

ACTIVIDAD PRÁCTICA LECCIÓN 3:

APARTADO 1:

1. Busque una herramienta alternativa a Sphinx para generar la documentación de código en Python.

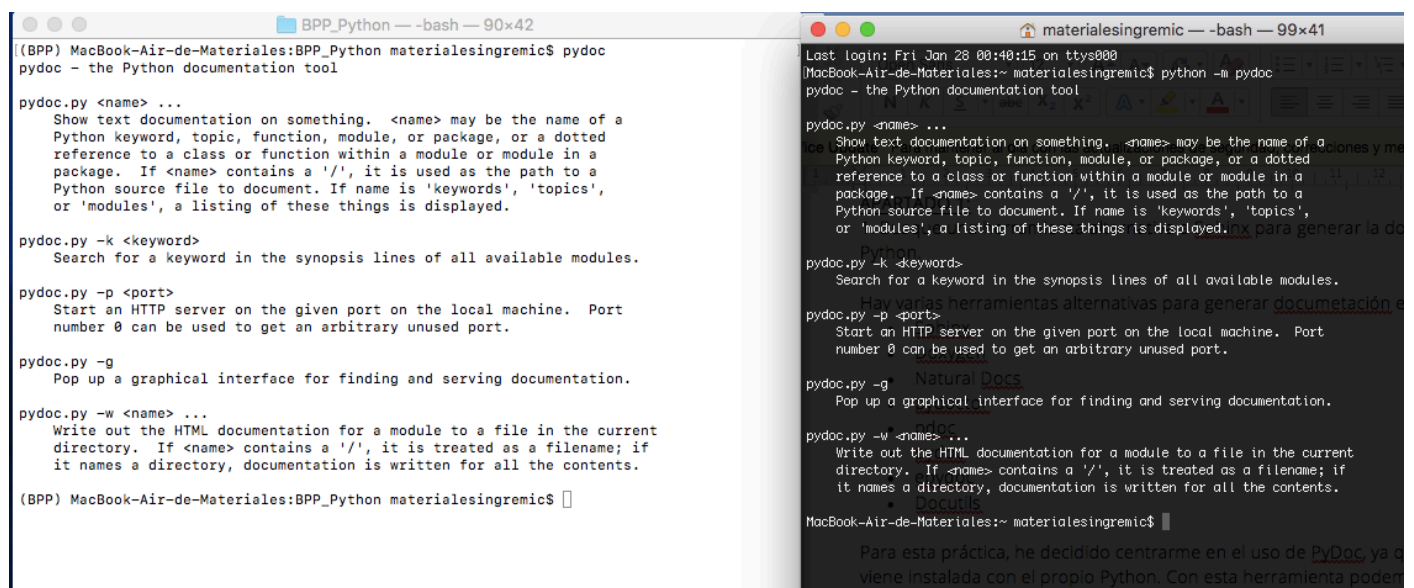
Hay varias herramientas alternativas para generar documentetación en Python:

- Sphinx
- Doxygen
- Natural Docs
- pydoctor
- pdoc
- pydoc
- epydoc
- Docutils

Para esta práctica, he decidido centrarme en el uso de PyDoc, ya que es una herramienta que viene instalada con el propio Python. Con esta herramienta podemos generar páginas web con nuestra documentación e incluso puede generar un servidor web en el cuál facilitar la documentación y el funcionamiento de nuestro código.

Lo primero que haremos, será repasar esta herramienta y su propia documentación.

Si escribimos en nuestra terminal el comando “python -m pydoc” o incluso “pydoc” como tal, veremos que tenemos una información previa de la herramienta.



```
BPP_Python --bash-- 90x42
((BPP) MacBook-Air-de-Materiales:BPP_Python materialesingremic$ pydoc
pydoc - the Python documentation tool

pydoc.py <name> ...
  Show text documentation on something. <name> may be the name of a
  Python keyword, topic, function, module, or package, or a dotted
  reference to a class or function within a module or module in a
  package. If <name> contains a '/', it is used as the path to a
  Python source file to document. If name is 'keywords', 'topics',
  or 'modules', a listing of these things is displayed.

pydoc.py -k <keyword>
  Search for a keyword in the synopsis lines of all available modules.

pydoc.py -p <port>
  Start an HTTP server on the given port on the local machine. Port
  number 0 can be used to get an arbitrary unused port.

pydoc.py -g
  Pop up a graphical interface for finding and serving documentation.

pydoc.py -w <name> ...
  Write out the HTML documentation for a module to a file in the current
  directory. If <name> contains a '/', it is treated as a filename; if
  it names a directory, documentation is written for all the contents.

((BPP) MacBook-Air-de-Materiales:BPP_Python materialesingremic$

materialesingremic --bash-- 99x41
Last login: Fri Jan 28 00:40:15 on ttys000
MacBook-Air-de-Materiales:~ materialesingremic$ python -m pydoc
pydoc - the Python documentation tool

pydoc.py <name> ...
  Show text documentation on something. <name> may be the name of a
  Python keyword, topic, function, module, or package, or a dotted
  reference to a class or function within a module or module in a
  package. If <name> contains a '/', it is used as the path to a
  Python source file to document. If name is 'keywords', 'topics',
  or 'modules', a listing of these things is displayed.

pydoc.py -k <keyword>
  Search for a keyword in the synopsis lines of all available modules.

pydoc.py -p <ports>
  Start an HTTP server on the given port on the local machine. Port
  number 0 can be used to get an arbitrary unused port.

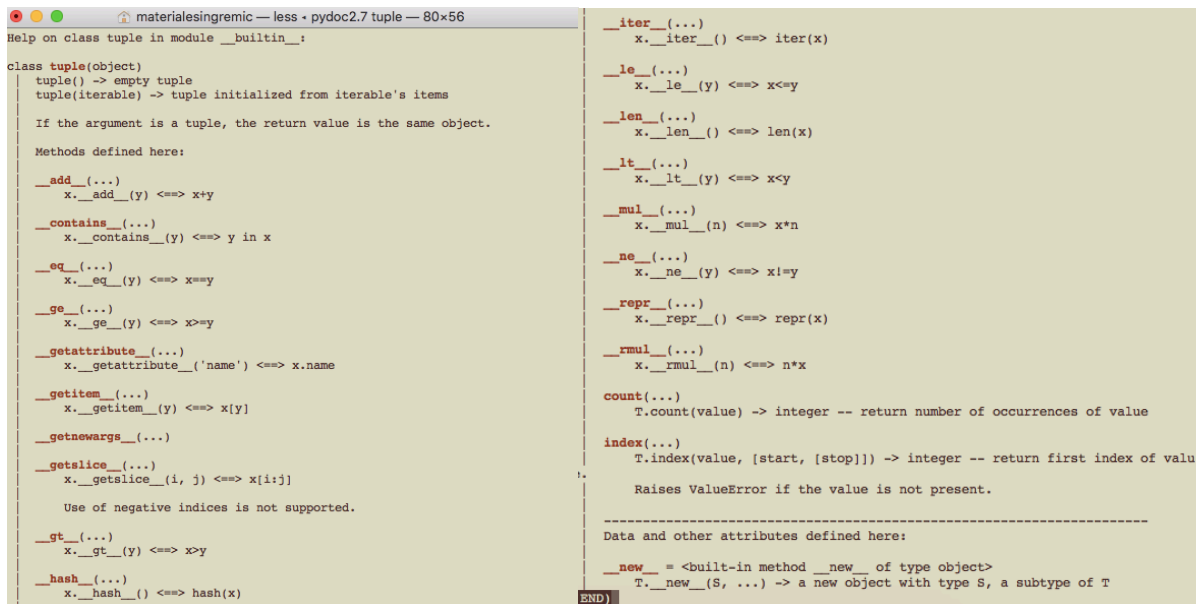
pydoc.py -g
  Pop up a graphical interface for finding and serving documentation.

pydoc.py -w <names> ...
  Write out the HTML documentation for a module to a file in the current
  directory. If <name> contains a '/', it is treated as a filename; if
  it names a directory, documentation is written for all the contents.

MacBook-Air-de-Materiales:~ materialesingremic$

Para esta práctica, he decidido centrarme en el uso de PyDoc, ya q
viene instalada con el propio Python. Con esta herramienta podem
```

Con el primer comando, podemos acceder a la documentación de algún módulo, clase, función o código. Por ejemplo, si escribimos: “python -m pydoc tuple” o “pydoc tuple” tendremos en pantalla toda la documentación módulo (a continuación 2 capturas de pantalla de dicha documentación mostrada en el terminal)



```

Help on class tuple in module __builtin__:

class tuple(object)
  tuple() -> empty tuple
  tuple(iterable) -> tuple initialized from iterable's items

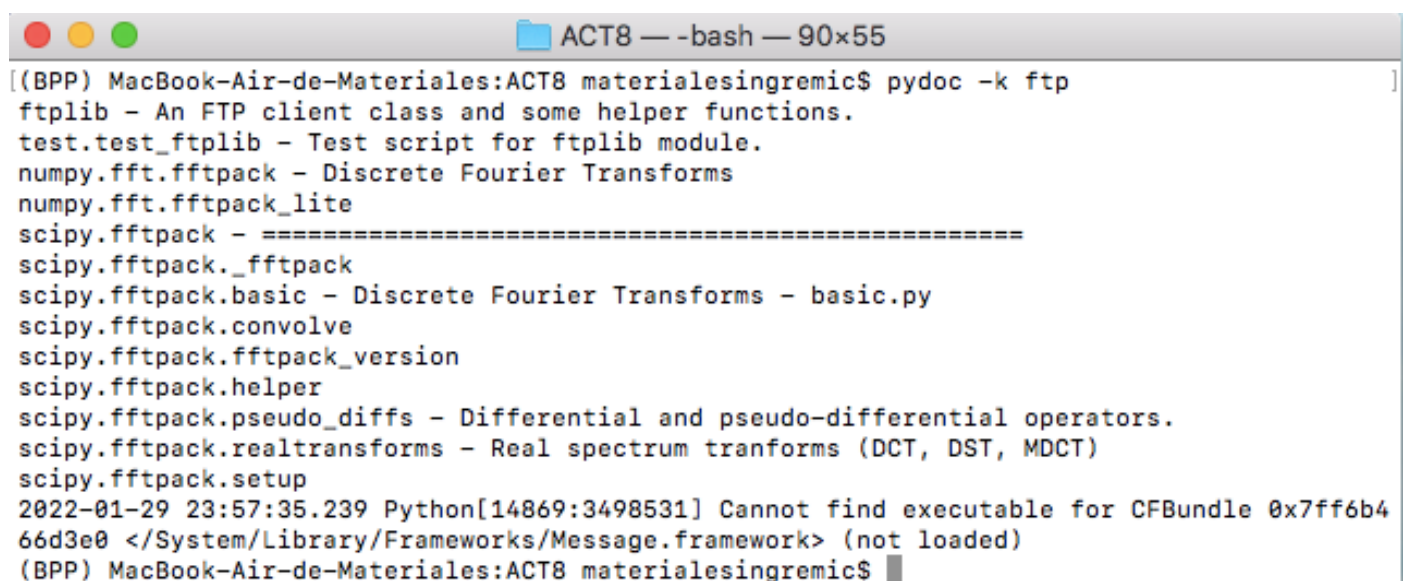
  If the argument is a tuple, the return value is the same object.

  Methods defined here:
    __add__(...)
      x.__add__(y) <=> x+y
    __contains__(...)
      x.__contains__(y) <=> y in x
    __eq__(...)
      x.__eq__(y) <=> x==y
    __ge__(...)
      x.__ge__(y) <=> x>=y
    __getattr__(...)
      x.__getattr__('name') <=> x.name
    __getitem__(...)
      x.__getitem__(y) <=> x[y]
    __getnewargs__(...)
    __getslice__(...)
      x.__getslice__(i, j) <=> x[i:j]

      Use of negative indices is not supported.
    __gt__(...)
      x.__gt__(y) <=> x>y
    __hash__(...)
      x.__hash__() <=> hash(x)
    __iter__(...)
      x.__iter__() <=> iter(x)
    __le__(...)
      x.__le__(y) <=> x<=y
    __len__(...)
      x.__len__() <=> len(x)
    __lt__(...)
      x.__lt__(y) <=> x<y
    __mul__(...)
      x.__mul__(n) <=> x*n
    __ne__(...)
      x.__ne__(y) <=> x!=y
    __repr__(...)
      x.__repr__() <=> repr(x)
    __rmul__(...)
      x.__rmul__(n) <=> n*x
    count(...)
      T.count(value) -> integer -- return number of occurrences of value
    index(...)
      T.index(value, [start, [stop]]) -> integer -- return first index of value
      Raises ValueError if the value is not present.

  -----
  Data and other attributes defined here:
    __new__ = <built-in method __new__ of type object>
      T.__new__(S, ...) -> a new object with type S, a subtype of T
  
```

El segundo uso del módulo pydoc podemos buscar una palabra en concreto contenida en todos los módulos disponibles. Por ejemplo: “pydoc -k ftp” nos muestra:



```

[ (BPP) MacBook-Air-de-Materiales:ACT8 materialesingremic$ pydoc -k ftp
ftplib - An FTP client class and some helper functions.
test.test_ftplib - Test script for ftplib module.
numpy.fft.fftpack - Discrete Fourier Transforms
numpy.fft.fftpack_lite
scipy.fftpack - =====
scipy.fftpack._fftpack
scipy.fftpack.basic - Discrete Fourier Transforms - basic.py
scipy.fftpack.convolve
scipy.fftpack.fftpack_version
scipy.fftpack.helper
scipy.fftpack.pseudo_diffs - Differential and pseudo-differential operators.
scipy.fftpack.realtransforms - Real spectrum tranforms (DCT, DST, MDCT)
scipy.fftpack.setup
2022-01-29 23:57:35.239 Python[14869:3498531] Cannot find executable for CFBundle 0x7ff6b4
66d3e0 </System/Library/Frameworks/Message.framework> (not loaded)
(BPP) MacBook-Air-de-Materiales:ACT8 materialesingremic$ ]
  
```

de esto, podemos buscar documentación sobre el módulo ftplib:

```

ACT8 — less « python -m pydoc ftplib — 90x55
Help on module ftplib:

NAME
    ftplib - An FTP client class and some helper functions.

FILE
    /System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/ftplib.py

MODULE DOCS
    https://docs.python.org/library/ftplib

DESCRIPTION
    Based on RFC 959: File Transfer Protocol (FTP), by J. Postel and J. Reynolds

    Example:

    >>> from ftplib import FTP
    >>> ftp = FTP('ftp.python.org') # connect to host, default port
    >>> ftp.login() # default, i.e.: user anonymous, passwd anonymous@
    '230 Guest login ok, access restrictions apply.'
    >>> ftp.retrlines('LIST') # list directory contents
    total 9
    drwxr-xr-x  8 root   wheel   1024 Jan  3  1994 .
    drwxr-xr-x  8 root   wheel   1024 Jan  3  1994 ..
    drwxr-xr-x  2 root   wheel   1024 Jan  3  1994 bin
    drwxr-xr-x  2 root   wheel   1024 Jan  3  1994 etc
    d-wxrwxr-x  2 ftp    wheel   1024 Sep  5 13:43 incoming
    drwxr-xr-x  2 root   wheel   1024 Nov 17  1993 lib
    drwxr-xr-x  6 1094    wheel   1024 Sep 13 19:07 pub
    drwxr-xr-x  3 root   wheel   1024 Jan  3  1994 usr
    -rw-r--r--  1 root   root    312 Aug  1  1994 welcome.msg
    '226 Transfer complete.'
    >>> ftp.quit()
    '221 Goodbye.'
    >>>

    A nice test that reveals some of the network dialogue would be:
    python ftplib.py -d localhost -l -p -l
    
```

Con el tercer uso de **pydoc** podemos iniciar un servidor HTTP en la máquina local que proporcione documentación a los navegadores web visitantes. Por ejemplo, **pydoc -p 314** iniciará un servidor HTTP en el puerto 314, lo que le permitirá navegar por la documentación en <http://localhost:1234/> en su navegador web preferido. Si especifica 0 como número de puerto, se seleccionará un puerto arbitrario no utilizado. En mi caso, debido a algún tipo de error, tuve que iniciar un servidor en un puerto libre.

```

ACT8 — Python -m pydoc -p 0 — 90x55
((BPP) MacBook-Air-de-Materiales:ACT8 materialesingremic$ python3 -m pydoc -p 314
[Errno 13] Permission denied
((BPP) MacBook-Air-de-Materiales:ACT8 materialesingremic$ python3 -m pydoc -p 315
[Errno 13] Permission denied
((BPP) MacBook-Air-de-Materiales:ACT8 materialesingremic$ python3 -m pydoc -p 234
[Errno 13] Permission denied
((BPP) MacBook-Air-de-Materiales:ACT8 materialesingremic$ python3 -m pydoc -p 777
[Errno 13] Permission denied
((BPP) MacBook-Air-de-Materiales:ACT8 materialesingremic$ python3 -m pydoc -p 543
[Errno 13] Permission denied
((BPP) MacBook-Air-de-Materiales:ACT8 materialesingremic$ python3 -m pydoc -p 0
Server ready at http://localhost:60974/
Server commands: [b]rowser, [q]uit
server>
    
```

si pulsamos b, automáticamente se nos abre una página web con el puerto local seleccionado:

Python 3.8.5 [v3.8.5:580fbb018f, Clang 6.0 (clang-600.0.57)]
macOS-10.13.6

Module Index : Topics : Keywords

Index of Modules

Built-in Modules

_abc	_operator	_warnings	marshal
_ast	_signal	_weakref	posix
_codecs	_sre	atexit	pwd
_collections	_stat	builtins	sys
_functools	_string	errno	time
_imp	_symtable	faulthandler	xxsubtype
_io	_thread	gc	
_locale	_tracemalloc	itertools	

/Users/materialesingremic/Desktop/M_Python/ProgAvPython/Lec8/CLASE8/ACT8

[main](#) [pruebas](#)

/Library/Frameworks/Python.framework/Versions/3.8/lib/python38.zip

/Library/Frameworks/Python.framework/Versions/3.8/lib/python3.8

_future	curses (package)	modulefinder	sre_compile
_bootlocale	dataclasses	multiprocessing (package)	sre_constants
collections_abc	datetime	netrc	sre_parse
compat_pickle	dbm (package)	nntplib	ssl
compression	decimal	ntpath	stat
dummy_thread	difflib	nturl2path	statistics
markupbase	dis	numbers	string
osx_support	distutils (package)	opcode	stringprep
py_abc	doctest	operator	struct
pydecimal	dummy_threading	optparse	subprocess

Podemos concluir de esta opción, que es una muy buena herramienta para estudio e información de ciertos módulos, ya que clicando en cada uno de ellos, obtendremos su correspondiente documentación. Por ejemplo, si pulsamos en django, obtenemos:

Python 3.8.5 [v3.8.5:580fbb018f, Clang 6.0 (clang-600.0.57)]
macOS-10.13.6

Module Index : Topics : Keywords

django (version 3.1.7)

/Library/Frameworks/Python.framework/Versions/3.8/lib/python3.8/site-packages/django/

Package Contents

__main__	db (package)	shortcuts	utils (package)
apps (package)	dispatch (package)	template (package)	views (package)
conf (package)	forms (package)	templatetags (package)	
contrib (package)	http (package)	test (package)	
core (package)	middleware (package)	urls (package)	

Functions

[setup\(set_prefix=True\)](#)
Configure the settings (this happens as a side effect of accessing the first setting), configure logging and populate the app registry.
Set the thread-local urlresolvers script prefix if `set_prefix` is True.

Data

[VERSION = \(3, 1, 7, 'final', 0\)](#)

La cuarta función es similar, en ella, automáticamente se nos abrirá un servidor disponible sin la necesidad de ocupar un espacio local para crearlo.

Finalmente, tenemos la opción de crear archivos html con la documentación de nuestros códigos. En el apartado 2, realizaremos la documentación de un código creado en actividades previas.

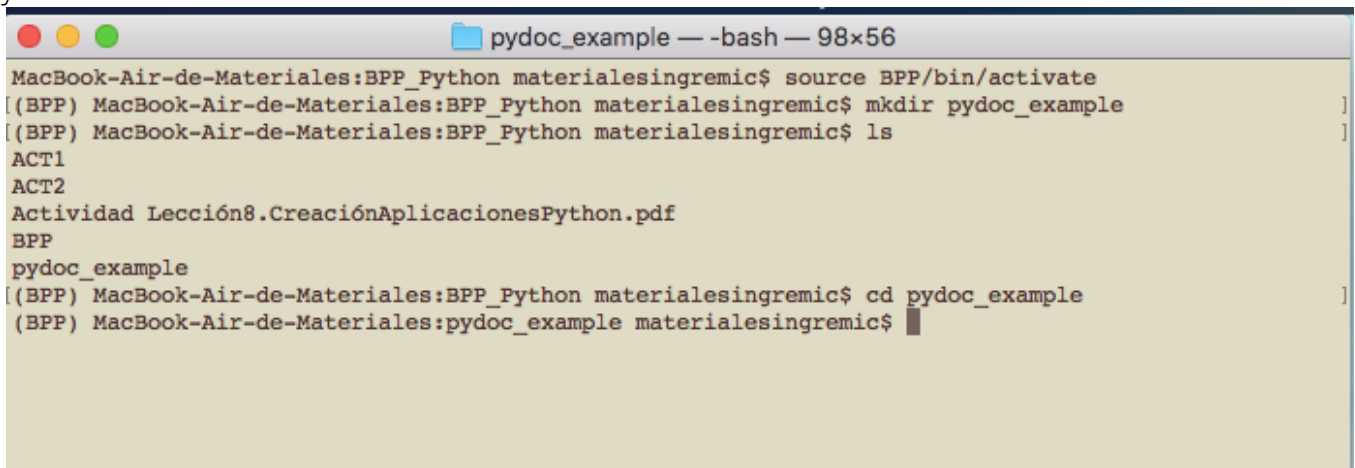
APARTADO 1:

2. Realice la documentación de alguno de los códigos que ha desarrollado a lo largo de este curso. La documentación puede generarla con Sphinx o con la herramienta que ha buscado en el primer ejercicio.

Para la ejecución de este apartado, utilizaremos la función `-w` del módulo `pydoc`.

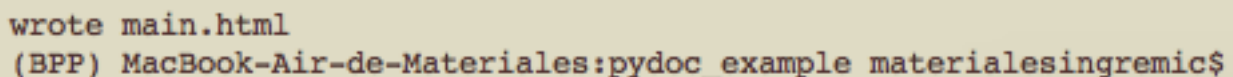
Primeramente, creamos una carpeta en la que guardaremos el contenido generado. Esta carpeta se llamará `pydoc_example`:

En esta carpeta, copiaremos el código del cuál queremos generar la documentación html con `pydoc`.



```
MacBook-Air-de-Materiales:BPP_Python materialesingremic$ source BPP/bin/activate
[(BPP) MacBook-Air-de-Materiales:BPP_Python materialesingremic$ mkdir pydoc_example ]
[(BPP) MacBook-Air-de-Materiales:BPP_Python materialesingremic$ ls ]
ACT1
ACT2
Actividad Lección8.CreaciónAplicacionesPython.pdf
BPP
pydoc_example
[(BPP) MacBook-Air-de-Materiales:BPP_Python materialesingremic$ cd pydoc_example ]
[(BPP) MacBook-Air-de-Materiales:pydoc_example materialesingremic$ ]
```

Para crear la documentación html del código, tecleamos: `python3 -m pydoc -w main`, de esto obtendremos:



```
wrote main.html
(BPP) MacBook-Air-de-Materiales:pydoc_example materialesingremic$ ]
```

Vemos como se ha ejecutado el código y en la parte final, leemos `wrote main.html`.

Si buscamos este archivo en la carpeta pydoc_example, podemos acceder a la documentación generada.

```
main /Users/materialesingremic/Desktop/M_Python/BPP_Python/pydoc_example/main.py index

Actividad 8 Asignatura Programación Avanzada en Python:
1. Desarrolla un script en Python que abra un fichero de texto '.txt'.
   Escribe unas cuantas líneas en el fichero y finalmente muestra el contenido de dicho fichero por pantalla.
2. Desarrolla un script que cargue el fichero cotizaciones.csv subido a la plataforma y construya un dataframe en pandas.
   Muestra el mínimo, máximo y media de cada columna.

Modules
pandas

Functions
maximo()
    #Esta función buscará el valor máximo de cada columna del dataframe creado a partir del documento csv
media()
    #Esta función calculará la media de cada columna del dataframe creado a partir del documento csv
minimo()
    #Esta función buscará el valor mínimo de cada columna del dataframe creado a partir del documento csv

Data
__warningregistry__ = {'version': 0}
datos = Nombre Final Máximo Mf...0 49.1200 48.4600 67.71 3295.43
df = Nombre Final Máximo Mf...0 49.1200 48.4600 67.71 3295.43
element = 'Sexta Línea del apartado 1 de la actividad 8'
fichero = <_io.TextIOWrapper name='Act8_ap1.txt' mode='r' encoding='UTF-8'>
```

Bibliografía y documentación:

- <https://www.youtube.com/watch?v=URBSvqib0xw>
- <https://docs.python.org/es/3/library/pydoc.html>
- <https://docs.hektorprofe.net/python/documentacion-y-pruebas/pydoc/>