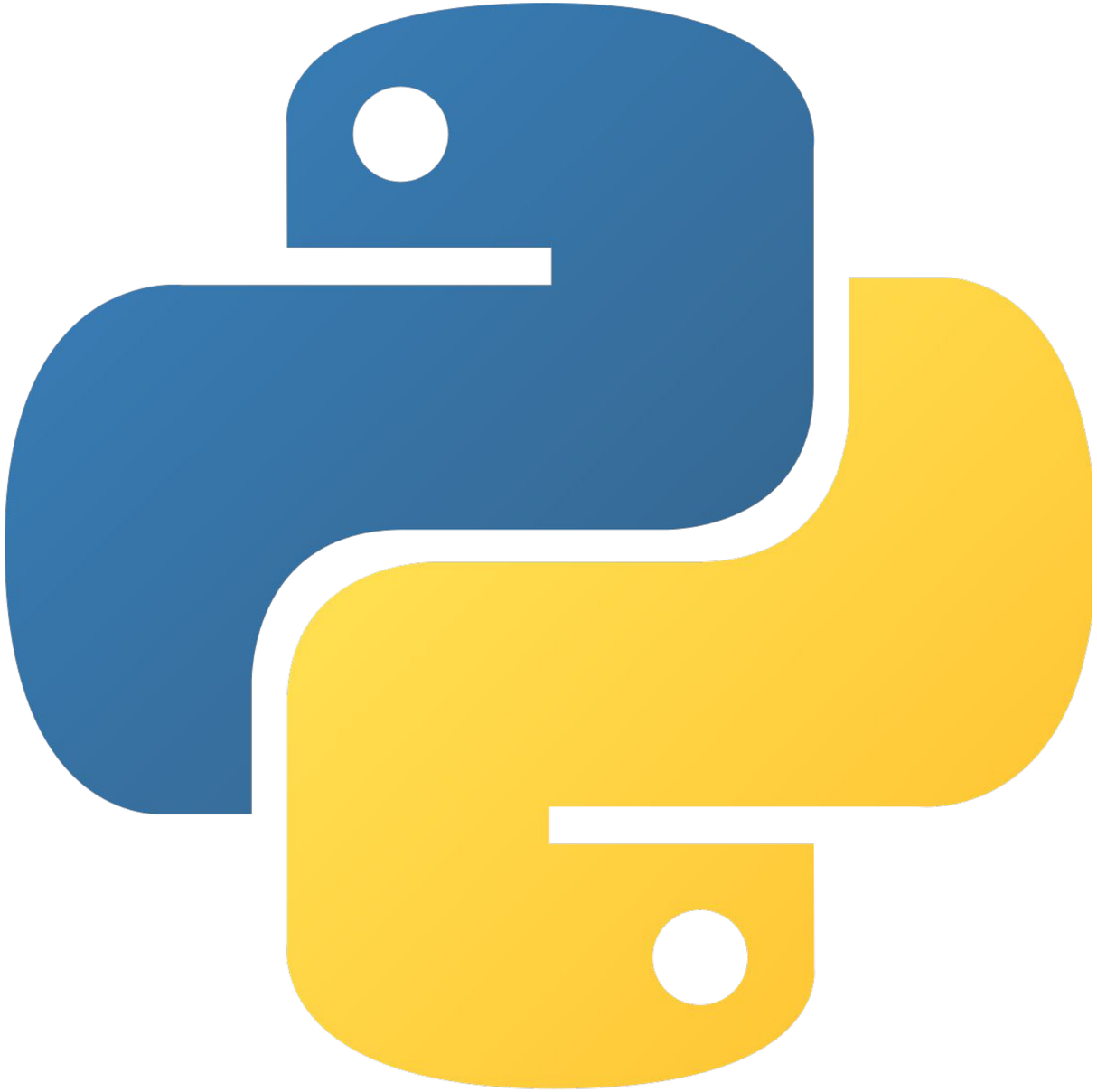

© All rights reserved. www.keepcoding.io

Alberto Vara

- Coordinador de equipos Python en Paradigma Digital
- Más de 5 años de experiencia en Python
- 11 años de experiencia en mundo web
- Más de 100 Repositorio en Github (5 o 6 útiles)
- Coorganizador del Meetup Python Madrid

© All rights reserved. www.keepcoding.io

Índice, día 1. Python

- Repaso rápido
- Preparación de entorno
    * Entorno virtual
    * Dependencias
    * Linterns
    * Tests
    * Otros: bandit, safety
- Decoradores
- PAUSA - 11:00
- List comprehension e iteradores
- Clases
- Métodos mágicos

© All rights reserved. www.keepcoding.io

Repaso rápido

© All rights reserved. www.keepcoding.io

Preparación del Entorno

© All rights reserved. www.keepcoding.io

Virtualenv

```
virtualenv --python=python3.6 venv
source venv/bin/activate
```

© All rights reserved. www.keepcoding.io

Virtualenv wrapper

```
pip install virtualenvwrapper
...
$ export WORKON_HOME=~/envs
$ mkdir -p $WORKON_HOME
$ source /usr/local/bin/virtualenvwrapper.sh
$ mkvirtualenv env1
$ workon env1
```

© All rights reserved. www.keepcoding.io

Pipenv

```
pip install pipenv
...
$ pipenv --python 3.7
$ pipenv shell
```

© All rights reserved. www.keepcoding.io

Pylint

- Detecta errores de estilo fijados en Pep8
  - Número de espacios
  - Nombres de las variables
  - Longitud de líneas
  - etc
- Librerías que no se usan y variables que no se utilizan
- Código duplicado (esto funciona regulero)
- Máximo de parámetros
- Mide la complejidad ciclomática

© All rights reserved. www.keepcoding.io

Pylint

- Hindice, dia 1. Python. Repaso rápido . Preparación de entorno * Entorno virtual * Dependencias * Linterns * Tests * Otro: bandit, safety Decoradores PAUSA - 11:00 List comprehension e iteradores Clasesh y metaprogramación Métds mágicos pip y Distribución de paquetes y librerías Testing Librerías típica

© All rights reserved. www.keepcoding.io

Pylint

```
pipenv install pylint
...
pylint --generate-rcfile > .pylintrc

pylint curso
************* Module curso-python-practico.curso.example01
curso/example01.py:1:0: C0114: Missing module docstring (missing-module-docstring)
....
************* Module curso-python-practico.curso.example02
curso/example02.py:10:0: C0301: Line too long (119/100) (line-too-long)
curso/example02.py:1:0: C0114: Missing module docstring (missing-module-docstring)
curso/example02.py:4:0: C0116: Missing function or method docstring
(missing-function-docstring)
```

© All rights reserved. www.keepcoding.io

Flake8

- Compuesto de varias herramientas en 1:
  - PyFlakes:
    - Número de espacios
    - Nombres de las variables
    - Longitud de líneas
    - etc
  - pycodestyle: Detecta errores de estilo fijados en Pep8
  - McCabe complexity checker: Mide la complejidad ciclomática

- Más rápido que Pylint, menos opciones

© All rights reserved. www.keepcoding.io

Flake8

```
pipenv install flake8
...
flake8 curso
curso/example02.py:10:80: E501 line too long (119 > 79 characters)
```

© All rights reserved. www.keepcoding.io

Safety

- Safety se encarga de analizar todas las dependencias de tu proyecto en busca de vulnerabilidades de seguridad conocidas.

- Se puede utilizar sin ningún coste si se conecta a Safety DB, o por el contrario con una suscripción mensual de $99/mes para organizaciones en pyup.io. La diferencia en que Safety DB solo se actualiza una vez al mes con las vulnerabilidades.

© All rights reserved. www.keepcoding.io

Safety

```
pipenv install safety
...
safety check -r requirements.txt
╒═══════════════════════════════════════════════════════════════════════════════╕
│ REPORT                                                                        │
│ checked 1 packages, using default DB                                          │
╞═══════════════╤═══════════╤═══════════╤═══════╕
│ package       │ installed │ affected  │ ID    │
╞═══════════════╪═══════════╪═══════════╪═══════╡
│ requests      │ 1.19.0    │ <2.3.0    │ 26101 │
│ requests      │ 1.19.0    │ <2.6.0    │ 26102 │
│ requests      │ 1.19.0    │ <=2.19.1  │ 36546 │
╘═══════════════╧═══════════╧═══════════╧═══════╛
```

© All rights reserved. www.keepcoding.io

Bandit

- Bandit busca problemas de seguridad comunes en Python, para ello analiza cada uno de los ficheros que se le indique mediante unos tests. Aunque por defecto lanza unos tests, qué tests se lanzan se puede configurar y agrupar en perfiles, siendo útil cuando el código tiene vulnerabilidades conocidas que para un proyecto en concreto no se consideren un problema e interese ignorarlas.

© All rights reserved. www.keepcoding.io

Bandit

```
pipenv install bandit
...
bandit -r * -x venv,src -ii -l -n 3

Test results:
    No issues identified.

Code scanned:
    Total lines of code: 1421
    Total lines skipped (#nosec): 0
```

© All rights reserved. www.keepcoding.io

Decoradores

© All rights reserved. www.keepcoding.io

List comprehensions

- Forma rápida e idiomática de generar listas
- Permite aplicar funciones y filtrar los elementos que genera (Map y Filter)
- Pueden llegar a ser complejas de usar, por lo que ante la duda usar un bucle for

© All rights reserved. www.keepcoding.io

Ejercicio 1

En 30 Líneas, resolver este problema:
Una distribuidora puede fabricar camisetas de los colores ["black", "white", "red"] y de las tallas ["S", "M", "L"].
Las tiendas le solicitan cada semana que tallas y que colores quieren para el siguiente pedido.
Crear una (o varias) funciones que listen para cada tienda las combinaciones de tallas y camisetas que tendrá que mandarles a cada una en su pedido tale que:

```
=============
Shop 1
=============

*Talla: M. Color: white
*Talla: L. Color: white
*Talla: M. Color: red
*Talla: L. Color: red

=============
Shop 2
=============

*Talla: S. Color: black
*Talla: M. Color: black
*Talla: S. Color: white
*Talla: M. Color: white
```

© All rights reserved. www.keepcoding.io

Classes

© All rights reserved. www.keepcoding.io

Clases Abstractas

© All rights reserved. www.keepcoding.io

Mixins

- Es un patrón para trabajar con herencia múltiple
- Los mixins son clases que encapsulan una serie de propiedades y comportamientos muy concretos
- No son útiles en sí mismos, están hechos para que se sean heredados por otras clases

© All rights reserved. www.keepcoding.io

Duck typing

- Python es muy flexible. Muy muy flexible.
- Permite, entre otras cosas, asignar nuevos atributos a objetos ya creados
- O… ¡¡Cambiar la metaclase de un objeto ya creado!!

© All rights reserved. www.keepcoding.io

Creación de librerías

- Generar una librería o un artefacto con Python es un proceso bastante simple. Basta con generar un fichero setup.py en la raíz de tu proyecto con esta definición

```
from setuptools import find_packages, setup

setup(name='myproject',
      version='1.0',
      packages=find_packages())
```

© All rights reserved. www.keepcoding.io

Creación de librerías

- Importante saber que el nombre que le damos dentro de nuestro setup.py y el nombre de la carpeta no tienen porqué ser el mismo. Puede que tu librería de conflicto, o por nomenclatura quieres darle otro nombre, por ejemplo, en nuestro setup.py podemos ponerle name='paradigma_genericforms' pero si a la carpeta la llamamos nada más genericforms cuando la importemos en nuestros scripts será llamada con import genericforms

```
from setuptools import find_packages, setup

setup(name='myproject',
      version='1.0',
      packages=find_packages())
```

© All rights reserved. www.keepcoding.io

Creación de librerías

- Para desplegar nuestras librerías en el repositorio oficial de Pypi, antiguamente, nos bastaba con tener una cuenta en Pypi y configurar nuestras credenciales en ~/.pypirc tal que:

```
# ~/.pypirc
[distutils]
index-servers =
  pypi

[pypi]
username:<your_pypi_username>
password:<your_pypi_passwd>

--

$ python setup.py sdist # genera el artefacto
$ twine upload -r pypi dist/* # sube el artefacto
```

© All rights reserved. www.keepcoding.io

Creación de librerías

- Servidor para probar en local:

```
htpasswd -sc .htpasswd pip
docker run -v $(pwd)/.htpasswd:/data/.htpasswd -p 8080:8080 pypiserver/pypiserver:latest -P .htpasswd packages
---

[distutils]
index-servers =
  pypi
  local

[pypi]
username:<your_pypi_username>
password:<your_pypi_passwd>

[local]
repository: http://localhost:8080
username: pip
password: local
---
python setup.py sdist upload -r local
twine upload -r local dist/*
```

© All rights reserved. www.keepcoding.io

Testing

Convenciones:

- Los tests se guardan en el directorio raíz bajo la carpeta “tests”
- Los archivos de tests se nombran tal que: test_*.py
- Las clases y los métodos que recogen un test empiezan por Test o test_

© All rights reserved. www.keepcoding.io

Testing

Tres librerías importantes:

- unittest y mock: Utilidades en la librería estándar de Python. Son fáciles de usar pero de características muy simples.
- pytest y pytest-mocker: Librería de testing basada en fixtures. Permite ejecutar los tests en paralelo, parametrizar tests…
- Nosetests: La “siguiente iteración” de unittest. Una suite más completa que permite ejecutar los test en paralelo entre otras cosas

© All rights reserved. www.keepcoding.io

Testing

Mocks:

- Objetos dummy que devuelven lo que nosotros queremos que devuelvan.
- Usados para mockear “terceros”. Un tercero puede ser o una llamada a una api externa o una llamada a una de nuestras funciones. Recordemos la filosofía de los tests unitarios.

© All rights reserved. www.keepcoding.io

Testing

Patch:

- Los mocks en python pueden usarse como objeto de por sí o patcheando una función ya existente.
- Patch: Reemplaza código ya existente por nuestros mocks, en tiempo de ejecución.
- Patcheamos la referencia de la función/objeto que queramos mockear, no dónde lo usamos.

© All rights reserved. www.keepcoding.io

Testing

Pytest:

- Librería externa de testing bastante avanzada.
- Tests en funciones, no tanto en clases (aunque también se puede)
- Se basa sobre todo en fixtures, que se inyectan en cada tests según el test lo pida.
- También permite la parametrización de tests, la ejecución de tests de forma paralela, tener setUp/tearDown para una misma sesión de tests....

© All rights reserved. www.keepcoding.io