

Buenas Practicas Python

Actividad 3

Por

Pablo Ujados Garcia

A1 – Herramienta de documentación alternativa a Sphinx

- Nombre :
 - o MkDocs ([MkDocs](#))
 - o MdDocString ([Overview - mkdocstrings](#))
 - o Material for MkDocs ([GitHub - squidfunk/mkdocs-material: Documentation that simply works](#))
- Modo de instalación
 - o `pip install mkdocs`
 - o `pip install mkdocstrings`
 - o `pip install mkdocstrings-python`
 - o `pip install mkdocs-material` (no es necesario, simplemente queda mas organizado)
- Procedimiento
 - o Creamos un entorno virtual (opcional)
 - o Instalamos las dependencias arriba descritas
 - o Nos aseguramos que tenemos los siguientes paquetes instalados

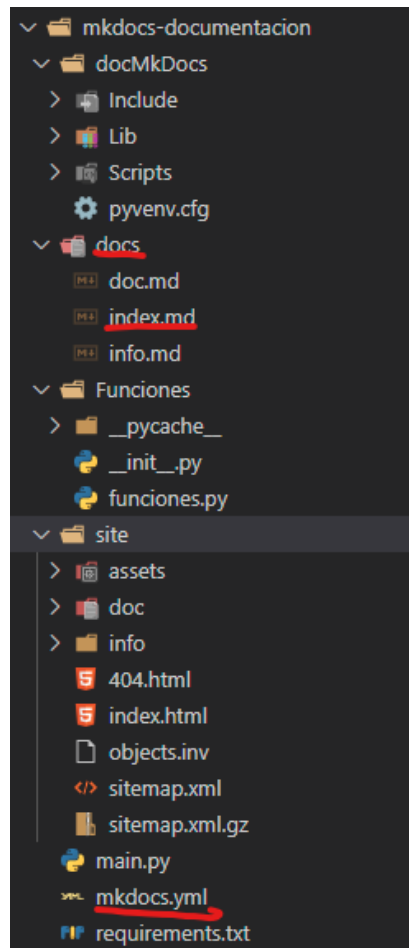
```
mkdocs 1.4.2
mkdocs-autorefs 0.4.1
mkdocs-material 8.5.11
mkdocs-material-extensions 1.1.1
mkdocstrings 0.19.1
mkdocstrings-python 0.8.2
```

Creacion de proyecto

- Mkddocs new [nombre del proyecto]

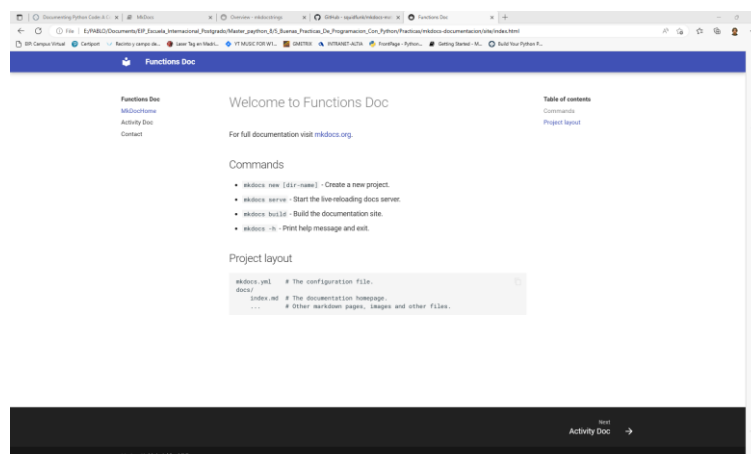
Cuando creamos el proyecto, nos aparece en nuestro entorno una carpeta con el nombre del proyecto y en su interior un archivo llamado “mkdocs.yml” y una carpeta “Docs” con un archivo en su interior llamado “index.md”

- mkdocs.yml: es el fichero de configuracion de MkDocs, podemos cambiar el tema de la página, la url de publicación, el índice de navegación entre páginas, los plugins utilizados etc.
- docs/index.md” será la página principal de la documentación



Para inicializar el servidor, nos posicionamos en la carpeta en la que este el fichero mkdocs.yml y ejecutamos “mkdocs serve”

Nos dirigimos a la IP y puerto en el que lo ha activado “<http://127.0.0.1:8000/>” y accederemos a nuestra documentación



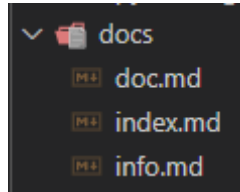
En este caso hemos creado un modulo de prueba con un par de funciones y lo hemos documentado con docstring al estilo google, ya que MkDoc, por defecto utiliza este modo de docstrings , admitiendo tambien Nupy y Sphinx

```
funciones.py X main.py
mkdocs-documentacion > funciones.py > Despidete
1 import mkdocs
2 from datetime import datetime
3
4 def Saludame(nombre: str, apellido: str, fechaNacimiento: str) -> int:
5
6     """Funcion que pide nombre, apellido y fecha de nacimiento
7     y devuelve un saludo diciendote los años que tienes.
8
9     Args:
10         nombre (str): Tu nombre
11         apellido (str): Tu apellido
12         fechaNacimiento (datetime): Tu fecha de nacimiento en formato dd/mm/yyyy
13
14     Return:
15         edad (int): Tu edad
16     """
17
18     fechaNac = datetime.strptime(fechaNacimiento, '%d/%m/%Y')
19
20     edad = datetime.today().date() - fechaNac.date()
21     edad = edad.days / 365
22     edad = int(edad)
23
24     print(f"Hola {nombre} {apellido}, me ha dicho que nació el {fechaNac}, con lo cual tienes {edad} años")
25
26     return edad
27
28 def Despidete(edad: int) -> None:
29     """Funcion que te despide segun la edad que tengas
30
31     Args:
32         edad (int): tu edad
33     """
34
```

Modificamos el "mkdocs.yml" para decir que va a utilizar el tema "Material" instalado en "mkdocs-material", el plugin "mkdocstrings" instalado en mkdocstrings y mkdocstrings-python y le ponemos las diferentes páginas que aparecerán en el panel de navegación con su nombre y referencia a la pagina

```
mkdocs.yml X funciones.py main.py
mkdocs-documentacion > mkdocs.yml
1 site_name: Functions Doc
2
3 theme:
4     name: "material"
5
6 plugins:
7     - mkdocstrings
8
9 nav:
10     - MkDocHome: index.md
11     - Activity Doc: doc.md
12     - Contact: info.md
13
14
```

Creamos los archivos que van a tener el contenido de las diferentes paginas dentro de la carpeta docs de nuestro proyecto



Como ya hemos dicho “index.md” es la página principal, en “doc.md” pondremos la documentación del desarrollo y en info.md la informacion del desarrollador, siendo estos sus respectivos contenidos

```
index.md 1 ## Welcome to Functions Doc
2
3 For full documentation visit [mkdocs.org]
4 (https://www.mkdocs.org).
5
6 ## Commands
7
8 * 'mkdocs new [dir-name]' - Create a new project.
9 * 'mkdocs serve' - Start the live-reloading docs
10 server.
11 * 'mkdocs build' - Build the documentation site.
12 * 'mkdocs -h' - Print help message and exit.
13
14 ## Project layout
15
16 mkdocs.yml # The configuration file.
17 docs/
18   index.md # The documentation homepage.
19   ...
20   # Other markdown pages, images
21   and other files.
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

info.md 1 ## Contacto
2
3
4 ## Nombre
5
6 - Pablo Ujados Garcia
7
8
9 ## Edad
10
11 - 29
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

doc.md 1 ## Functions Documentation
2
3 ::: Funciones.funciones
```

Como podemos ver “Index.md” e “info.md” no tiene más que la sintaxis estándar YAML, pero en doc.md vemos algo diferente, se trata de “:::Contenido.contenido” que gracias al plugin anteriormente puesto en el archivo de configuracion (mkdocstrings) lee los docstrings que tengamos en las diferentes funciones de nuestro modulo y los posiciona con un estilo concreto en el lugar que le hayamos dicho de la página, donde hayamos puesto este comando.

Si teniamos el servidor abierto, se nos ha tenido que ir actualizado poco a poco según introducíamos cambios y no habia ningun error, caso en el que el servidor no se inicializa, sino solo tendríamos que volver a lanzar el comando “mkdocs serve” en la carpeta en la que este el fichero “mkdocs.yml” y nos aparecería lo siguiente

```
(docMkDocs) PS E:\PABLO\Documents\EIP_Escuela_Internacional_Postgrado\Master_python_8\5_Buenas_Practicas_De_Programacion_Con_Python\Practicas\mkdocs-documentacion> mkdocs serve
INFO - Building documentation...
INFO - Cleaning site directory
INFO - Documentation built in 0.56 seconds
INFO - [21:30:35] Watching paths for changes: 'docs', 'mkdocs.yml'
INFO - [21:30:35] Serving on http://127.0.0.1:8000/
```

En el servidor con la documentación creada nos aparecería lo siguiente

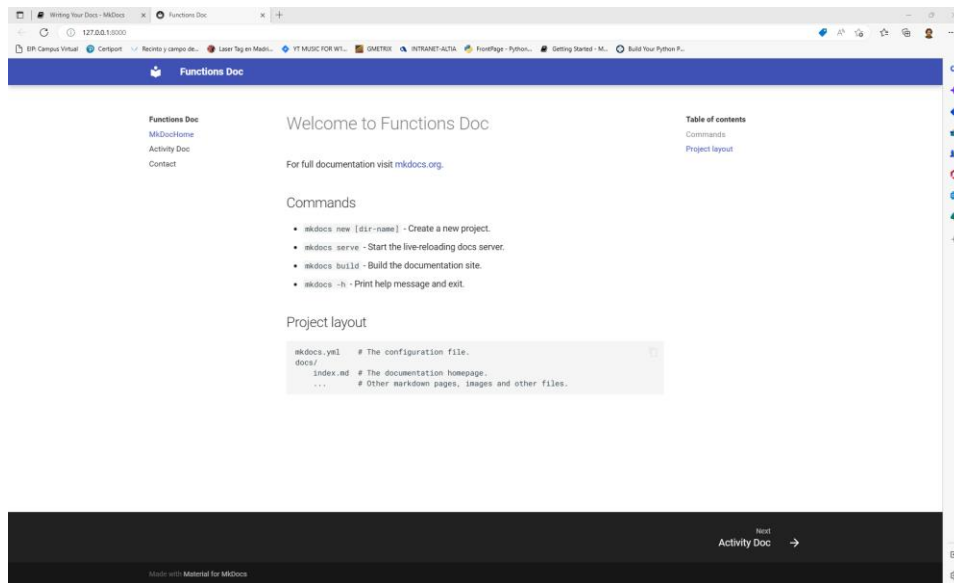


Ilustración 1 Página principal

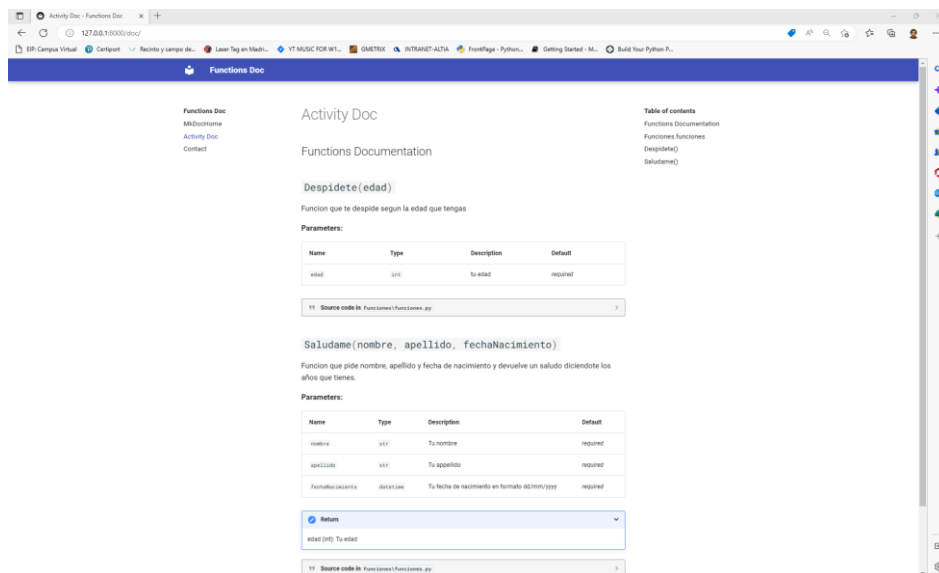


Ilustración 2 Página de documentación

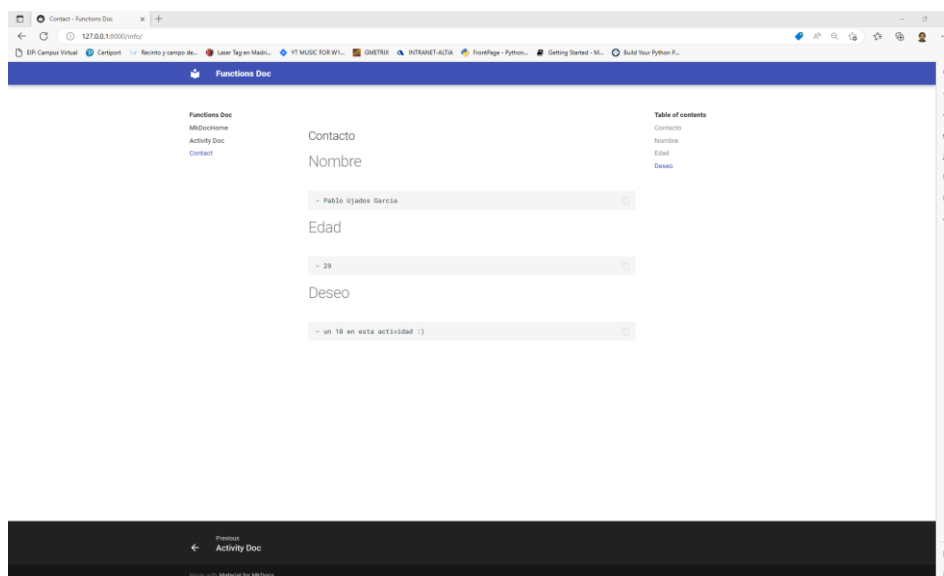
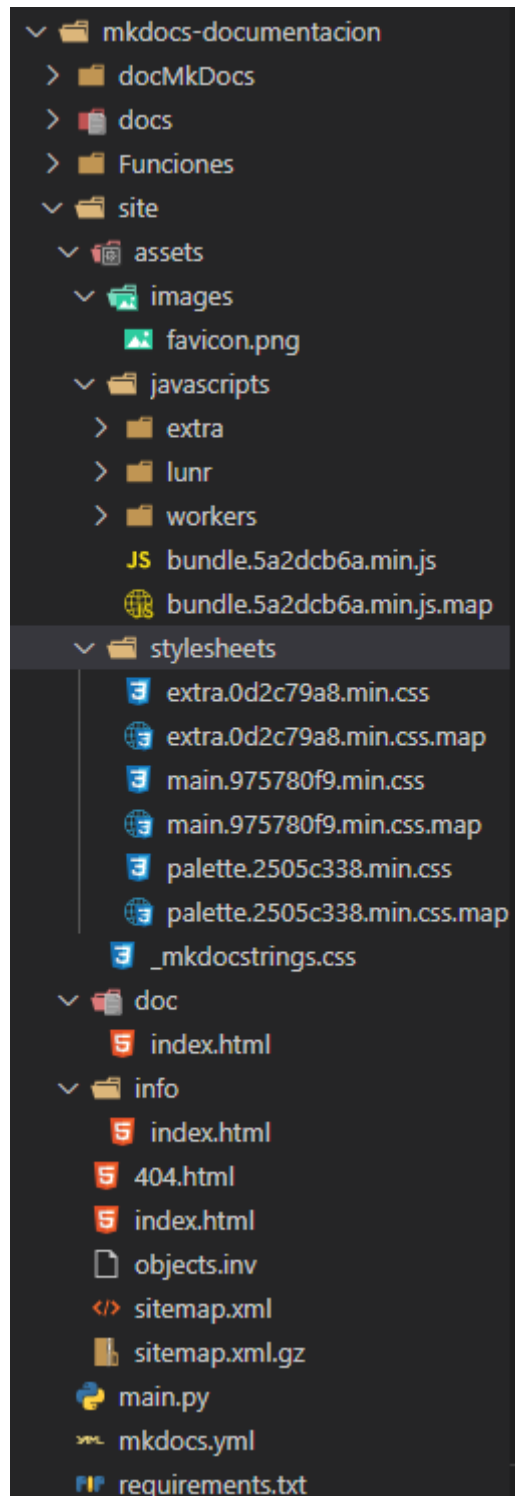


Ilustración 3 Página de contacto

Por último, ya solo nos faltaría un paso final, que sería la construcción de todo el árbol de directorios para la subida a un servidor real. Esto lo haríamos primero desactivando el servidor que como siempre en Python sería con CNTL + C en la consola Python. Lo siguiente que tendríamos que hacer es lanzar el siguiente comando

- `mkdocs build`

Con este comando se nos crea en nuestro proyecto una carpeta llamada “site” que contendrá todo lo necesario para desplegar nuestra página web en un servidor



Con esto habremos terminado nuestra documentación y será totalmente escalable a futuro.