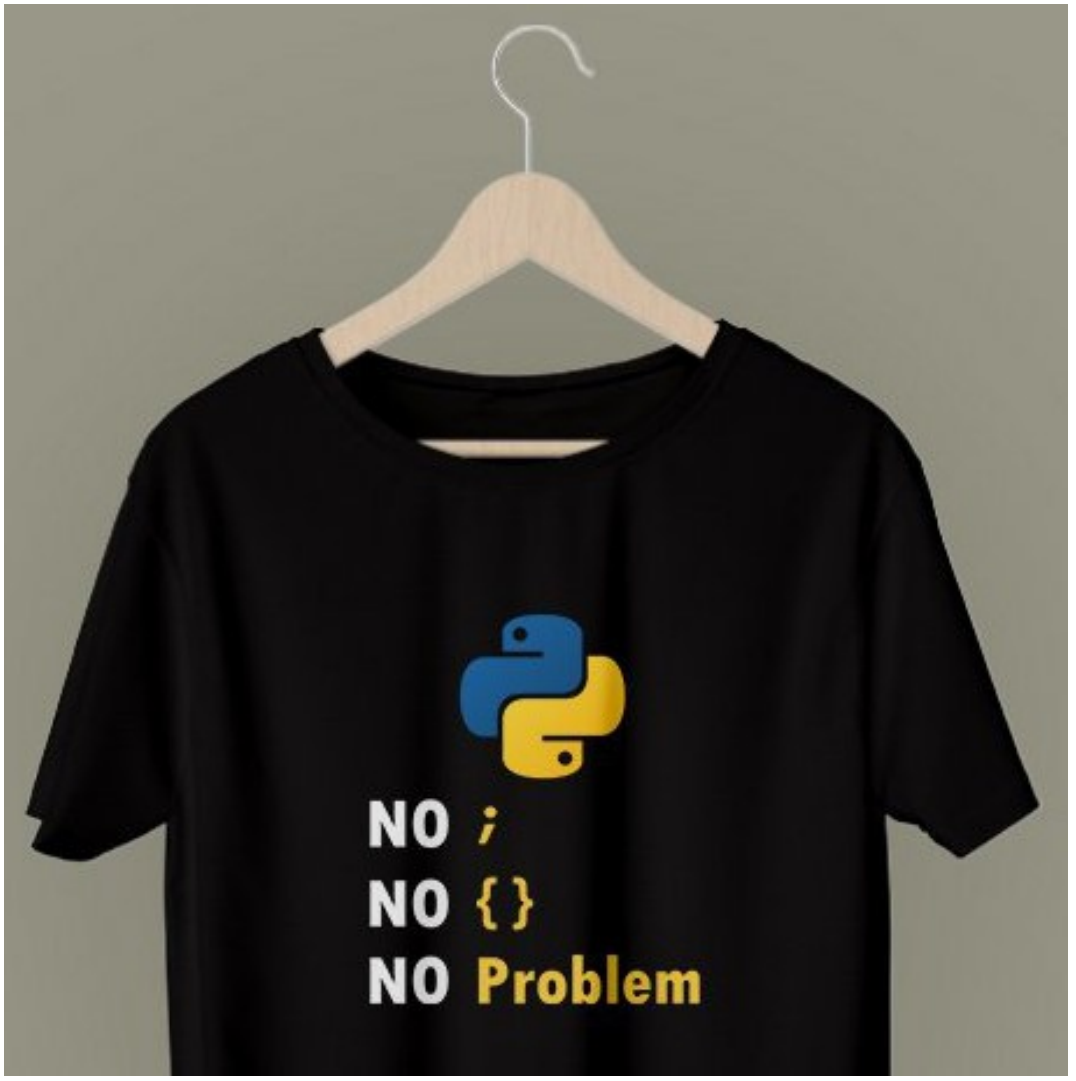
“Python - Programación”

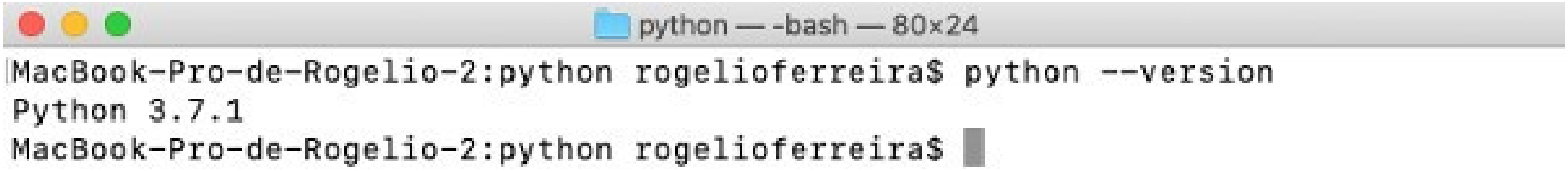


Rogelio Ferreira Escutia

Programación

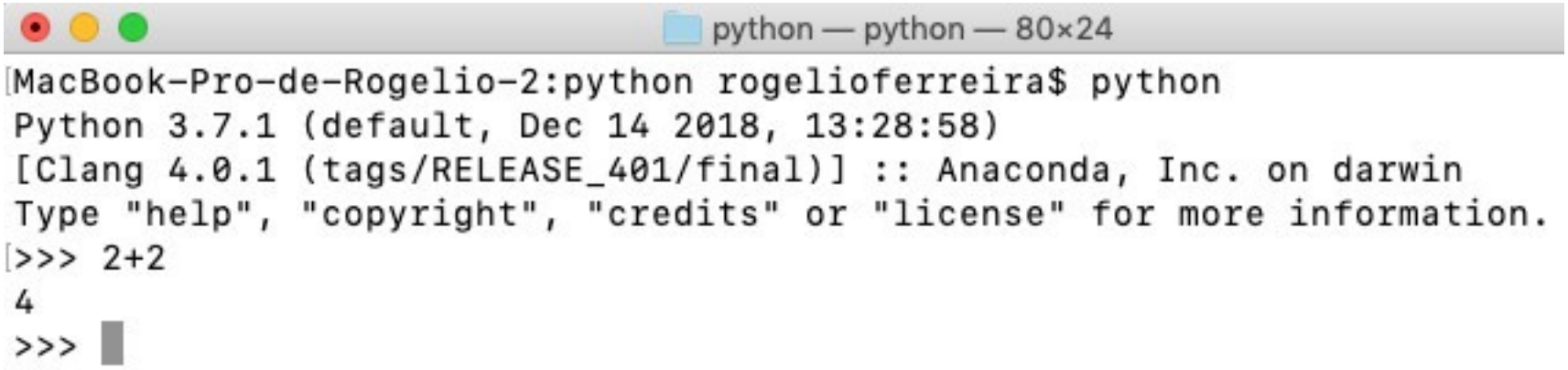
Python - Versión

- **Para ver la versión instalada (en consola):**



Python - Intérprete

- **Para entrar al intérprete de Python (en consola):**



Python – Intérprete (Ayuda)

- **Para entrar a la ayuda en el intérprete de Python (en consola):**

```
>>> help()

Welcome to Python 3.7's help utility!

If this is your first time using Python, you should definitely check out
the tutorial on the Internet at https://docs.python.org/3.7/tutorial/.

Enter the name of any module, keyword, or topic to get help on writing
Python programs and using Python modules.  To quit this help utility and
return to the interpreter, just type "quit".

To get a list of available modules, keywords, symbols, or topics, type
"modules", "keywords", "symbols", or "topics".  Each module also comes
with a one-line summary of what it does; to list the modules whose name
or summary contain a given string such as "spam", type "modules spam".

help> keywords

Here is a list of the Python keywords.  Enter any keyword to get more help.

False               class               from                or
None                continue            global              pass
True                def                 if                  raise
and                 del                 import              return
as                  elif                in                  try
assert              else                is                  while
async               except              lambda              with
await               finally             nonlocal            yield
break               for                 not

help>
```

Python – Hola Mundo (consola)

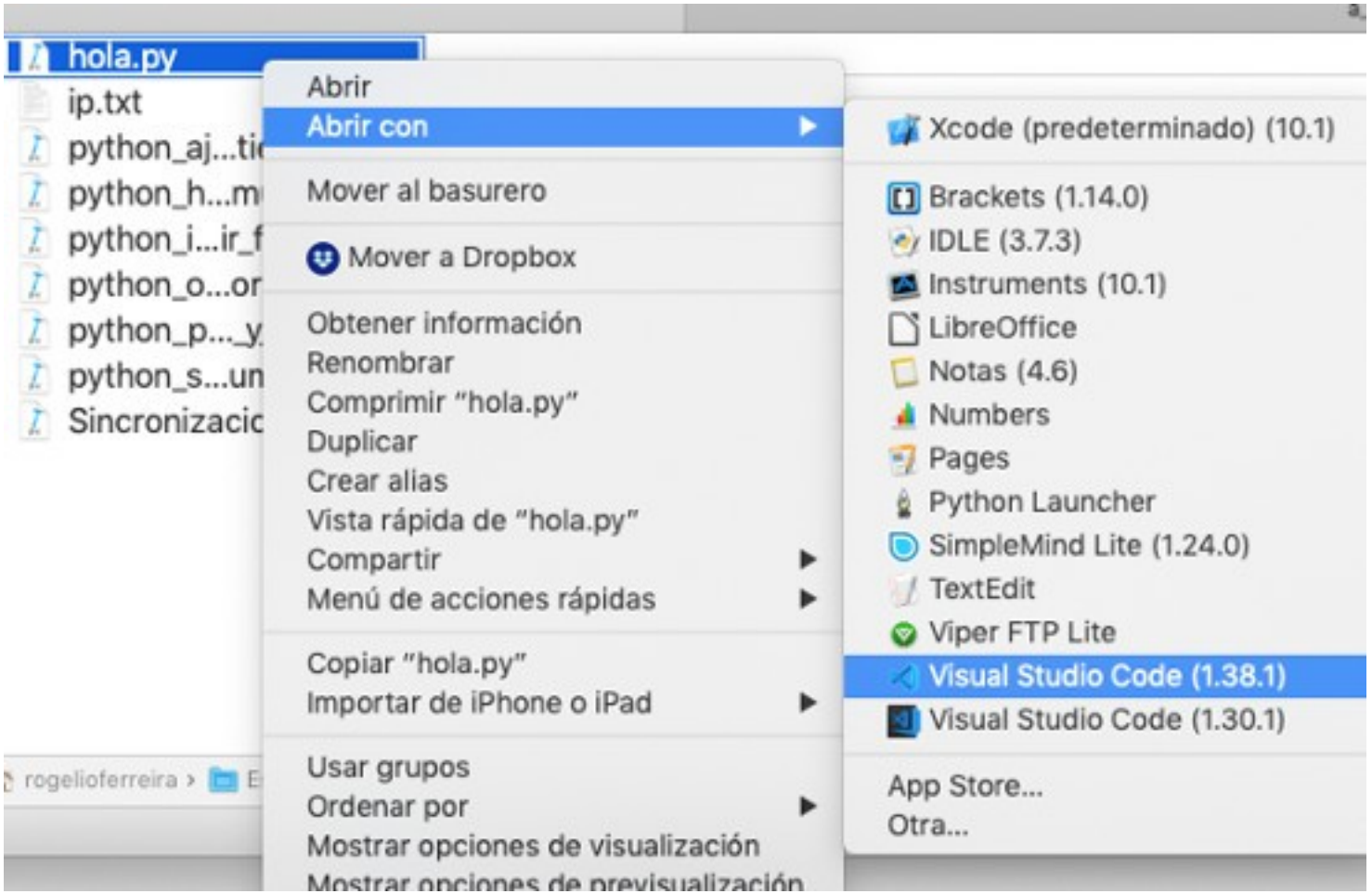
- **> nano hola.py**

- **> python hola.py**

Hola Crayola!!!

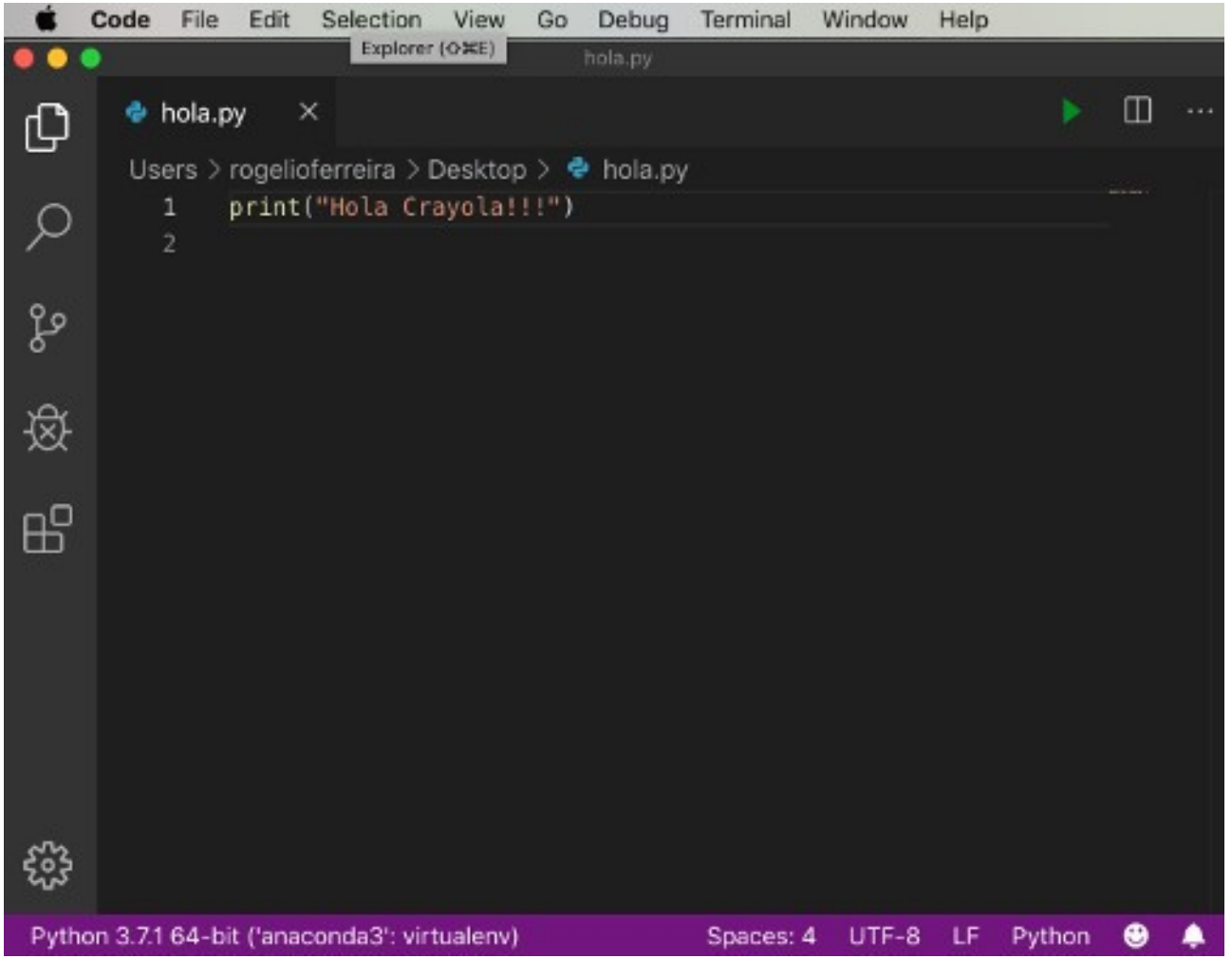
Python – Hola mundo (con VSCode)

- **Abrir el archivo con el Visual Studio Code (previamente instalado)**



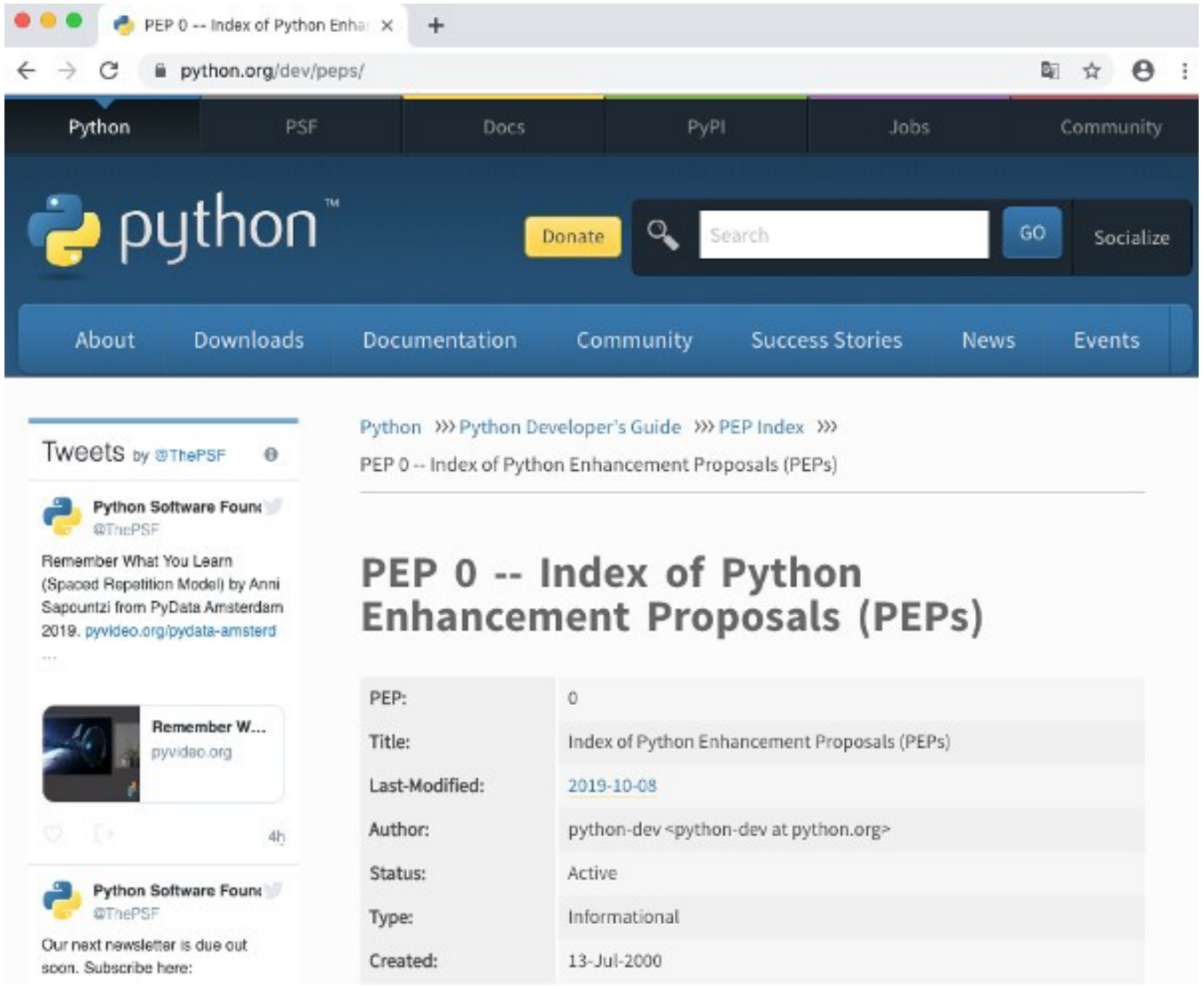
Python – Hola mundo (con VSCode)

- **Abrir el archivo con el Visual Studio Code (previamente instalado) y correrlo en la consola:**



Zen of Python

- **Es una lista de principios de diseño para el lenguaje Python:**

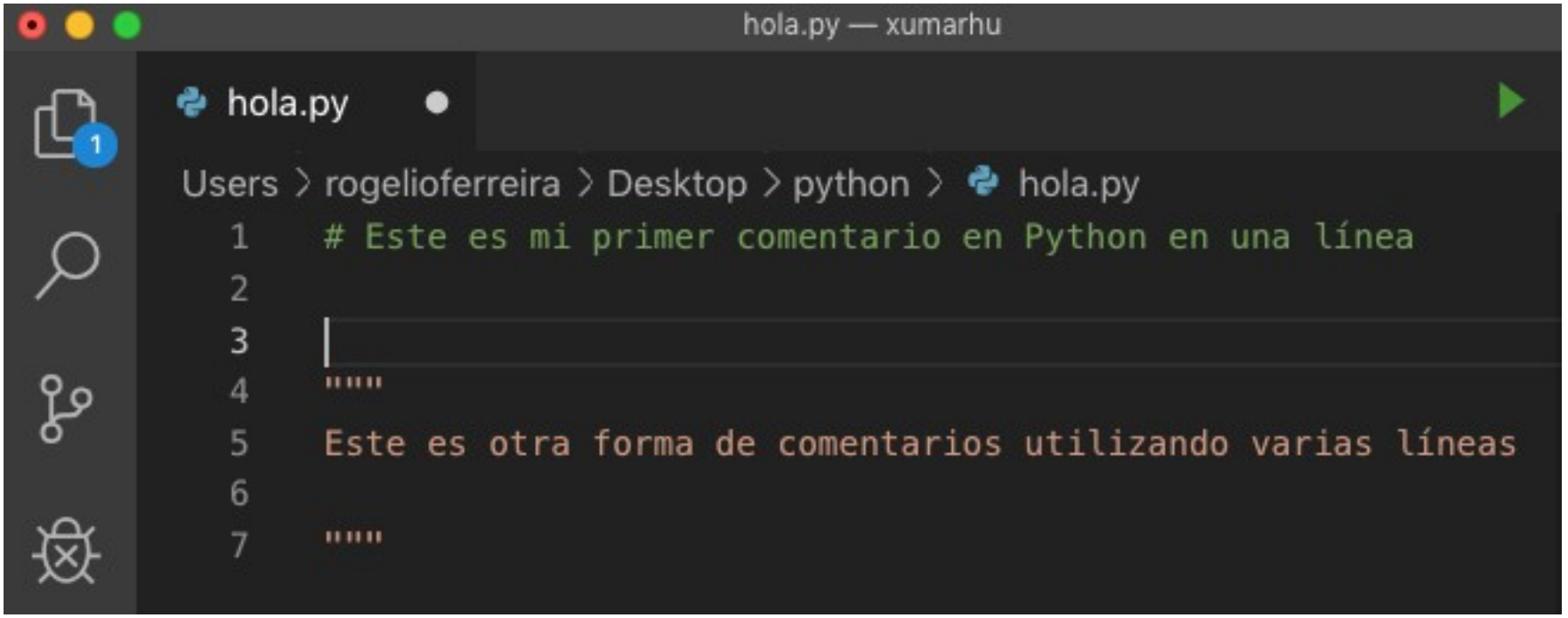


The Zen of Python

```
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

Comentarios

- **Para poner comentarios en nuestro código:**



Tipos de Datos

Enteros (int)

Number	Okay?	Reason
1	Good	A whole number (integer)
1.1	Good	A number with a decimal point
1234567.89	Good	A large number with a decimal point and no commas
-2	Good	A negative number, as indicated by the starting hyphen
.99	Good	A number that starts with a decimal point because it's less than one.
$1.99	Bad	Contains a $
12,345.67	Bad	Contains a comma
1101 3232	Bad	Contains a space
91740-3384	Bad	Contains a hyphen
123-45-6789	Bad	Contains two hyphens
123 Oak Tree Lane	Bad	Contains spaces and words
(267)555-1234	Bad	Contain parentheses and hyphens
127.0.0.1	Bad	Only one decimal point is allowed

Cadenas (strings)

```
"Hi there, I am a string"
'Hello world'
"123 Oak Tree Lane"
"(267)555-1234"
"18901-3384"
```

Boleanas (boolean)

```
x = True
```

```
X = False
```

"Python All in one for Dummies", John Sovic y Alan Simpson, Jhon Wiley and Sons Editorial, New Jersey USA 2019

Operadores

Operadores Aritméticos

Operator	Description	Example
+	Addition	1 + 1 = 2
-	Subtraction	10 - 1 = 9
*	Multiplication	3 * 5 = 15
/	Division	10 / 5 = 2
%	Modulus (remainder after division)	11 % 5 = 1
**	Exponent	3**2 = 9
//	Floor division	11 // 5 = 2

"Python All in one for Dummies", John Sovic y Alan Simpson, Jhon Wiley and Sons Editorial, New Jersey USA 2019

Operadores de comparación

Operator	Meaning
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to
!=	Not equal to
is	Object identity
is not	Negated object identity

"Python All in one for Dummies", John Sovic y Alan Simpson, Jhon Wiley and Sons Editorial, New Jersey USA 2019

Operadores Boleanos

Operator	Code Example	What It Determines
or	`x or y`	Either x or y is true
and	`x and y`	Both x and y are true
not	`not x`	x is not full

Variables

Asignación de valores a variables

```
variablename = value
```

```
x = 10
```

```
user_name = "Alan"
```

Manejo de variables

hello.py

```
1   # This is a Python comment in my first Python app.
2   # This variable contains an integer
3   quantity = 10
4   # This variable contains a float
5   unit_price = 1.99
6   # This variable contains the result of multiplying quantity times unit price
7   extended_price = quantity * unit_price
8   # Show the results
9   print(extended_price)
10
```

Funciones integradas

Funciones integradas

- **Python cuenta con algunas funciones ya integradas al lenguaje:**

Built-In Function	Purpose
`abs(x)`	Returns the absolute value of number *x* (converts negative numbers to positive)
`bin(x)`	Returns a string representing the value of *x* converted to binary.
`float(x)`	Converts a string or number *x* to a the float data type
`format(x,y)`	Returns *x* formatted as directed by format string *y*. In modern Python you're more likely to use f-strings, as described later in this chapter
`hex(x)`	Returns a string containing *x* converted to hexadecimal, prefixed with 0x.
`int(x)`	Converts *x* to the integer data type by truncating (not rounding) the decimal point and any digits after it.
`max(x,y,z ...)`	Takes any number of numeric arguments and returns whichever one is the largest.
`min(x,y,z ...)`	Takes any number of numeric arguments and returns whichever one is the smallest.
`oct(x)`	Converts *x* to an octal number, prefixed with 0o to indicate octal.
`round(x,y)`	Rounds the number *x* to *y* number of decimal places.
`str(x)`	Converts number *x* to the string data type.
`type(x)`	Returns a string indicating the data type of *x*.

"Python All in one for Dummies", John Sovic y Alan Simpson, Jhon Wiley and Sons Editorial, New Jersey USA 2019

Toma de Decisiones

Toma de decisiones (if)

```
num = 10
if num > 0:
    print("Positive number")
else:
    print("Negative number")
```

"Python All in one for Dummies", John Sovic y Alan Simpson, Jhon Wiley and Sons Editorial, New Jersey USA 2019

Rogelio Ferreira Escutia

Instituto Tecnológico de Morelia
Departamento de Sistemas y Computación

Correo: ***rogelio@itmorelia.edu.mx***
rogeplus@gmail.com

Página Web: ***http://sagitario.itmorelia.edu.mx/~rogelio/***
http://www.xumarhu.net/

Twitter: ***http://twitter.com/rogeplus***
Facebook: ***http://www.facebook.com/groups/xumarhu.net/***