

En este documento comentaré algún error cometido en el libro, además de notas aclaratorias sobre algunos aspectos.

Empezaré por los errores/aclaraciones, donde marco con una viñeta de color rojo los que considero mas importantes:

- El capítulo 4, "Programación funcional", tiene un título no muy afortunado, ya que en él trato, en realidad, la programación **imperativa** haciendo uso de **funciones**. Es por ello que el título del capítulo sería mas conveniente que fuese simplemente "Funciones". Por lo tanto, al encontrar en el capítulo "programación funcional" debemos sustituirlo por "programación mediante funciones", siempre teniendo en cuenta que es dentro de la programación imperativa. Como comento en la página 21 del libro, *Python* sí permite la programación funcional, pero no es tratada como tal en el libro. Elementos de los que sí hablo en el libro (como las comprensiones de lista, o listas de comprensión) tienen relación con ella. En la página 460 se hace referencia a las funciones *lambda*, que sí tienen que ver con la programación funcional. También comentar que en *Python* existen librerías para hacer de él un lenguaje mas *funcional*.

Una referencia para saber mas sobre qué es exactamente la programación funcional está en la Wikipedia:

[https://es.wikipedia.org/wiki/Programación\\_funcional](https://es.wikipedia.org/wiki/Programación_funcional)

- En el tema 5.12, una forma alternativa (y temporal) al uso de *PYTHONPATH* es añadir, al inicio de nuestros programas, la dirección de las carpetas que deseemos se incluyan en la variable de sistema *path*, que contiene las rutas de búsqueda del intérprete *Python*. Lo lograremos de la siguiente manera:

```
import sys  
sys.path.append(r"C:\Users\flop\Desktop\Ficheros_Python")
```

Esta forma es muy limpia y se deshace al terminar la ejecución del programa (o la sesión del intérprete), aunque con ella no se mantendrá de forma permanente nuestra dirección en *path*, como sí logramos mediante la variable de sistema *PYTHONPATH*. He hecho uso del módulo *sys* para añadir la ruta hasta mi carpeta. Recordar que el lector deberá cambiar esa dirección por la suya particular.

- En el apartado 9.3.1, en la página 487, al configurar la ejecución externa (que no cambia el motor remoto como digo erróneamente en el libro) debemos desactivar la opción "Ocultar consola" (se muestra en la imagen derecha incorrectamente seleccionada) para que podamos visualizar posteriormente la salida de los programas. También, en la nota al pie 16 colocaremos *pythonw.exe* en lugar de *python.exe*, ya que es mas correcto al tratar con aplicaciones gráficas. Además, en la nota al pie 17 configuro la ejecución, dentro del IDE, de un intérprete *Python* independiente, algo que está al margen de la ejecución externa y del motor de *Python* (interno o externo) que usa *PyScripter*, por lo que no influye en la ejecución de nuestros programas y no nos serviría como alternativa para ello, salvo que añadiésemos los parámetros *\${ActiveDoc-Short} \${CmdLineArgs}* en el apartado *Comando*, algo que no parece del todo natural. Por lo tanto debemos distinguir entre motor interno (que viene internamente con *PyScripter* y no nos permite

la ejecución de aplicaciones gráficas), motor remoto (que tenemos en nuestra instalación estándar de *Python* ya que *PyScripter* lo busca ahí), ejecución externa (que nos permite ejecutar aplicaciones externas con una serie de parámetros, por lo que podremos ejecutar nuestros programas con el intérprete que le indiquemos) y, dentro de las herramientas, algo similar a la ejecución externa pensado para tener disponibles varias utilidades (entre las que puede estar un determinado intérprete de *Python*, o incluso varios).

De todas maneras, ejecutar externamente todos nuestros programas puede ser algo que no nos interese de forma práctica. Para que *PyScripter* haga uso de la librería *matplotlib* (y el motor remoto con el que hemos estado trabajando hasta el momento) simplemente debemos añadir la dirección de ésta al *path*, bien sea de forma permanente mediante la variable *PYTHONPATH* o de forma temporal usando lo visto en el punto anterior, que sería, considerando que el nombre de nuestro entorno es *Python3\_3\_5*, como sigue:

```
import sys
sys.path.append(r"C:\Users\flor\Anaconda3\envs\Python3_3_5\Lib\site-packages")
```

De esta forma añadimos no solo *matplotlib*, sino la dirección de todas las librerías de nuestro entorno *Python3\_3\_5*, a pesar de seguir usando (dentro de *PyScripter*) como motor remoto de *Python* el ubicado en nuestra instalación original, contenido en la carpeta *C:\Python33*.

- En la página 230, al hablar de **sobrecargar un método** (*overloading method*) en realidad me refiero a **sobrescribir un método** (*override method*). Sobrecargar una función (o un método) es la capacidad de definir varias/os de ellas/os con el mismo nombre pero con diferentes parámetros. Tendremos entonces la capacidad de realizar diferentes tareas dependiendo del número de parámetros (o de sus tipos) que le pasemos. La sobrecarga de funciones se realiza en Python de forma muy natural, como se puede observar en el tema 4.9.
- Si no trabajamos con *Windows* (y por lo tanto no podemos usar *PyScripter*) la opción de instalar *PyCharm* y un entorno virtual en *Anaconda3* con todo lo necesario (intérprete de *Python*, *PyQt* y *matplotlib*) es muy recomendable y cómodo. Tras crear el entorno tendremos todos los elementos almacenados en una carpeta. Posteriormente, al crear un nuevo proyecto en *PyCharm*, indicaremos la dirección del intérprete *Python* alojado en esa carpeta, tras lo cual podremos trabajar con todas las librerías incluidas en nuestro entorno.
- Página 196: en el tercer párrafo comento que los métodos se deberían nombrar como las clases pero con la primera letra en minúscula. No es correcto, se nombran igual que las funciones, con minúsculas y guiones bajos para separar palabras. En el libro no he seguido de forma estricta la guía de estilo PEP-8:

<http://pep8.org>

Seguirla por completo es algo altamente recomendable, sobre todo al principio para coger buenos hábitos.



- El uso de *eval()* que hago en muchos momentos en el libro puede ser sustituido por *int()* o *float()* para un funcionamiento con menos probabilidades de potenciales errores.
- Se me han colado varias *anotaciones de funciones* a lo largo del libro sin explicar exactamente qué son. Por ejemplo, en la página 283, en la cabecera de la función *elegir\_2* anoto los tipos de *numero* y *lista*. Las *anotaciones de funciones* las interpretaremos como una característica que nos permite añadir metadatos a los parámetros (o al valor de retorno) de las funciones. En nuestro caso añadimos el *tipo* de los parámetros de entrada.
- Cuando hablo de objetos *hashable* en la página 31, 327 y 328 estaría bien indicar que los conjuntos no son *hashable*, ya que son contenedores mutables. Por lo tanto, los conjuntos no se pueden usar como llave en los diccionarios. Existe, no obstante, una variante inmutable de los conjuntos (y que por lo tanto si se podría usar como llave en diccionarios) cuya clase tiene de nombre *frozenset*.
- En cadenas tenemos también un método *index()* está. Es similar a *find()*, con la salvedad de que se lanza una excepción si no se encuentra lo buscado.
- Comentar que una alternativa al uso de *Anaconda3* para la gestión de paquetes en *Python 3* es *pip3*, que viene por defecto a partir de la versión 3.4.

La fe de erratas (donde incluyo errores de todo tipo, muchos de ellos sintácticos o gramaticales) es la siguiente:

- Pág. 15: "...la reunión de 8 bits es un **byte**."
- Pág. 16: "Dos tablas de las distintas unidades que podemos encontrar **son las siguientes**:"
- Pág. 22: cuando hago referencia a *py2to3* realmente me estoy refiriendo a la herramienta **2to3** (<https://docs.python.org/3/library/2to3.html>)
- Pág. 24: 1) "Al finalizar haremos clic en la flecha..."  
2) El fichero de nombre *python* es un ejecutable (*python.exe*), algo que no aparece de forma explícita en el libro.
- Pág. 29: "...vamos al menú File → New File y..."
- Pág. 30: 1) "...hacemos clic en File → Save As, tras ..."  
2) "...las dos líneas de texto, **se ha** salido del modo consola..."
- Pág. 31: 1) "...lo hacemos, por defecto, **en** nuestra carpeta de usuario."  
2) La imagen inferior no corresponde exactamente con la ejecución del comando *Python Primer\_script.py* sino con la de *Primer\_script*.  
3) Para ejecutar sin problema los ficheros *Python* debemos añadir la extensión, por lo que teclearemos *Primer\_script.py* en los dos casos en los que la omití .
- Pág. 32: "python C:\Users\flop\Desktop\Ficheros\_Python\Primer\_script.py"
- Pág. 34: "...(podemos ser nosotros mismos pasado un tiempo)..."
- Pág. 36: al hacer referencia a la "imagen de la izquierda" debería poner "imagen **inferior**" o "**siguiente** imagen".
- Pág. 47: "Por ejemplo *numero\_de\_usuarios*, **valor\_temperatura**."

- Pág. 49: cuando indico "esquema superior" debería poner "esquema **anterior**".
- Pág. 59: en "Si se le pasa una cadena con un número entero en ella." sobra el punto final.
- Pág. 61: "...sin decimales, truncando el número hacia uno **entero**."
- Pág. 64: "Devuelve ab" es en realidad "Devuelve **a<sup>b</sup>**".
- Pág. 72: 1) "Imprime en **pantalla**", sin 's' final.  
2) "...hemos indicado mediante **el** parámetro *sep*."
- Pág. 73: la nota al pie 36 es redundante y podría ser eliminada.
- Pág. 74: 1) "EOL (*End Of Line*) en inglés."  
2) En:  

```
>>> print("Nombre\tEdad\tProfesión")
NombreEdad      Profesión
falta algo de espacio entre Nombre y Edad.
```
- Pág. 78: "...entero correspondiente al carácter **c** en el código **Unicode**."
- Pág. 79: "Es una **built-in function** con el siguiente formato:"
- Pág. 81: "...o **trabajar** con el intérprete de forma interactiva."
- Pág. 86: "Tenemos también un tipo *complex* para tratar los números..."
- Pág. 88: "... si hubiésemos hecho en lugar de *modelo = 22*, *modelo = modelo + 1* se **habría** creado..."
- Pág. 96: "**a = ((21 \* 4) + 3) \*\* 2**" en lugar de "**a = ((21 \* 4) + 3) ^ 2**". Lo mismo pasa en la página siguiente.
- Pág. 98: "En **la** segunda línea tenemos una expresión".
- Pág. 99: 1) "Muy diferente **habría** sido si hubiésemos hecho:"  
2) "Si **b** hubiese estado definido, la expresión al menos nos **habría** devuelto un dato válido."  
3) "La respuesta a estas preguntas está en el siguiente **tema**".
- Pág. 102: "6. Formando parte de lo que se **denominan** *expresiones condicionales*".
- Pág. 103: "...nos indican las posibles direcciones que **puede** llevar el programa..."
- Pág. 104: "# El resultado en este caso es...nada, al ser **a < 7**."
- Pág. 105: "**Habríamos** conseguido el mismo efecto tecleando *Ctrl+n* o..."
- Pág. 116: 1) En el código:  

```
print("El número introducido es cero") if numero == 0 else print("El número introducido no es cero")
```

  
2) Cuando hablo de :  
*expresión1 if expresión booleana else expresión2*  
me estoy refiriendo al *ternary operator* (operador ternario).
- Pág. 130: la salida que indico al introducir los números 99, 1 y 1 no es la correcta. Lo es la siguiente:  

```
>>>
El programa irá pidiendo números y los irá sumando hasta que la suma supere el valor 100
La suma total al superar los 100 ha sido: 101
>>>
```
- Pág. 135: "Donde **12 es** el valor de **x**".
- Pág. 136: 1) "...le dota de **gran** potencia dado..."  
2) "...varias páginas desde **las** que obtener esos números..."  
3) "...en el caso **de** que nos fuese..."
- Pág. 137: 1) "Devuelve un número entero aleatorio entre los números enteros **comienzo** y **final**, excluyendo este último, pero solo elige entre los números que estén en la

serie siguiente:

$comienzo, comienzo + paso, comienzo + 2 * paso, \dots$

Podremos no incluir el parámetro **paso** (por defecto valdrá 1) solo si **comienzo < final**.

La función **randrange()** es más completa para tratar los enteros que **randint()**, ya que podremos emular a esta última de la siguiente manera:"

2) La función **random.choice()** tiene como parámetro de entrada en realidad lo que se denomina una *secuencia*. Las *cadena*s son un ejemplo de *secuencias*.

- Pág. 139: en el código:

```
>>> random.random(100)           # La función random no necesita ningún argumento.
>>> random.uniform(10, 110, 2)   # La función uniform solo necesita dos argumentos.
```

- Pág. 140: 1) En la tabla que aparece, *a*, *b*, *comienzo*, *final* y *paso* dentro de la columna "Observaciones" deberían aparecer en cursiva.  
2) Comento al final del último párrafo que hay que pulsar dos veces *Enter* tras el cero. No es estrictamente necesario.
- Pág. 143: 1) Vuelvo a comentar que hay que pulsar dos veces *Enter* tras el cero. Nuevamente no es necesario.  
2) "En resumen, lo que hemos hecho en este **tema** ha sido..."
- Pág. 147: 1) En la parte superior izquierda de la hoja aparece "**cabecera**" cortado.  
2) La última frase del primer párrafo sería mas correcta de la siguiente manera: "El bloque de instrucciones podrá incluir (una o varias veces, aunque no es obligatorio) un comando especial llamado *return*, que tiene el siguiente formato:"
- Pág. 152: a pesar de que en el fichero de código está solucionado, en la última imagen de la página aparece "Introduce" en lugar de "**Introduce**".
- Pág. 154: "En el **tema** anterior di una primera explicación de lo que son las..."
- Pág. 155: 1) "Es algo que comprobamos en la práctica en el ejemplo del **tema** anterior"  
2) La imagen asociada a *ejemplo\_variables\_locales.py* no es correcta. Lo sería la siguiente:

```
1
2 def funcion1():
3     x = 5
4     x = x + 2
5     print("La variable x dentro de funcion1 vale: ", x)
6
7 def main():
8     x = 1
9     funcion1()
10    print("La variable x dentro de main vale:      ", x)
11
12 main()
13
```

- Pág. 157: el error salta realmente en la línea 8 del código, por el mismo motivo indicado en el libro.
- Pág. 158: "De no haberla usado en el *main()*, nos **habría** dado error".
- Pág. 160: 1) "...que almacenamos en las variables *numero\_1* y *numero\_2*."  
2) Tal como lo tengo, en el dibujo no debería estar *calculo\_dos\_datos\_2* en cursiva. Además, el código de la función (que incluye un *return*) debería estar con sangría y *res* no es necesario que esté entre paréntesis.
- Pág. 164: lo de "1. Paso 1" y similares es un tanto redundante. Lo mismo pasa en las tres páginas siguientes.

- Pág. 169: 1) En " asignado por el argumento posicional" hay un tabulador perdido.  
2) "...y en un fichero nuevo **pegaremos** (sin guardar) su contenido..."
- Pág. 172: "...en cuyo caso sí que nos **habría** dado un error debido a que la variable *sum* la hubiese solapado y no podríamos..."
- Pág. 174: 1) "Como solo quedaría **a** por defiiir, coge su valor..."  
2) " No se puede intentar dar dos valores a **a**"
- Pág. 176: "Hice referencia en **temas** anteriores a la..."
- Pág. 180: en el fichero de código *ejemplo\_factorial.py* aparece "Introduce" en lugar de "**Introduce**", algo que se puede apreciar en la imagen.
- Pág. 184: 1) En el esquema de la parte superior de la hoja aparece "Renaul" en lugar de "**Renault**".  
2) " En su último **tema** tratamos los conceptos de..."
- Pág. 186: en la última línea del tercer párrafo, "algo que veremos más adelante podemos modelizar mediante POO" estaría mas correcto como "algo que, **como** veremos más adelante, podemos modelizar mediante POO'.
- Pág. 187: "...ejemplo similar al que comentamos en el **tema** anterior."
- Pág. 195: "...y dentro de él la opción **sep** para que..."
- Pág. 196: 1) " En los anteriores **temas** aprendimos a:"  
2) " Basándonos en el programa que llevamos desarrollando todo el **capítulo**..."
- Pág. 198: se me ha colado una coma en **setNombre\_del\_campo\_a\_escribir**.
- Pág. 200: 1) En el intérprete de *Python* de la imagen superior aparece la versión 3.3.2, algo que como ya comenté con anterioridad no influye para nada en el ejemplo.  
2) Sobra un "en" en "...observamos que con un determinado icono nos representa los campos..."
- Pág. 201: 1) "...impedimos que desde el *main()* se le pueda..."  
2) "...llama al método privado *sueldo()* mediante el formato..."
- Pág. 202: estaría mas ortodoxo así: "**UML (Unified Modeling Language)**".
- Pág. 203: "Es lógico que no aparezca **self** ya que para usar..."
- Pág. 204: 1) El título del tema sería mas claro como: " 5.6 OBJETOS MUTABLES E INMUTABLES PASADOS A FUNCIONES".  
2) "En los primeros **capítulos** del libro vimos..."
- Pág. 205: "A lo largo del presente **capítulo** hemos aprendido a crear..."
- Pág. 210: "El objeto **a** aparece como *n/d*..."
- Pág. 211: "...fue comentada en el **tema** 4.6 al hablar de las pilas..."
- Pág. 214: "...a pesar de no ser tan cómoda en casos sencillos como **la** anterior..."
- Pág. 217: el símbolo de la viñeta es un círculo en lugar del triángulo habitual.
- Pág. 219: "...se inicializa con una estrella su campo **\_\_estrellas**, como queríamos."
- Pág. 220: " Para finalizar este **tema** haré notar..."
- Pág. 224: " En el **tema** anterior expliqué **los** conceptos de herencia..."
- Pág. 230: " En los dos **temas** anteriores vimos cómo podemos..."
- Pág. 231: "aprender más cosas sobre lo visto en **temas** anteriores..."
- Pág. 233: "...las usaremos en el siguiente **tema**."
- Pág. 234: "Al final del **tema** 5.7 comenté que..."

- Pág. 242: 1) "...he introducido el parámetro **nombre** para poder dar..."  
2) Sobra un punto en "... un condicional **if-elif**, usamos..."
- Pág. 249: "...en la propia estabilidad del programa."
- Pág. 251: "...podemos pensar en **almacenar** todas las muestras en formato texto en una cadena, separándolas..."
- Pág. 260: "El formato de los métodos será representado en **base al** estándar *UML*".
- Pág. 261: tal y como tengo los demás formatos, los dos puntos del formato de *isupper()* no deberían estar en negrita.
- Pág. 264: siguiendo el formato aplicado a los métodos, el de *replace()* debe ser así:  
***replace(a: str, d:str): str***
- Pág. 265: 1) En nota al pie 13: " Nuevamente incluiremos el carácter '□' para indicar espacio en blanco".  
2) En la explicación de los 4 métodos la palabra "ancho" debe ir en cursiva.
- Pág. 266: estaría mas correcto, en la nota al pie 14, lo siguiente: "...usaríamos '{' y '}'" para ello."
- Pág. 267: cuando hago referencia a la "numeración manual" es a la que he llamado "por posición".
- Pág. 268: al indicar los posibles valores del parámetro *tipo* me he olvidado de incluir (a pesar de que luego lo uso en los ejemplos) la opción '**f**' para los números reales (*float*).
- Pág. 273: 1) "En la *lista\_10* tenemos la expresión *x+y* y dos bucles *for* anidados, el más interior recorriendo la *lista\_5* y el más **exterior** la *lista\_2*."  
2) "...y a *[8,10]* para el **interior**..."
- Pág. 279: el texto del apartado 6.2.3 está justo al revés. Lo correcto sería:  
"Veremos en este apartado la relación entre funciones y listas. En primer lugar, conoceremos una serie de funciones predefinidas útiles para efectuar cálculos sobre una lista. Posteriormente observaremos el comportamiento de las listas al ser pasadas como argumentos de entrada a una función, o ser su valor de salida."
- Pág. 283: quizá es algo redundante hablar de "argumento de entrada", pudiéndose eliminar las dos últimas palabras.
- Pág. 284: "...la variable **numero** de la función *elegir\_2()*..."
- Pág. 288: 1) A pesar de que en el fichero de código *ejemplo\_funciones\_y\_listas\_2.py* está ya corregido, en la salida que indico al inicio de la página aparece "fucion" en lugar de "**función**".  
2) El antepenúltimo párrafo está cortado en dos.
- Pág. 298: cuando hablo de "...hacer recorridos personalizados..." estaría mejor decir "hacer recorridos **adecuados**..."
- Pág. 299: en la nota al pie 43 sobran los dos últimos puntos.
- Pág. 302: en el esquema del final de la hoja, la flecha que sale hacia abajo desde lista1 es en realidad la de lista2.
- Pág. 305: el segundo y el tercer bloque de código es el mismo, por lo que está repetido.
- Pág. 307: cuando hablo de *ejemplo\_tupla\_tridimensional.py* en realidad me refiero a ***ejemplo\_lista\_tridimensional.py***.



- Pág. 308: 1) Cuando comento que " Existen otros que , por motivos como la seguridad de los datos o la velocidad en su tratamiento, serán más eficaces que las listas para operar con la información" me refiero en casos determinados, no de forma genérica.  
2) "Al primero de ellos dedicaremos este **tema**."  
3) Cuando comento "...casos en los que no está permitido alterar (sea o no de forma accidental) el elemento donde almacenamos los datos." me refiero a la estructura general (en este caso la tupla) ya que, como vemos mas adelante, ésta puede tener elementos mutables donde se nos permitiría alterar su contenido de forma individual.  
4) Sería mas correcto decir "entre paréntesis" que "entre dos paréntesis".
- Pág. 309: lo correcto es: "En el primer caso lo que tenemos es una lista y en segundo un entero", sobrando un "él" en el texto.
- Pág. 316: " Si hubiésemos hecho lo siguiente para crear una tupla copia de *t15* **habríamos** cometido un error:"
- Pág. 317: 1) Lo correcto es poner: "..sino una nueva referencia al objeto que queremos copiar, que en este caso no es lo deseado.", sobrando un "lo" en el texto.  
2) "Las tuplas que vimos en el **tema** anterior..."
- Pág. 319: "...una estructura de datos que veremos en el siguiente **tema**."
- Pág. 324: "Si no lo está, devuelve un error de tipo **KeyError**."
- Pág. 326: defino *c13* de forma errónea (justo de la manera que comenté con anterioridad que no era correcta). Estaría "bien" de la siguiente manera:

```
>>> c13=set()
>>> c13.pop()
Traceback (most recent call last):
  File "<string>", line 301, in runcode
  File "<interactive input>", line 1, in <module>
KeyError: 'pop from an empty set'
```
- Pág. 329: 1) En lugar de "Para añadir o actualizar un nuevo elemento..." lo correcto sería "Para añadir un nuevo elemento (o actualizarlo)..."  
2) "Si lo está, se **actualiza** su valor."
- Pág. 331: "...con la particularidad de que **actúan** sobre las llaves:"
- Pág. 336: en la tabla falta un "Si" en la columna de las listas para el operador índice, ya que no aparece.
- Pág. 342: "...que dependerá **de** si vamos a leerlo, a escribirlo, o a añadirle elementos."
- Pág. 343: "Por ejemplo, al abrir el fichero en modo lectura, este **se** coloca al inicio..."
- Pág. 344: "...buscará el fichero en el directorio de trabajo **en el** que nos encontremos."
- Pág. 345: 1) "...mediante el número entero **desp** tomando como referencia..."  
2) "Nos devuelve una lista **con** el contenido del fichero..."
- Pág. 347: sobran los puntos suspensivos al final del tercer párrafo de texto.
- Pág. 355: "...éste hará que el contenido del fichero ***ejemplo\_ficheros\_5.txt*** sea el que..."
- Pág. 356: en 7.1.2.1, ***read***, ***FileNotFoundError*** (dos veces), ***write*** y ***append*** no aparecen en cursiva.
- Pág. 359: en la salida "Alfredo" no aparece de forma correcta, sino como "b'fredo".

- Pág. 364: 1) "...almacenar de forma directa el número **123.456** o cualquier..."  
2) Nota al pie 30: "... a ambos procesos también se les denomina *marshalling* y *flattening*, **respectivamente**."
- Pág. 369: mas correcto sería: "El método *sync()* vacía la caché. Es llamado por el método *close()* cuando cerramos el fichero".
- Pág. 372: 1) "...necesitaremos conocer lo tratado en el siguiente **tema**."  
2) " En este **tema** aprenderemos a tratar los errores que..."  
3) Lo correcto es: " Hemos visto múltiples ejemplos de ello hasta ahora, y sabemos que el intérprete nos informa mediante un determinado mensaje de error"  
4) Lo correcto es: "Como vemos, obtenemos la notificación de tres..."
- Pág. 381: 1) El símbolo correcto del kilogramo es **kg**, en minúscula. En el código uso Kg, con la 'K' mayúscula, lo que sería erróneo.  
2) Al final del código: "print("Seguimos **con** el flujo del programa.")".
- Pág. 384: mas apropiado es: "...intercalando unas líneas discontinuas entre las palabras almacenadas."
- Pág. 386: 1) "...donde los datos **se** introducen mediante casillas, botones y varios elementos más, dentro de lo que se denomina un..."  
2) Cuando hablo sobre GUI diciendo que es "*Graphics User Interface*" debería ser "**Graphical User Interface**".
- Pág. 387: "...de enlace entre **éste** (escrito en *Python*) y las librerías..."
- Pág. 391: "...serán dejadas para sucesivos **temas**."
- Pág. 392: 1) En nota al pie 7: "...como comentamos en **capítulos** anteriores."  
2) En nota al pie 10: comento que más adelante veremos cómo indicar que la ventana tenga un estilo determinado. No lo hago. El lector curioso puede mirar el menú siguiente de *Qt Designer*:  
*Settings*→*Preferences*...→*Forms*→*Print/Preview Configuration*  
y seleccionar alguna de las opciones para *Style*.
- Pág. 393: 1) En la definición de la clase *primer\_ejemplo\_GUI* no sigo *PEP-8*.  
2) "...y el programa, es decir: un botón, un selector..."
- Pág. 394: "El método ***setWindowTitle()*** nos permite..."
- Pág. 396: "Como ya comenté, *Qt Designer* es **un** programa usado..."
- Pág. 400: "**Lo llevaremos** hasta nuestro contenedor, y daremos valor *100*...".  
El icono al que apunto con la flecha en la última imagen de la página no es "Edit Signals/Slots" sino "Edit Widgets", por lo que debería apuntar al icono del al lado.
- Pág. 401: "Nos aparecen los dos elementos que conectamos:". Sobra el "Donde".
- Pág. 402: 1) La extensión *.ui* viene de "user interface".  
2) Cuando hablo de *XML* como un lenguaje relacionado con el diseño de páginas *web* sería más preciso indicar que es un lenguaje de marcado para intercambio de datos.
- Pág. 405: en la primera línea de la página debería decir "... creado en *Primer\_ejemplo\_GUI\_qtdesigner.py* y ejecutamos..."
- Pág. 406: "La **clase** *ui\_Form* de *Fichero.py* podría ser *ui\_Dialog* o..."

- Pág. 411: 1) Mas adecuado así: "**También** cualquier otro que queramos que tenga distribución cuadrangular."  
2) Estaría mejor de la siguiente forma: " En esta disposición *formulario* tendremos dos..."
- Pág. 414: "...es que al hacer clic **sobre** ellos realicen una operación."
- Pág. 415: "Donde cortar, copiar, borrar, seleccionar todo..."
- Pág. 416: "... como aparece en la imagen **superior**."
- Pág. 418: "**Observaremos** que, por defecto, tanto la opción checkable..."
- Pág. 428: "En el superior y el inferior izquierda podemos seleccionar..."
- Pág. 431: "Sobre un rectángulo en forma de barra..."
- Pág. 453: "En el **tema** anterior vimos como, de cara a..."
- Pág. 457: 1) "Los códigos no serán comentados tan detalladamente como en **temas** anteriores"  
2) " Este **tema** puede también interpretarse como..."  
3) Lo correcto sería: "...y, si lo hace *MainWindow*, una ventana principal (clase *QMainWindow*)."  
4) En el último párrafo de la hoja debería poner "...a dos números **reales** introducidos mediante teclado".
- Pág. 471: cuando hago referencia a la clase *QMdiArea*, ésta se refiere a interfaz multidocumento, no a elementos multimedia.
- Pág. 485: el IDE científico que viene con la instalación de Anaconda3 es **Spyper**, no Spider.
- Pág. 487: "...configuraremos *PyScripter* para que haga uso del entorno **Python3\_3\_5** que hemos creado..."
- Pág. 492: " En este caso repetimos **el** proceso anterior para la..."
- Pág. 502: la nota al pie 21 es " Que es **la** que tiene mayores valores de las cuatro".
- Pág. 529: "...El segundo hace uso de **set\_xticklabels()** para indicar..."