Pyclub

27 Feb 2018

1 The interface

1.1 Getting started

A project is just a simple folder with a folder called .idea (which contains metadata for pycharm for that specific project) hidden inside. Thus, the simplest way to create a project where you want is to create a folder and select your folder after clicking *open project* Right click on the folder in the project and select new > python file

1.2 The interface

- You can use the python console at the bottom to execute code but this is not the most efficient way, it is more efficient to use debug or to use jupyter after linking the jupyter kernel to the pycharm debugger (we will cover this with jupyter)
- The right margin displays the location of issues or todo items as color coded strips.
- The left margin displays line number and is used to insert break points.
- At the very bottom, the status bar is active when analysing the project project and will be used when working with git.

1.3 Customising the interface

Go to View > Tool windows

- Enable the structure panel
- Enable the data panel

1.4 Further customisation

Always remember to use the search feature in the settings menu

- Select the interpreter for your project
- Choose your keymap
- Configure TODO items

2 Running code

2.1 What is the interpreter

Python is an interpreted language. This means that the code you write (with mostly english words) needs to be translated to instructions the processor of the machine can execute. This job of translation is what the python interpreter does. You can use the interpreter either by telling it to run your program (i.e. calling python name_of_the_program.py) or by using the interactive interpreter like with jupyter (somewhat similar to the bottom panel in matlab).

2.2 Configuring the interpreter

To execute and analyse your code, the Pycharm IDE needs to know which interpreter to use and where to find it.

1. Create a startup point Run > Edit Configurations

   script select your program

   parameters options as on command line

   python interpreter select your anaconda environment interpreter. It will offer by default to use the interpreter selected in the settings menu.

2. Insert break point in the margin.
3. Click the icon with the bug to start.
4. To restart a debugging session click the circular arrow beside the debug window. (clicking the bug again will start a parallel process to the first one).

3 Using the features

3.1 Navigation

- ctrl+click on element to jump to definition
- back mouse button (or ctrl+alt+left) to go back

3.2 Use the features of the IDE

- Use the interface to install packages (will indicate in top bar)
- Use suggested quick fixes (use alt+enter)
- Use code completion
- Use rename
- Use extract variable / use inline variable
- Right click find usages
- Use live templates (code > insert live template) (e.g. for or main) (configure yours and contribute them) File — Settings — Editor — Intentions
- Use suggested refactorings (intention actions) **wisely**
- Use View > Quick definition
- Use View > Quick documentation (configurable)