

4

COMENZANDO A PROGRAMAR

4.1. HOLA MUNDO

Es una especie de tradición comenzar a mostrar el funcionamiento de un lenguaje de programación haciendo un programa que muestre el texto “Hola Mundo”.

Hacer esto en Python es muy sencillo; solo es necesario un programa de una línea:

```
print ("Hola Mundo")
```

<i>Salida del programa:</i>

Hola Mundo

En este ejemplo la palabra *print* es una instrucción propia de Python (una *sentencia*) que sirve para mostrar un texto en la pantalla, y el texto “Hola Mundo” entrecomillado es lo que se llama una *cadena de texto* o *cadena (string)*. En conjunto, la expresión resultante significa algo así como “imprime en la pantalla el texto ‘Hola Mundo’”.

Usaremos nuestro editor de texto para escribir eso en un archivo y lo guardaremos con el nombre *holamundo.py*. En los sistemas tipo Windows será necesario que el nombre que elijamos acabe siempre en “.py”, porque esa es la forma que tiene Windows de reconocer que el archivo es un *script* de Python.

NOTA

A la parte del nombre de un fichero que va después del punto se le suele llamar “extensión”.

NOTA

En Mac el terminal debería estar en el menú Aplicaciones -> Utilidades -> Terminal. El Windows aparece como 'Windows Terminal' / 'Windows PowerShell' (antes 'Símbolo del sistema').

Ahora deberemos abrir un terminal.

En nuestro terminal podremos ver lo que se llama una "línea de comandos" (o línea de órdenes). Su apariencia varía de un sistema a otro, pero consiste en una interfaz de interacción con los programas a través de instrucciones en modo texto.

NOTA

El uso de la línea de comandos va más allá de este curso. Se puede encontrar más información, por ejemplo, en estas direcciones:

- Windows: <http://windows.microsoft.com/es-xl/windows/command-prompt-faq>
- Mac OS X: <http://foro-mac.com.ar/tutorial-como-usar-la-terminal-en-mac>
- Linux: https://es.wikibooks.org/wiki/Manual_de_consola_Bash_de_Linux

En esa línea de comandos, usando las instrucciones del sistema, debemos desplazarnos al directorio donde está el *script*, y ejecutarlo escribiendo:

```
python holamundo.py
```

NOTA

En Windows no es necesario escribir "python", escribiendo solo "holamundo.py" es suficiente. (Suponiendo que "holamundo.py" sea el nombre que le hemos dado a nuestro programa).

Esto, si no nos hemos equivocado en nada, hará que se nos muestre el texto "Hola Mundo" en la terminal en la que estamos ejecutando programa.

¿Qué hemos hecho? Con esta orden lo que hacemos es ejecutar el intérprete de Python, pero esta vez pasándole como parámetro el nombre del programa que hemos escrito, para que lo ejecute directamente en lugar de iniciar una sesión interactiva como vimos anteriormente.

Ejecutar: invocar a un programa para que haga lo que sea que debe hacer.

En entornos gráficos los programas se ejecutan haciendo doble clic sobre su icono; en línea de comandos se hace escribiendo su nombre.

Podemos probar a cambiar el texto (sin olvidar las comillas) y guardar de nuevo el archivo con los cambios y veremos que, al ejecutar de nuevo el *script*, se imprime en pantalla lo que le indiquemos.

NOTA

Naturalmente, si estamos usando el IDLE, también es posible ejecutar nuestros programas directamente con la opción *Run Module* del menú *Run*.

En algunas plataformas Windows, al ejecutar el programa haciendo doble clic sobre su icono, se abre un terminal con resultado de la ejecución y se cierra instantáneamente, sin darnos tiempo a ver nada.

Hoy una solución, no muy elegante pero útil, es añadir la línea `input()` al final del programa, con lo que este esperará a que se pulse la tecla **ENTER** antes de que se cierre la ventana.

```
1 #Nuestro primer programa con Python
2
3 # Escribe en pantalla la expresión 'Hola Mundo'
4 print ("Hola Mundo")
5
6 input()
```

Salida del programa:

Hola Mundo

NOTA

`input()` es una función de Python que sirve para obtener entradas desde teclado, y la veremos un poco más adelante.

Naturalmente, podemos ejecutar más de una instrucción ubicándolas en líneas consecutivas. El intérprete de Python las ejecutará una a una de arriba abajo.

```
1 #Variante del programa holamundo.py
2
3 print ("Hola") # Escribe en pantalla la expresión 'Hola'
4
5 print ("Mundo") # Escribe en pantalla la expresión 'Mundo'
6
7 input() # Esta línea permite ejecutar 'holamundo.py' en el entorno gráfico
```

Salida del programa:

Hola
Mundo

La línea en blanco entre las instrucciones está ahí simplemente por claridad. El programa funcionará exactamente igual sin ella, o añadiendo varias. Python ignora las líneas en blanco y, a la hora de programar, nosotros deberíamos usarlas cuando sea necesario para que el programa sea más legible.

Este es un tema más importante de lo que parece a primera vista. A menudo, al escribir un programa, nos olvidamos de que llegará el día en que queramos leerlo o modificarlo de nuevo, o de que puede que otras personas quieran leer ese código. Por eso debemos hacer que sea siempre lo más legible y comprensible posible.

NOTA

La instrucción *print* introduce automáticamente un retorno de carro después del texto que se muestra en pantalla, por lo que el siguiente texto se imprime en otra línea. Si queremos evitar que ponga ese retorno de carro, solo tenemos que acabar la instrucción con una coma, haciendo algo como `print("texto",)`.

Una herramienta para ayudarnos en esto son los comentarios. Se puede añadir comentarios a un programa, que serán completamente ignorados por el intérprete, pero que pueden ser muy útiles para que los seres humanos entiendan el código.

Un comentario se pone precedido del símbolo “almohadilla” (#).

Todo lo que haya después de la almohadilla será ignorado por el intérprete, de modo que se puede poner un comentario a continuación de una línea de código.

```
#Nuestro primer programa con Python

# Escribe en pantalla la expresión 'Hola Mundo'
print ("Hola Mundo")

input()# Esta línea permite ejecutar 'holamundo.py' en el entorno gráfico
```

Es importante hacer notar que el comentario será ignorado aunque se trate de código de Python, lo que a veces puede ser muy útil cuando estamos haciendo pruebas y queremos “desactivar” alguna orden:

```
1 #Variante del programa holamundo.py
2
3 # Escribe en pantalla la expresión 'Hola Mundo'
4 print ("Hola Mundo")
5
6 # Pero esto no imprime nada, porque está marcado como comentario
7 #print ("Hola Mundo")
```

Salida del programa:

Hola Mundo

4.2. ¿QUÉ HEMOS VISTO EN ESTE TEMA?

Los pasos necesarios para escribir y ejecutar nuestro primer programa.

4.2.1. Tareas sugeridas

- ✓ Para empezar, deberíamos (si no lo hemos hecho ya) probar a escribir nuestro primer “Hola Mundo”, para después ejecutarlo y ver el resultado.
- ✓ Podemos probar a escribir varios textos diferentes y añadir al programa varias órdenes *print* diferentes.
- ✓ Intentar ejecutarlos alguna vez desde el entorno gráfico (explorador de archivos) y asegurarnos de que veáis el resultado.