Sistemas de Gestion de Calidad

EPET N°12

PEP 20

El Zen de Python

Calidad en Python
Documentación

Ing. Eduardo Kunysz

Temas

- PEP-020 Zen of Python
- PEP-257 Docstring
- Sphinx Style

The Zen of Python (PEP-0020)

- Lo bello es mejor que lo feo.
- Lo explícito es mejor que lo implícito.
- Lo simple es mejor que lo complejo.
- Lo complejo es mejor que lo complicado.
- Lo plano es mejor que lo anidado.
- Lo disperso es mejor que lo denso.
- La legibilidad cuenta.
- Los casos especiales no son lo suficientemente especiales como para romper las reglas.
- Aunque la practicidad supera a la pureza.

The Zen of Python (PEP-0020)

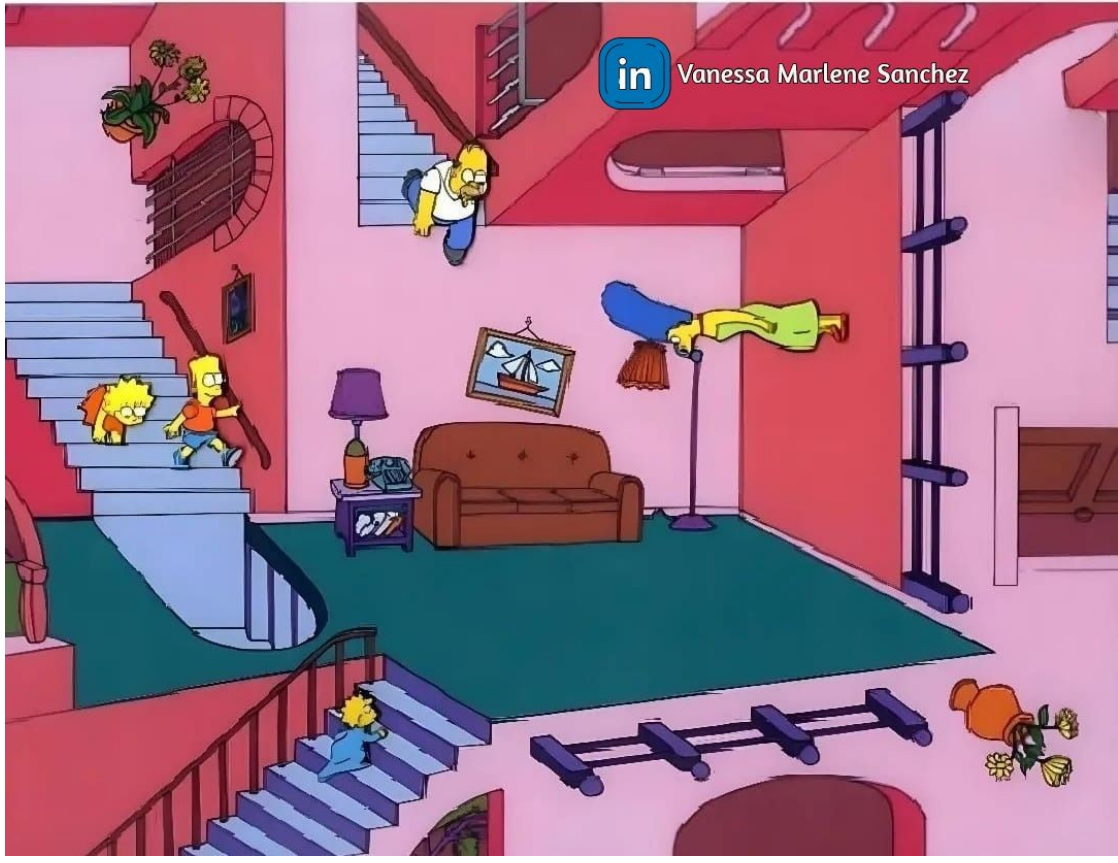
- Los errores nunca deberían pasar en silencio.
- A menos que sean explícitamente silenciados.
- Ante la ambigüedad, rechaza la tentación de adivinar.
- Debería haber una —y preferiblemente solo una— manera obvia de hacerlo.
- Aunque esa manera puede no ser obvia al principio, a menos que seas holandés.
- Ahora es mejor que nunca.
- Aunque nunca es a menudo mejor que justo ahora.
- Si la implementación es difícil de explicar, es una mala idea.
- Si la implementación es fácil de explicar, puede que sea una buena idea.
- Los espacios de nombres son una gran idea — ¡usemos más de esos!

Temas

Estilo de Documentación con PEP-257

- La documentación suele ser el punto menos atendido. Es código no interpretado por la máquina. Por lo que permite la libre implementación, lo que lleva a la libre interpretación.
-
- Python nombra DocString a su documentación.

El programador/a intentando entender el código de un proyecto mal documentado



Temas

PEP-257 nos sugiere:

- Qué componentes deben tener DocString.
- Cómo implementar DocString de una línea.
- Cómo implementar DocString de múltiples líneas.
- Indentación de DocStrings.

```
def string_reverse(str1):
    '''
    Returns the reversed String.

    Parameters:
        str1 (str):The string which is to be reversed.

    Returns:
        reverse(str1):The string which gets reversed.
    '''

    reverse_str1 = ''
    i = len(str1)
    while i > 0:
        reverse_str1 += str1[i - 1]
        i = i- 1
    return reverse_str1
```

Sphinx Style

- Sphinx es una herramienta que convierte el código fuente y los comentarios en documentos legibles en varios formatos (HTML, PDF, etc.).
- Usos comunes: Documentación de proyectos Python, generando documentación de API de forma automática a partir del código.

Instalación y Configuración

```
pip install sphinx
```

o si usamos el gestor conda (de Anaconda):

```
conda install sphinx
```

Configuración Inicial:

Navega a la carpeta raíz de tu proyecto.

Ejecuta:

```
Sphinx-quickstart     en windows Sphinx-quickstart.exe
```

Con utilizar las opciones por defecto se logra una buena configuración.

Estructura Básica de Sphinx

Archivos importantes

conf.py: Configuración de Sphinx.

index.rst: El archivo principal de la documentación.

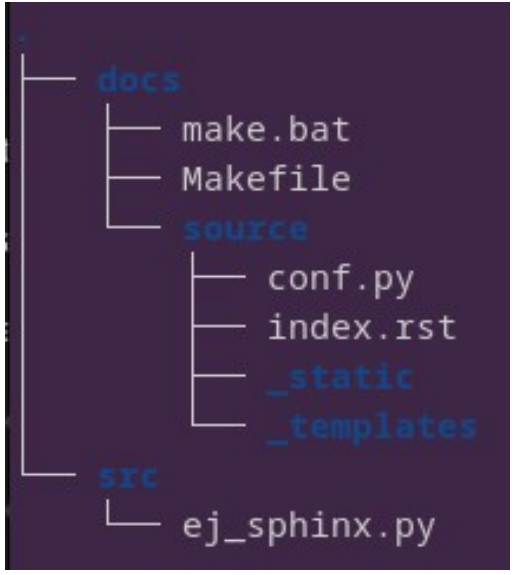
reStructuredText

Directivas y Roles

Directivas: Instrucciones especiales en los archivos .rst (por ejemplo, .. automodule::).

Roles: Etiquetas especiales para referenciar partes del código.

Estructura recomendada



Documentación con código

```python
def mi_funcion(param1, param2):
    """
    Descripción breve de la función.

    :param param1: Descripción del primer parámetro.
    :type param1: str
    :param param2: Descripción del segundo parámetro.
    :type param2: int
    :return: Descripción del valor de retorno.
    :rtype: bool
    """
    return True
```

Archivo config.py

Importar librerias y configuraciones de ubicación de los fuentes:

```
import os
import sys
sys.path.insert(0, os.path.abspath('../src'))
```

Autodoc: Extensión que genera documentación automáticamente:

```
   extensions = [
    'sphinx.ext.autodoc',
    'sphinx.ext.mathjax',
    'sphinx.ext.napoleon',
    'sphinx.ext.coverage',
    'sphinx.ext.doctest'
]
```

Archivo index.rst

El proyecto 'ReStructured Text es un lenguaje de marcas ligero creado para escribir textos con formato definido de manera cómoda y rápida. Es parte del proyecto Docutils dentro de la comunidad de Python, y es formalizado por el grupo Python Doc-SIG (Documentation Special Interest Group).

```
Welcome to Prueba's documentation!
==================================
.. toctree::
   :maxdepth: 2
   :caption: Contents:

   modules
```

Algún título

Directiva para índice

Profundidad de índice

Pone en este lugar el archivo modules.rst

```
Indices and tables
==================

* :ref:`genindex`
* :ref:`modindex`
* :ref:`search`
```

Archivo modules.rst

Crear este archivo y agregarle las directivas para incluir la documentción

```
Module Documentation
====================
.. automodule:: ej_sphinx
    :members:
```

Generar Documentación HTML

```
make html
```

o

```
make.bat html
```

El resultado se encontrará en la carpeta _build/html. Abrir **index.html** en un navegador.