



Buenas Prácticas de Programación

Lección 3: Documentación del código

Indice

Introducción.....	3
Alternativas para Sphinx.....	4
Ejemplos.....	5
Conclusiones.....	5

Introducción

En esta lección se trata de hallar un módulo en Python que permita realizar la documentación de un código de forma sencilla y eficiente.

Alternativas para Sphinx

Atendiendo a la tarea de buscar alternativas a sphinx para generar la documentación de los programas se ha hallado la herramienta de pydoc.

Ademas de la posibilidad de generar la documentación referente a un proyecto, el modulo cuenta con otras funciones interesantes:

- `pydoc <nombre>`

Imprime por pantalla la documentación del objeto especificado mediante <nombre>, este objeto puede ser un modulo, una función o un framework entre otros.

- `pydoc -k <key>`

Busca e imprime toda la información relacionada con la palabra clave key.

- `pydoc -b`

Lanza un servidor HTTP con la información de todos los módulos de la distribución actual de python.

- `pydoc <filename>`

Imprime por pantalla la documentación referente al archivo python <filename>.

- `pydoc -w <filename>`

Genera un html con la documentación del archivo <filename>

El módulo cuenta con otras funciones que no se tratan en este documento, unicamente se han mostrado ciertas funciones.

Otro modulo hallado como alternativa a sphynx es el módulo pycco. Este modulo ofrece mas versatilidad que el modulo anterior, a la hora de generar índices y títulos resulta mas cómodo y rápido ya que basta con añadir ciertos caracteres en el comentario como puede ser `=== Titulo ==` para añadir un titulo, `**texto**` para añadir negrita etc. Además, al generar el archivo html con pycco, este generará dos columnas, en la izquierda se mostrarán todas las aclaraciones y comentarios del código y a la derecha se mostrará el código sin estos comentarios.

Ejemplos

Ejemplo de documentación con pydoc:

```

Archivo Editar Ver Buscar Terminal Ayuda
Help on module pydoc_test:

NAME
pydoc_test

DESCRIPTION
Author: Oscar Gutierrez
Email: o.guty66@gmail.com
Python Version: 3.6.9
Version: 0.0.1

CLASSES
builtins.object
CSV

class CSV(builtins.object)
    Esta es la clase empleada para realizar un informe de gastos a partir de
    un archivo csv.

    Methods defined here:
    __init__(self, file)
        Initialize self. See help(type(self)) for accurate signature.

    clean(self)
        Lee la información del CSV y la normaliza, convierte todos los datos a
        valores numericos si puede y si no, les asigna el valor 0.

    gen_graph(self)
        Genera un grafico con la evolucion solicitada a lo largo del año, bien
        sea la evolucion de los gastos, la de los ingresos o la de los
        ahorros

    get_month_income(self)
        Genera un diccionario con la suma de los ingresos mensuales, almacena
        en una variable el total de ingresos anuales.

    get_month_save(self)
        Almacena en un diccionario el valance mensual.

    get_month_spend(self)
        Genera un diccionario con la suma de los gastos mensuales, almacena en
        una variable el total de gastos anuales y calcula la media de gasto
        mensual.

    print_info(self)
        Da formato e imprime la información relevante del archivo CSV.

    select_case(self, case)
        Simulando una estructura Select/Case, genera la línea que se mostrará
        en el grafico segun la petición del usuario.

    select_title(self, case)
        Simulando una estructura Select/Case, Selecciona el titulo del grafico.

FUNCTIONS
run()
    Ejecución del código.

FILE
/home/elidas/Escritorio/Master/Buenas_Practicas/Lecion_3/Actividad_3/pydoc_test.py
(END)

```

Dada la extensión del archivo de documentación generado mediante pycco, se adjuntara con el archivo pdf el html.

Conclusiones

A termino personal, me ha resultado mas interesante el formato del html generado mediante pydoc ya que resulta mas visual.