

## Actividad Práctica 4: Fundamentos de Python

Para esta práctica he elegido la opción de implementar una función en Python. El código es el siguiente:

```
4 def primos(num):
5     numPrimos = []
6     for i in range(2, num + 1):
7         primo = True
8         for x in range(2, i):
9             if x != num and i % x == 0:
10                primo = False;
11        if primo:
12            numPrimos.append(i)
13    return numPrimos
14
15 prueba = int(input("Introduzca un número: "))
16
17 print(primos(prueba))
```

La ejecución del programa sería: se pide un número al usuario y, haciendo uso de dos bucles *for*, se comprueba qué números son primos desde 2 hasta *num + 1* (partimos de 2 ya que 1 no es un número primo y le sumamos 1 a *num* ya que los bucles *for* en Python no incluyen el propio número establecido de fin, por lo que, si el número introducido es primo, no aparecerá en la lista) comprobando el resto de dicho número al dividirlo por todos sus anteriores. Cada número se va añadiendo a un array previamente declarado para, finalmente, ser mostrado al usuario.

Aquí un ejemplo de su uso:

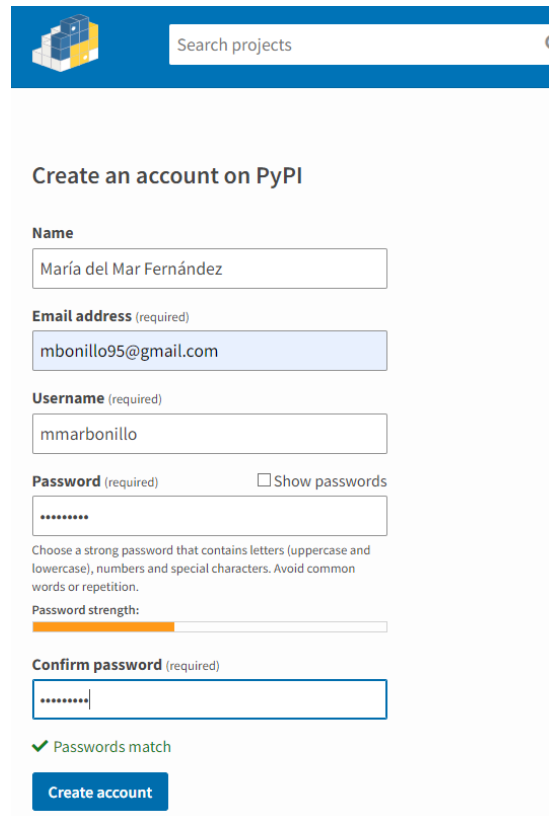
```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL

PS C:\Users\Boni\Documents\Máster\Fundamentos de Python\ej04> python .\ej04.py
Introduzca un número: 61
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61]
PS C:\Users\Boni\Documents\Máster\Fundamentos de Python\ej04> █
```

Pero para subirlo a pypi.org lo dejo así:

```
ej04 > ej04.py > ...
1
2 def primos(num):
3     numPrimos = []
4     for i in range(2, num + 1):
5         primo = True
6         for x in range(2, i):
7             if x != num and i % x == 0:
8                 primo = False;
9         if primo:
10            numPrimos.append(i)
11    return numPrimos
12
```

Lo primero va a ser crear el paquete, por lo que, antes de nada, nos registramos en pypi y comprobamos que el nombre que le queremos poner al paquete, no exista ya. Yo le voy a llamar *primosMmarbonillo*.



The image shows the 'Create an account on PyPI' form. It includes fields for Name, Email address, Username, Password, and Confirm password. The Name field contains 'María del Mar Fernández', Email address contains 'mbonillo95@gmail.com', and Username contains 'mmarbonillo'. The Password field is masked with dots, and there is a 'Show passwords' checkbox. Below the password field is a strength indicator bar. The Confirm password field also contains masked text. A green checkmark indicates 'Passwords match'. At the bottom is a 'Create account' button.

Search projects

### Create an account on PyPI

**Name**

María del Mar Fernández

**Email address** (required)

mbonillo95@gmail.com

**Username** (required)

mmarbonillo

**Password** (required) ☐ Show passwords

\*\*\*\*\*

Choose a strong password that contains letters (uppercase and lowercase), numbers and special characters. Avoid common words or repetition.

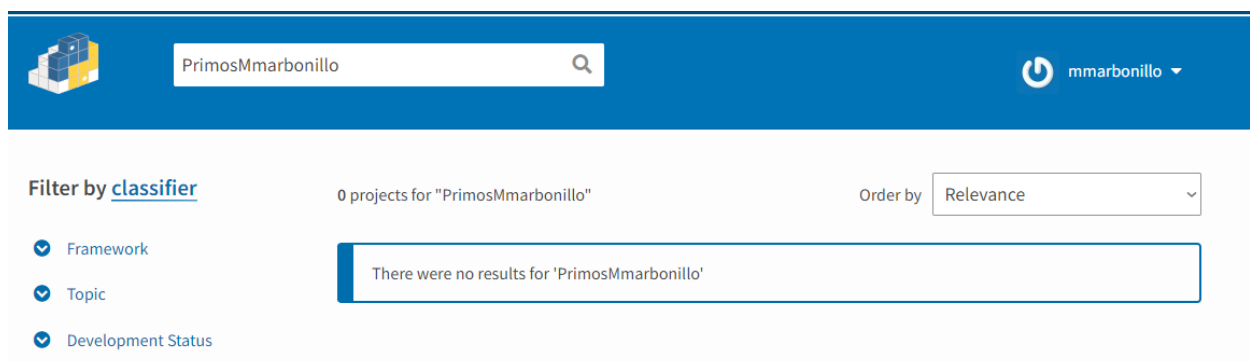
Password strength:

**Confirm password** (required)

\*\*\*\*\*

✓ Passwords match

Create account



The image shows the PyPI search results page for 'PrimosMmarbonillo'. The search bar at the top contains 'PrimosMmarbonillo'. Below the search bar, there are filters for 'classifier' (Framework, Topic, Development Status) and a dropdown for 'Order by' set to 'Relevance'. The results section shows '0 projects for "PrimosMmarbonillo"'. A message box states 'There were no results for 'PrimosMmarbonillo''.

PrimosMmarbonillo

mmarbonillo

Filter by [classifier](#)

0 projects for "PrimosMmarbonillo"

Order by Relevance

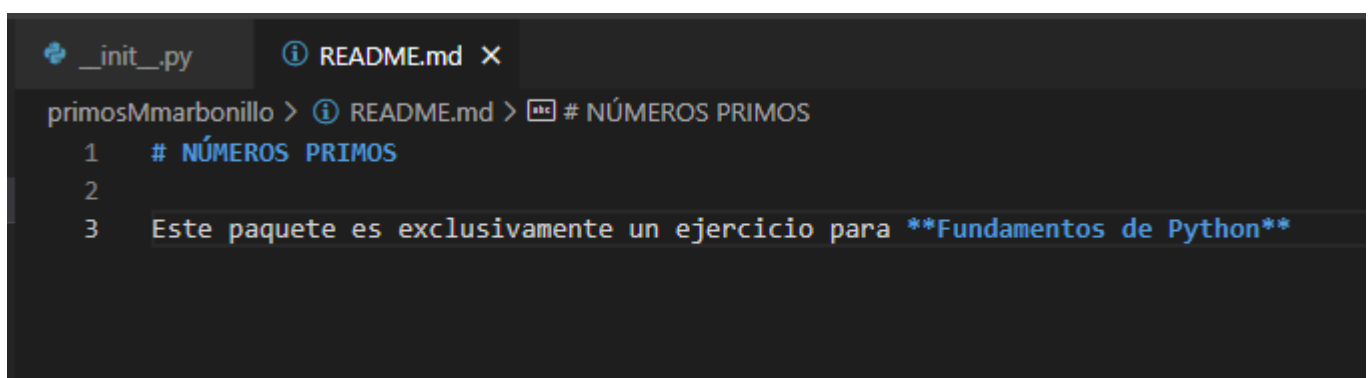
Framework

Topic

Development Status

There were no results for 'PrimosMmarbonillo'

Como pasos iniciales voy a cambiar el nombre del archivo que contiene la función por `__init__.py` y voy a crear un archivo `README.md` muy sencillo



The image shows a code editor with two files open: `__init__.py` and `README.md`. The `__init__.py` file is selected, and the code inside is: `primosMmarbonillo > # NÚMEROS PRIMOS`. The `README.md` file is also open, and its content is: `1 # NÚMEROS PRIMOS`, `2`, and `3 Este paquete es exclusivamente un ejercicio para **Fundamentos de Python**`.

```
primosMmarbonillo > # NÚMEROS PRIMOS
1 # NÚMEROS PRIMOS
2
3 Este paquete es exclusivamente un ejercicio para **Fundamentos de Python**
```

Creo el *License.txt*. Como no se como se hace, he buscado el texto clásico de licencias de Python de código libre:

```
primosMmarbonillo > License.txt
1 Copyright (c) 2018 The Python Packaging Authority
2
3 Permission is hereby granted, free of charge, to any person obtaining a copy
4 of this software and associated documentation files (the "Software"), to deal
5 in the Software without restriction, including without limitation the rights
6 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
7 copies of the Software, and to permit persons to whom the Software is
8 furnished to do so, subject to the following conditions:
9
10 The above copyright notice and this permission notice shall be included in all
11 copies or substantial portions of the Software.
12
13 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
14 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
15 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
16 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
17 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
18 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
19 SOFTWARE.
```

Lo último que hay que crear es el archivo *Setup.py*:

```
primosMmarbonillo > Setup.py > ...
1 import setuptools
2
3 with open("README.md", "r") as fh:
4     long_description = fh.read()
5
6 setuptools.setup(
7
8     name="primosMmarbonillo",
9     version="0.0.1",
10    author="María Del Mar Fernández Bonillo",
11    author_email="mbonillo95@gmail.com",
12    keywords="mmarbonillo primosMmarbonillo primos",
13    packages=setuptools.find_packages(),
14    license="MIT",
15
16    classifiers=[
17        "Development Status :: 4 - Beta",
18        "Intended Audience :: Developers",
19        "Topic :: Utilities",
20        "Programming Language :: Python :: 3",
21        "License :: OSI Approved :: MIT License",
22        "Operating System :: OS Independent",
23    ]
24 )
```

Pasamos ahora a crear el paquete y subirlo. Primero vamos a instalar *wheel* y *twine* ejecutando estos comandos:

```
PS C:\Users\Boni\Documents\Máster\Fundamentos de Python\ej04> pip install wheel
PS C:\Users\Boni\Documents\Máster\Fundamentos de Python\ej04> pip install twine
PS C:\Users\Boni\Documents\Máster\Fundamentos de Python\ej04> |
```

Usamos *wheel* para crear el paquete

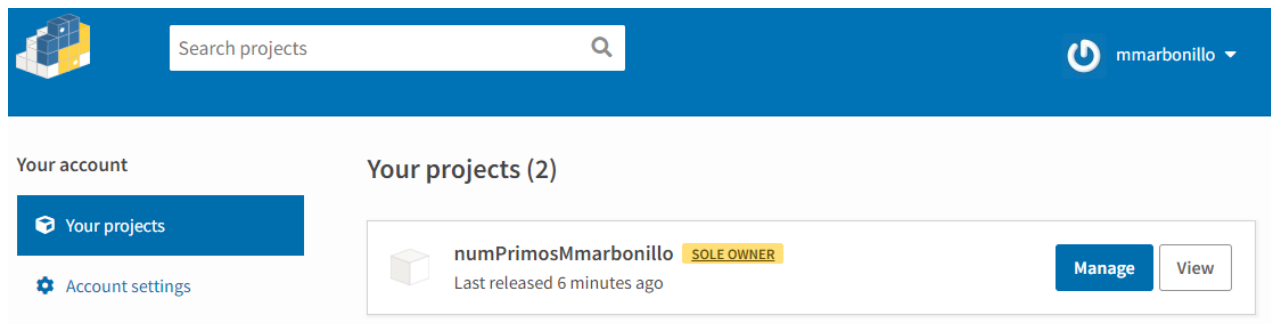
```
C:\Users\Boni\Documents\Máster\Fundamentos de Python\primosMmarbonillo>python ./setup.py bdist_wheel
running bdist_wheel
running build
installing to build\bdist.win-amd64\wheel
running install
running install_egg_info
running egg_info
creating primosMmarbonillo.egg-info
writing primosMmarbonillo.egg-info\PKG-INFO
writing dependency_links to primosMmarbonillo.egg-info\dependency_links.txt
writing top-level names to primosMmarbonillo.egg-info\top_level.txt
writing manifest file 'primosMmarbonillo.egg-info\SOURCES.txt'
reading manifest file 'primosMmarbonillo.egg-info\SOURCES.txt'
adding license file 'License.txt'
writing manifest file 'primosMmarbonillo.egg-info\SOURCES.txt'
Copying primosMmarbonillo.egg-info to build\bdist.win-amd64\wheel\.\primosMmarbonillo-0.0.1-py3.10.egg-info
running install_scripts
adding license file "License.txt" (matched pattern "LICEN[CS]E*")
creating build\bdist.win-amd64\wheel\primosMmarbonillo-0.0.1.dist-info\WHEEL
creating 'dist\primosMmarbonillo-0.0.1-py3-none-any.whl' and adding 'build\bdist.win-amd64\wheel' to it
adding 'primosMmarbonillo-0.0.1.dist-info/License.txt'
adding 'primosMmarbonillo-0.0.1.dist-info/METADATA'
adding 'primosMmarbonillo-0.0.1.dist-info/WHEEL'
adding 'primosMmarbonillo-0.0.1.dist-info/top_level.txt'
adding 'primosMmarbonillo-0.0.1.dist-info/RECORD'
removing build\bdist.win-amd64\wheel
C:\Users\Boni\Documents\Máster\Fundamentos de Python\primosMmarbonillo>
```

Ejecuto el comando para subir el paquete y pide usuario y contraseña:

```
Administrador: Símbolo del sistema
C:\Users\Boni\Documents\Máster\Fundamentos de Python\primosMmarbonillo>twine upload dist/*
Uploading distributions to https://upload.pypi.org/legacy/
Enter your username: mmarbonillo
Enter your password:
Uploading primosMmarbonillo-0.0.1-py3-none-any.whl
100% ----- 5.6/5.6 kB • 00:00 • ?
View at:
https://pypi.org/project/primosMmarbonillo/0.0.1/
C:\Users\Boni\Documents\Máster\Fundamentos de Python\primosMmarbonillo>
```

Ahora vamos a instalar y probar el paquete. (Por el camino de instalarlo y probarlo, como no funcionaba porque daba error de “No existe el módulo *primosMmarbonillo*”, he intentado arreglarlo y lo he acabado rompiendo ya que lo borrado de mi cuenta de pypi pero no me ha devuelto el nombre *primosMmarbonillo*, así que a partir de aquí, se llama *numPrimosMmarbonillo*. Al final todo era que el árbol de directorios estaba mal montado, me faltaba una carpeta dónde meter el archivo `__init__.py`)

Primero comprobamos que se ha subido bien a pypi



Y ahora lo instalamos:

```
C:\Users\Boni\Documents\Máster\Fundamentos de Python>pip install numPrimosMmarbonillo
Collecting numPrimosMmarbonillo
  Downloading numPrimosMmarbonillo-0.1.0-py3-none-any.whl (2.5 kB)
Installing collected packages: numPrimosMmarbonillo
Successfully installed numPrimosMmarbonillo-0.1.0

C:\Users\Boni\Documents\Máster\Fundamentos de Python>
```

Para probarlo, en otra carpeta me creo un pequeño archivo que importe y haga uso de esta librería:

The screenshot shows a code editor with several tabs: "\_\_init\_\_.py", "Readme.md", "License.txt", "Setup.py", and "prueba.py". The "prueba.py" tab is active, showing the following code:

```
1 import numPrimosMmarbonillo as primos
2
3 prueba = int(input("Introduzca un número: "))
4
5 print(primos.primos(prueba))
```

Below the code editor, there is a terminal window with tabs for "PROBLEMAS", "SALIDA", "CONSOLA DE DEPURACIÓN", and "TERMINAL". The "TERMINAL" tab is active, showing the command prompt output:

```
PS C:\Users\Boni\Documents\Máster\Fundamentos de Python\pruebaPrimos> python .\prueba.py
Introduzca un número: 61
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61]
PS C:\Users\Boni\Documents\Máster\Fundamentos de Python\pruebaPrimos>
```