Viktor Porvaznik | CGI Deutschland B.V. & Co. KG
| July 26, 2021

Python Advanced Training

Before training

- Read and understand Chapters 0-7 and 9 of the https://automatetheboringstuff.com/
- Make sure that you have an account and repo.it and can access https://replit.com/join/yusotaojdv-porvik
- Installed Python 3.9 from https://www.python.org/downloads/ and can launch IDLE

Before training

- Have some terminal e.g. Windows Terminal or Tilix
- Have some epic IDE or a text editor e.g. Notepad++ or Gvim
- Have at hand the cheatsheet for in case you need it

Chapter structure

- Introduction and objective
- Hands-on including Q&A
- Code exercises

Agenda

Agenda

- Introduction

Agenda

- Introduction
- Module 1: Binary reader

Agenda

- Introduction
- Module 1: Binary reader
- Module 2: Build and packaging

Agenda

- Introduction
- Module 1: Binary reader
- Module 2: Build and packaging
- Module 3: More on packaging and distribution

Agenda

- Introduction
- Module 1: Binary reader
- Module 2: Build and packaging
- Module 3: More on packaging and distribution
- Module 4: Web applications and templating

Agenda

- Introduction
- Module 1: Binary reader
- Module 2: Build and packaging
- Module 3: More on packaging and distribution
- Module 4: Web applications and templating
- Module 5: Web scraping

Introduction

Python environments

- System
- Virtualenv - https://docs.python.org/3/tutorial/venv.html
- Conda - https://conda.io/en/latest/
- Pipenv

Introduction

Common environment deployment targets

- Native (Linux, Windows, etc.)
- Containers (Docker, Lxc, etc.)
- Remote (SSH)
- WSL
- Jupyter notebook

1 Binary reader

Goal is to create *binary_reader* application capable of reading binary documents and exporting their contents based on configurable structure.

- argparse - https://docs.python.org/3/howto/argpar
- lxml - https://lxml.de/1.3/index.html#introduction
- etree - https://docs.python.org/3/library/xml.etree.elemen

1 Binary reader

- struct - https://docs.python.org/3/library/struct.htm
- openpyxl - https://openpyxl.readthedocs.io/en/stal
- More on decorators - https://wiki.python.org/moin/PythonDecoratorLibr
- INI parser - https://docs.python.org/3/library/confi
- unittest - https://docs.python.org/3/library/unittest

1 Exercise

- Implement INI, JSON or XML without lxml config parser.
- Add unit tests check some part of the application.

2 Build and packaging

Goal is to build package with the *binary_reader* application and deploy it.

- General info about packaging - https://packaging.python.org/tutorials/packaging-projects/#

2 Build and packaging

- Packaging using `setuptools` module - https://setuptools.readthedocs.io/en/latest/userguide/quickstart.htm from-setup-py-to-setup-cfg
- The main package configuration stored in `setup.c` https://setuptools.readthedocs.io/en/latest/userguide/declarative_co highlight=data_files#options
- Building the package using `build` module - https://py build.readthedocs.io/en/latest/index.html

2 Exercise

Build the package and upload it into the main repository.

3 More on packaging and distribution

Goal is to further enhance our package by adding a C++ (non-Python source file) library and distribution it together with the package.

- PyPi classifiers - https://pypi.org/classifiers/
- ctypes - https://docs.python.org/3/library/ctypes.html

3 More on packaging and distribution

- `package_data` - defines files related to the python package. eg. documentation, static image files, configurations
- `data_files` - defines files that will be installed system-wise, not in site-package directory. eg. desktop icons, fonts

4 Web applications and templating

Goal is to create a simple webserver serving built packages and exposing a REST API.

- jinja - https://jinja.palletsprojects.com/en/3.0.x/templates
- Flask - https://flask.palletsprojects.com/en/2.0.x/

4 Exercise

- Implement the jinja template that allows you to submit test data file;
- Implement the ability to download XLSX exports from binary reader app through the REST API.

5 Web scraping with `requests` and `selenium`

Goal is to create and learn how to access and parse webpages and other text based APIs.

- requests - https://docs.python-requests.org/en/ma
- selenium - https://www.selenium.dev/selenium/docs/api/py/in 0

5 Exercise

Implement unit tests verifying our REST API from previous webserver application.

Extra topics

- Thread-based parallelism - https://docs.python.org/3/library/threading.html
- Process-based parallelism - https://docs.python.org/3/library/multiprocessing.html#module-multiprocessing

Excluded stuff

- Exceptions
- Lambdas
- Language bindings

Q&A

Questions?

What will happen next?

A short errata explaining topics that were explained incompletely / incorrectly or ambiguously, corrected materials and a commented source code will be provided withing this week

Feedback

Please fill-out the feedback form at
https://forms.gle/9xrkEojnRyfxxRbt8

Dive into further reading

Apart from links inside individual modules / chapters provided throughout the presentation I strongly recommend to follow-up with following materials:

- https://diveintopython3.net/

Thank you again for the participation in the training