Science & Technology Facilities Council | UK Research and Innovation | NERC SCIENCE OF THE ENVIRONMENT

Python introductions

Thanks to all contributors:

Alison Pamment, Sam Pepler, Ag Stephens, Stephen Pascoe, Matt Pryor, Anabelle Guillory, Graham Parton, Esther Conway, Wendy Garland, Alan Iwi, Matt Pritchard and Tommy Godfrey.

BEFORE WE START (1)

We cannot teach you Python in a day...

...but hopefully we can show you that it is:

- Human-readable (relatively)
- Useful (even if you only know a bit)
- Flexible (you can use it in many places/ways)

BEFORE WE START (2)

Teaching materials courtesy of:

http://www.software-carpentry.org/

Overview

- Introduction – why we recommend Python
- Basics and control flow
- Lists, tuples and slicing
- Input/output
- Strings and text processing
- Functions, libraries and scripts
- Error handling and logging
- Sets, dictionaries and OOP

What is Python?

- A simple interpreted language
- Very human-readable with clean syntax
- Making it a very good "learn to programme" language

Why do we recommend Python?

- It is **open source** and **free**
- It is **cross-platform** (including Windows)
- It can be used for simple scripting through to writing full-blown complex applications
- Many libraries/tools to tackle all kinds of problems
- In the environmental science community it continues to grow in popularity...so **we can share code**!

Python version?

Python has multiple personalities!

- **Python 3** – new; standard;

  Most people updating old code to standard.

  Used in this course!

- **Python 2.6+** – old; established;

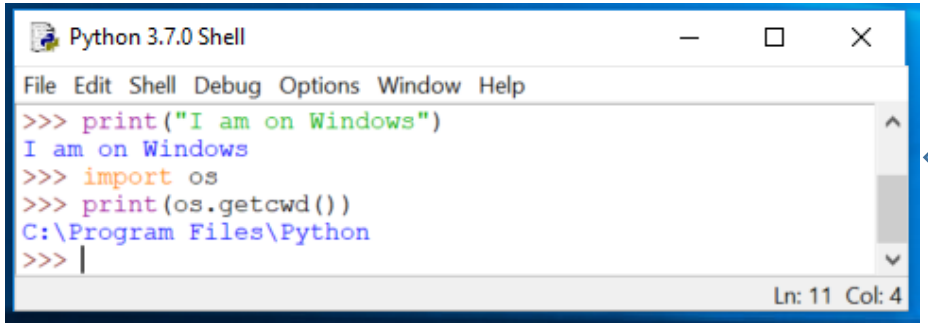
  Nearing end of life.

But what changed?

E.g. print changed from a statement to a function...

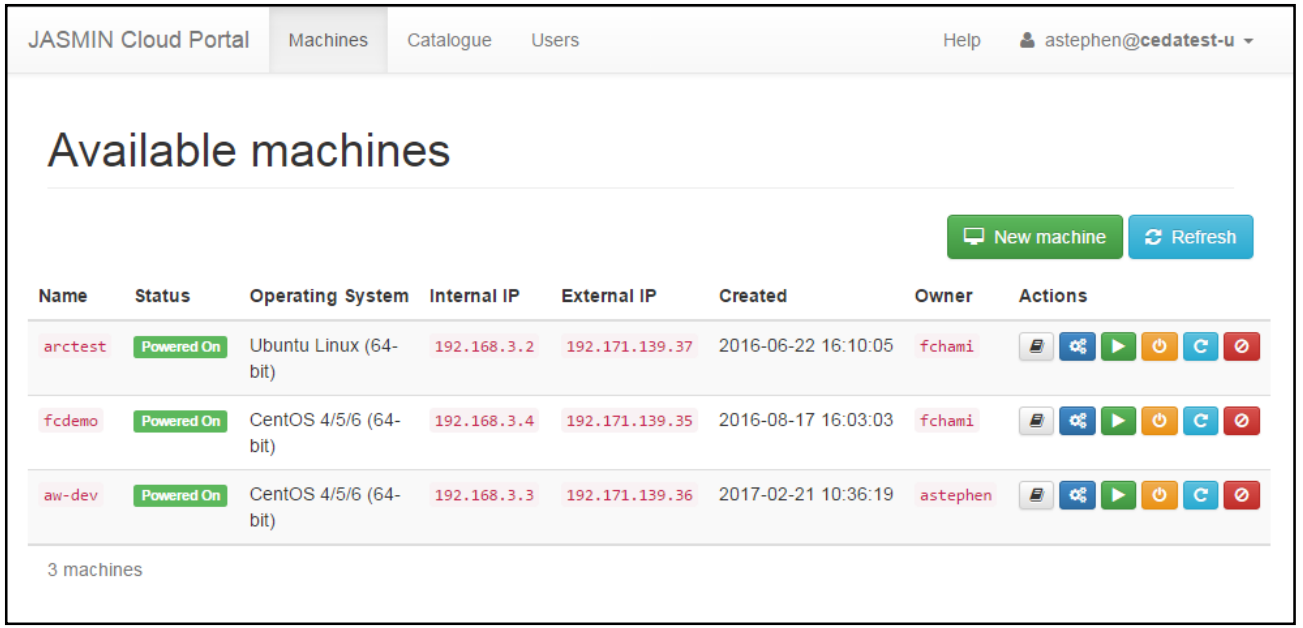
>>> **print "hello"** # in Python 2.* to...
>>> **print ("hello")** # in Python 3.*

What can you do with python?



Interactive "shell": allows quick learning/testing/use.

Web-programming: frameworks make this easy.



What can you do with python?

Iris v2.1

A powerful, format-agnostic, and community-driven Python library for analysing and visualising Earth science data.

home | examples | gallery | contents |

Iris user guide

How to use the user guide

If you are reading this user guide for the first time it is strongly recommended that you read the user guide fully before experimenting with your own data files.

Much of the content has supplementary links to the reference documentation; you will not need to follow these links in order to understand the guide but they may serve as a useful reference for future exploration.

Since later pages depend on earlier ones, try reading this user guide sequentially using the `next` and `previous` links.

User guide table of contents

- 1. Introduction
  - 1.1. Iris data structures
  - 1.2. Cubes in practice
- 2. Loading Iris cubes
  - 2.1. Loading multiple files
  - 2.2. Lazy loading
  - 2.3. Constrained loading
  - 2.4. Strict loading
- 3. Saving Iris cubes
  - 3.1. Controlling the save process
  - 3.2. Customising the save process
  - 3.3. Bespoke Saver
- 4. Navigating a cube
  - 4.1. Cube string representations
  - 4.2. Working with cubes
  - 4.3. Accessing coordinates on the cube
  - 4.4. Adding metadata to a cube
  - 4.5. Adding and removing metadata to the cube at load time

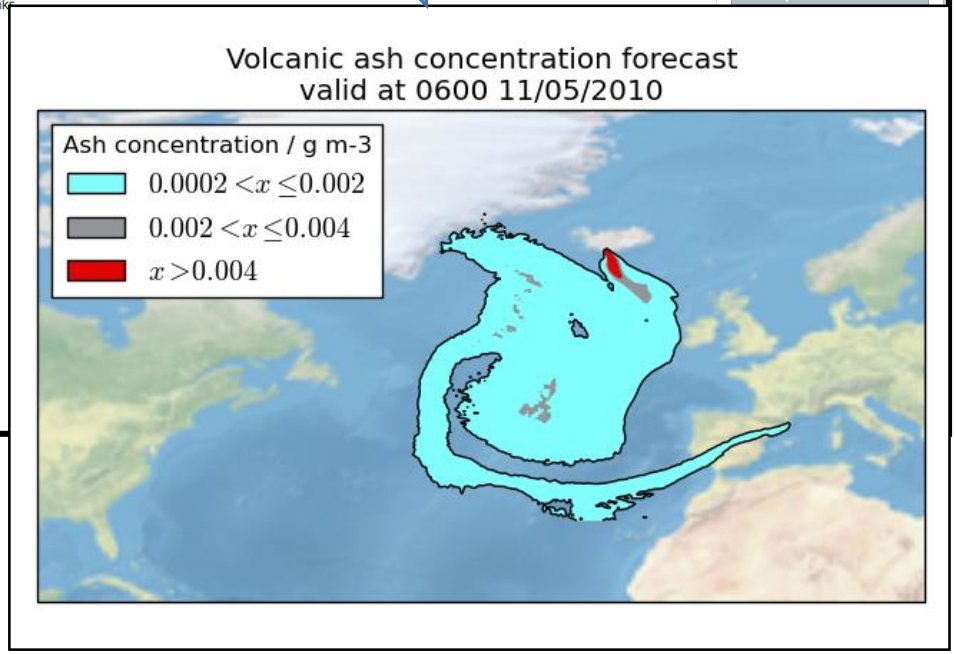
Fork me on GitHub

Open source packages for data analysis and visualisation.

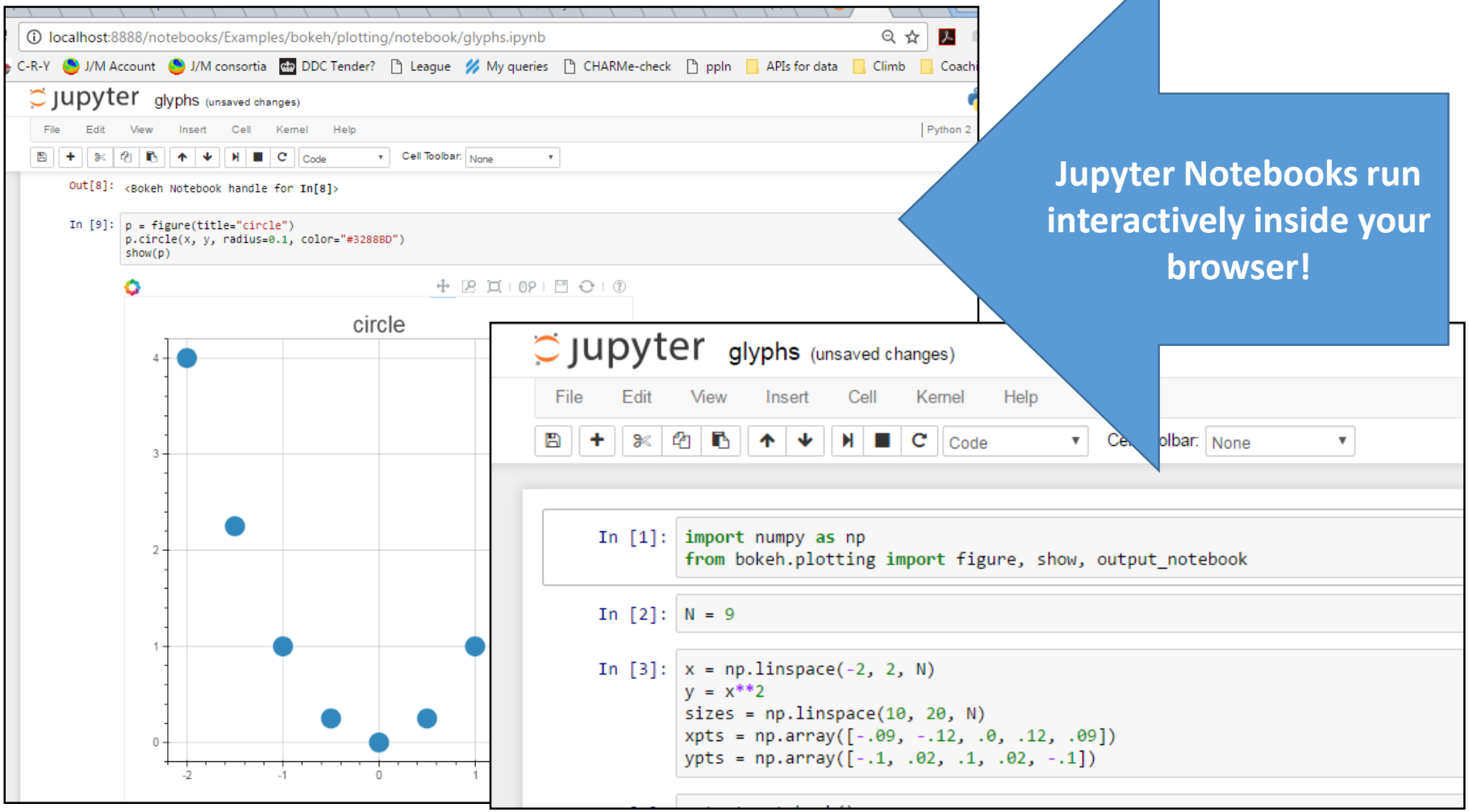
Previous topic

Installing Iris

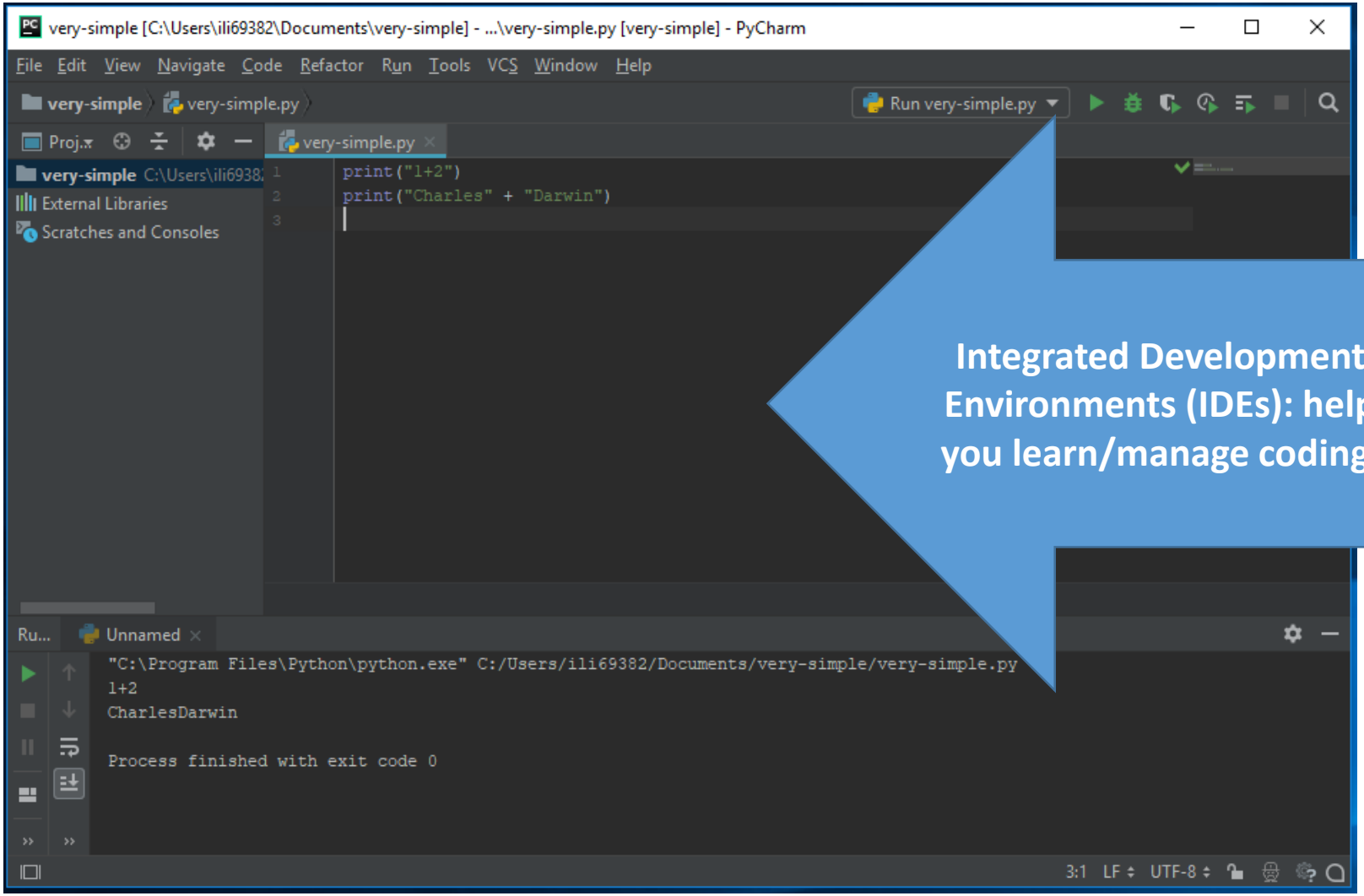
Next topic



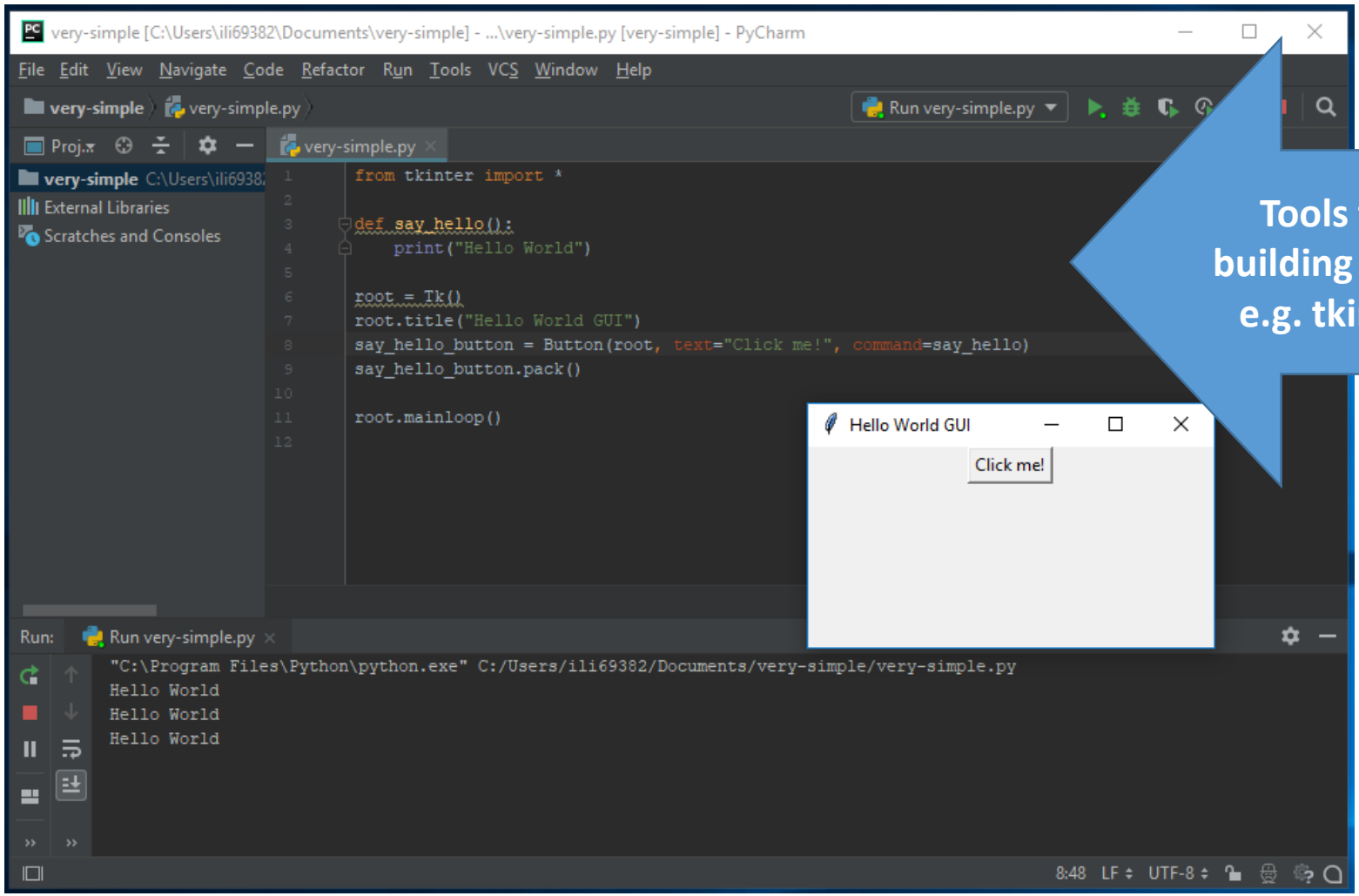
What can you do with python?



What can you do with python?



What can you do with python?



Let's get to work...