Creating a Productive Python Environment

Dr. Chris Brown

Python Development Environments

Python interpreter

+ plain old text editor

iPython

Enhanced interpreter

IDLE

Minimal Integrated Environment

PyCharm

Fully Integrated Environment

Python Interpreter

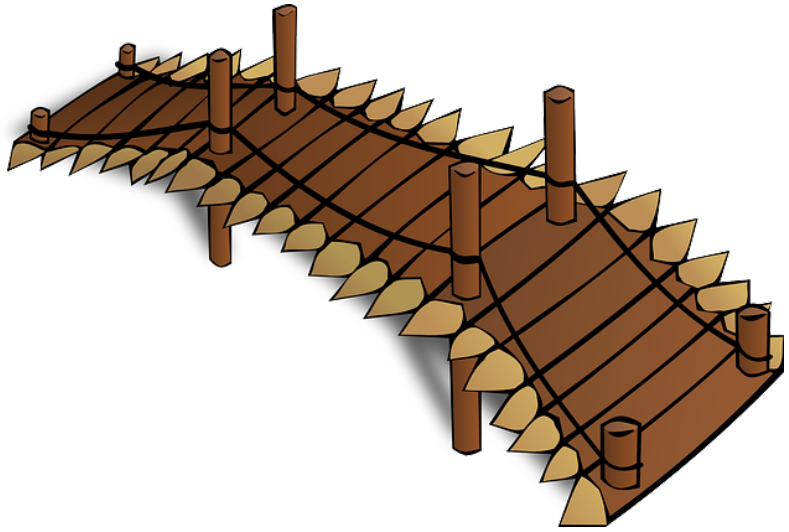
Using the Python Interpreter Interactively

Read
Evaluate
Print
Loop

IPython

Introducing IPython

bash



python

Enhanced python interpreter

REPL + “Magic” commands

- Built-in shell-like commands
- Aliases for arbitrary shell commands

Easy to capture command output

Building a bridge between bash and python

IDLE

Introducing IDLE

Written in Python using Tkinter toolkit

“No frills” development environment

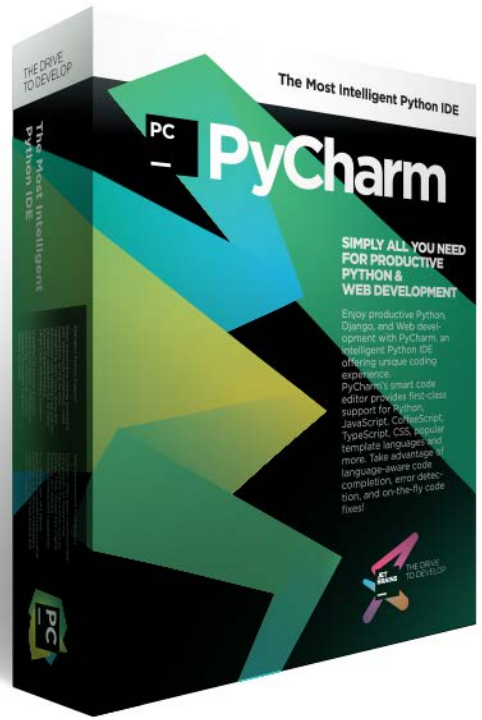
- Shell window (REPL)
- Language-aware editor window
- Class browser
- Simple debugger

Editor support for:

- Colour syntax highlighting
- Code completion
- Automatic indentation

PyCharm

Introducing PyCharm



Full Integrated Development Environment

- Professional and Community Editions
- Python-aware editor
- Code completion
- Integration with source code control
- Visual debugger
- Code refactoring
- Support for multiple interpreters
- Intention actions
- Console REPL

PyCharm use cases:

Large projects

Dozens of files

1000’s of lines of code

Complex logic

Summary

Python
Interpreter
+
Text Editor

REPL useful for familiarisation and testing

Four basic data types:

- String, integer, float, boolean

Creating a standalone script

IPython

- Magic functions
- Command aliases
- Shell escapes
- TAB completion
- Capturing and processing command output

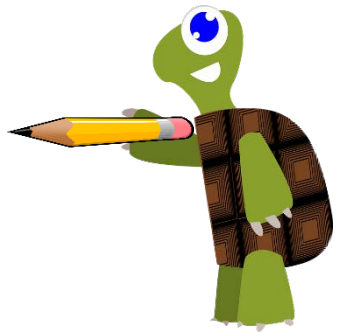
IDLE

Command window and editor window

- Simple code completion
- Auto-indentation

Turtle Graphics

- Instantiating classes
- Calling methods



Class Browser, Path Browser

PyCharm

Project-based, fully-integrated

Professional and Community Editions

Code Completion

Code Refactoring

Visual Debugger

In the Next Module ...

Managing the file system with python