Desconocido

## Epílogo: A dónde ir a partir de ahora



Has aprendido algunos conceptos fundamentales de programación en tu recorrido por Python, pero hay mucho más por descubrir, ya sea con Python o utilizando otros lenguajes de programación. Aunque Python es increíblemente útil, no siempre es la mejor herramienta para cada tarea, así que no tengas miedo de probar otras formas de programar tu ordenador.

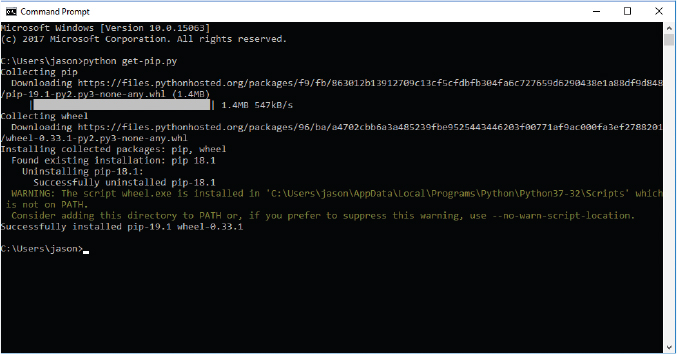
Si quieres seguir con Python y buscas libros más avanzados para leer, la wiki de Python (página de libros) es un buen lugar para empezar: [*https://wiki.python.org/moin/PythonBooks/*](https://wiki.python.org/moin/PythonBooks/) .

Si sólo quieres explorar más de lo que Python puede hacer, tiene un montón de módulos incorporados. (Esta es la filosofía "pilas incluidas" de Python. Consulta la documentación de Python para ver la lista completa de lo que hay disponible: [*https://docs.python.org/3/py-modindex.html*](https://docs.python.org/3/py-modindex.html) .) Además, hay un gran número de módulos que proporcionan gratuitamente programadores de todo el mundo. Por ejemplo, puedes probar *Pygame* ( [*https://www.pygame.org*](https://www.pygame.org) ) para el desarrollo de juegos, o *Jupyter Notebooks* ( [*https://jupyter.org*](https://jupyter.org) ), un entorno basado en la web para editar y ejecutar código Python en el navegador. Puedes consultar estos y otros módulos en https://pypi.org [*.*](https://pypi.org) Para instalar estos módulos, necesitarás utilizar una herramienta llamada pip , que veremos brevemente a continuación.

### Instalar Python pip en Windows

Siempre que hayas instalado Python 3.10 o posterior, pip debería estar instalado por defecto. Para instalar pip en Windows, si utilizas una versión anterior de Python, puedes descargar el script *get-pip.py* de https://bootstrap.pypa.io/get-pip.py [*.*](https://bootstrap.pypa.io/get-pip.py)

Guarda el archivo en tu carpeta de inicio y, a continuación, abre un símbolo del sistema (haz clic en **Inicio** e introduce cmd en el cuadro de búsqueda). Para instalarlo, escribe  **python get-pip.py**  .

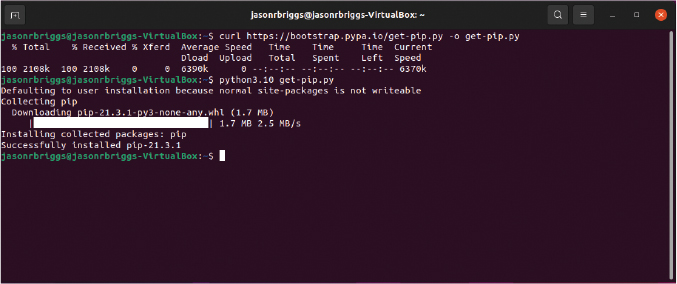


### Instalar pip Python en Ubuntu

Para instalar pip en Ubuntu Linux, necesitarás la contraseña de administrador del sistema. Abre un terminal e introduce los siguientes comandos (asegúrate de cambiar el número de versión en el siguiente comando según sea necesario):

sudo apt install python3.10-distutils  
sudo apt install curl  
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py  
python3.10 get-pip.py

Con los dos primeros comandos, es posible que aparezca un error diciendo que ya están instalados; puedes ignorarlo sin problemas.

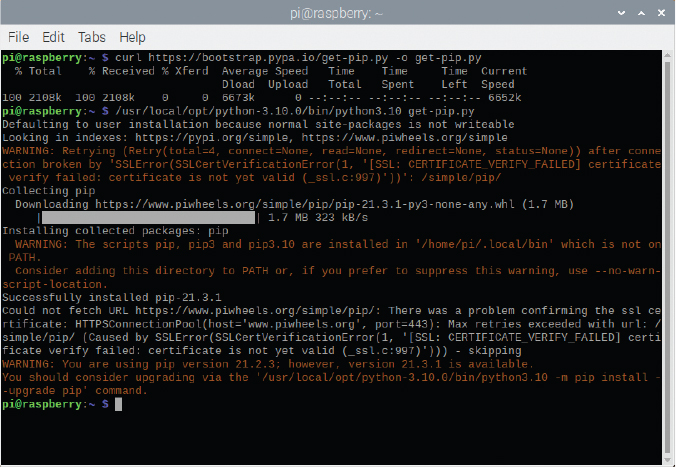


### Instalación de Python pip en la Raspberry Pi

Para instalar pip en la Raspberry Pi, abre un terminal e introduce los siguientes comandos:

curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py  
python3.10 get-pip.py

Puedes ignorar con seguridad cualquier mensaje de advertencia que aparezca.

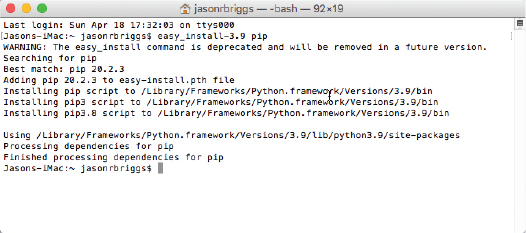


### Instalar pip Python en macOS

Siempre que hayas instalado Python 3.10 o una versión más reciente, pip debería estar instalado por defecto. Si utilizas una versión anterior de Python, puede que tengas que instalar pip abriendo un terminal y ejecutando el siguiente comando:

easy\_install-3.9 pip

Dependiendo de la versión de Python que tengas instalada, puede que necesites introducir una versión diferente de easy\_install (como easy\_install-3.7 , por ejemplo).



### Probar PyGame

Para hacerte una idea de cómo funciona *PyGame* , instálalo primero utilizando pip . Abre un símbolo del sistema (en Windows, introduce cmd en el cuadro de búsqueda; en Ubuntu, Raspberry Pi o macOS, busca Terminal ) e introduce el siguiente comando (tu número de versión puede ser diferente):

pip3.10 install pygame

Nota

*Dependiendo de tu versión de Windows y de Python, puede que esto no funcione. Si aparece un error como éste*

'pip' no se reconoce como comando interno o externo,  
programa operable o archivo por lotes.

*entonces intenta ejecutar los siguientes comandos:*

cd %HOMEPATH%  
AppData\Local\Programs\Python\Python310\python -m pip install pygame

Escribir código con PyGame es un poco más complicado que utilizar tkinter . Por ejemplo, en el [Capítulo 10](ch10.xhtml#ch10) , mostramos una imagen utilizando tkinter con este código:

from tkinter import \*  
tk = Tk()  
canvas = Canvas(tk, width=400, height=400)  
canvas.pack()  
myimage = PhotoImage(file='c:\\Users\\jason\\test.gif')  
canvas.create\_image(0, 0, anchor=NW, image=myimage)

Para mostrar una imagen con PyGame , puedes utilizar lo siguiente:

import pygame  
pygame.init()  
display = pygame.display.set\_mode((500, 500))  
img = pygame.image.load('c:\\Users\\jason\\test.gif')  
display.blit(img, (0, 0))  
pygame.display.flip()

Tras importar el módulo pygame , llamamos a la función init ; esto inicializa el módulo para que pueda ser utilizado. A continuación, configuramos  la visualización, pasando una tupla para la anchura y la altura (500 píxeles de ancho por 500 píxeles de alto). Observa que los paréntesis adicionales aquí son importantes: primero, paréntesis para la propia función ( set\_mode(...) ), y luego paréntesis para una tupla ( 500, 500 ).

A continuación, cargamos la imagen, cuya referencia (o etiqueta) es la variable img (recuerda que, dependiendo del sistema operativo que utilices, puede que tengas que cambiar la ruta de esa imagen). A continuación, dibujamos la imagen en la pantalla utilizando la función blit , pasándole la variable img y una tupla que contiene la posición superior izquierda de la imagen ( 0, 0 ). La imagen aún no se mostrará en la ventana; la siguiente función de la última línea -donde flip la muestra- hace que la imagen aparezca realmente. Esta línea le está diciendo a Pygame que redibuje la ventana de visualización.

Nota

*Si ejecutas esto fuera de IDLE, tendrás que añadir unas líneas adicionales al final:*

while True:  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT:  
 raise SystemExit

Este código es para evitar que la ventana se cierre inmediatamente después de mostrar la imagen.

### Otros juegos y programación gráfica

Si quieres hacer algo más con juegos o programación de gráficos, verás que hay muchas opciones disponibles, no sólo con Python. Aquí tienes sólo algunas:

* Scratch ( [*http://scratch.mit.edu*](http://scratch.mit.edu) ), una herramienta para desarrollar juegos y animaciones. (Es un lenguaje de programación visual basado en bloques, por lo que es bastante diferente de programar con Python).
* Construct3 ( [*https://www.construct.net*](https://www.construct.net) ), una herramienta comercial para crear juegos en un navegador.
* Game Maker Studio ( [*https://www.yoyogames.com*](https://www.yoyogames.com) ), otra herramienta comercial para crear juegos.
* Godot ( [*https://godotengine.org*](https://godotengine.org) ), un motor de juegos gratuito para crear gráficos en 2D y 3D.
* Unity ( [*http://unity.com*](http://unity.com) ), otra herramienta comercial para crear juegos.
* Unreal Engine ( [*https://www.unrealengine.com*](https://www.unrealengine.com) ), otra herramienta comercial para crear juegos.

Una búsqueda en Internet te descubrirá una gran cantidad de recursos que te ayudarán a iniciarte en cualquiera de estas opciones, o al menos te mostrarán las posibilidades que tienes si sigues programando en el futuro.

### Otros lenguajes de programación

Si te interesan otros lenguajes de programación, considera echar un vistazo a algunos de los más populares: JavaScript, Java, C#, C, C++, Ruby, Go, Rust y Swift (aunque hay muchos más). Haremos un breve recorrido por estos lenguajes y veremos cómo sería un programa Hola Mundo (como la versión de Python con la que empezamos en el [Capítulo 1](ch01.xhtml#ch01) ) en cada uno de ellos. Ten en cuenta que ninguno de estos lenguajes está pensado específicamente para programadores principiantes, y la mayoría son significativamente diferentes de Python.

#### JavaScript

JavaScript ( [*https://dev.java/*](https://dev.java/) ) es un lenguaje de programación utilizado normalmente para páginas web, programación de juegos y otras actividades. Puedes crear fácilmente una simple página *HTML* (el lenguaje utilizado para crear páginas web) que contenga un programa JavaScript y ejecutarlo dentro de un navegador sin necesidad de un intérprete de comandos, una línea de comandos ni nada parecido.

Un ejemplo de "Hola Mundo" en JavaScript será diferente dependiendo de si lo ejecutas en un navegador o en un shell. En un shell, puedes escribir lo siguiente:

print('Hello World');

En un navegador, podría tener este aspecto:

<html>  
 <body>  
 <script type="text/javascript">  
 alert("Hello World");  
 </script>  
 </body>  
</html>

#### Java

Java ( [*http://www.oracle.com/technetwork/java/index.html*](http://www.oracle.com/technetwork/java/index.html) ) es un lenguaje de programación moderadamente complicado con una gran biblioteca incorporada de módulos (llamados *paquetes* ). Java es el lenguaje de programación utilizado en los dispositivos móviles Android, y puedes utilizarlo en la mayoría de los sistemas operativos.

Aquí tienes un ejemplo de impresión de "Hola Mundo" en Java:

public class HelloWorld {  
 public static final void main(String[] args) {  
 System.out.println("Hello World");  
 }  
}

#### C#

C# ( [*https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide*](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide) ), pronunciado "C sharp", es un lenguaje de programación moderadamente complicado para Windows que es muy similar a Java y JavaScript en cuanto a sintaxis. Forma parte de la plataforma .NET de Microsoft.

Aquí tienes un ejemplo de impresión de "Hola Mundo" en C#:

public class Hello  
{  
 public static void Main()  
 {  
 System.Console.WriteLine("Hello World");  
 }  
}

#### C/C++

C ( [*http://www.cprogramming.com*](http://www.cprogramming.com) ) y C++ ( [*http://www.stroustrup.com/C++.html*](http://www.stroustrup.com/C++.html) ) son complicados lenguajes de programación que se utilizan en todos los sistemas operativos. Encontrarás disponibles versiones gratuitas y comerciales. Ambos lenguajes (aunque quizás C++ más que C) tienen una curva de aprendizaje pronunciada (en otras palabras  , no son necesariamente buenos para principiantes). Por ejemplo, verás que tienes que codificar manualmente algunas funciones que ofrece Python (como decirle al ordenador que necesitas utilizar un trozo de memoria para almacenar un objeto). Muchos juegos comerciales y videoconsolas están programados en alguna forma de C o C++.

Un ejemplo de impresión de "Hola Mundo" en C es el siguiente:

#include <stdio.h>  
int main ()  
{  
 printf ("Hello World\n");  
}

Un ejemplo en C++ podría ser así:

#include <iostream>  
int main()  
{  
 std::cout << "Hello World\n";  
 return 0;  
}

#### Ruby

Ruby ( [*http://www.ruby-lang.org*](http://www.ruby-lang.org) ) es un lenguaje de programación libre disponible en los principales sistemas operativos. Se utiliza sobre todo para crear sitios web, concretamente utilizando el framework *Ruby on Rails .* (Un *framework* es un conjunto de bibliotecas que apoyan el desarrollo de tipos específicos de aplicaciones).

Aquí tienes un ejemplo de impresión de "Hola Mundo" en Ruby:

puts "Hello World"

#### Go

Go ( [*https://golang.org*](https://golang.org) ) es un lenguaje de programación similar a C, pero ligeramente más sencillo.

Puedes imprimir "Hola Mundo" con Go de esta forma:

package main  
import "fmt"  
func main() {  
 fmt.Println("Hello World")  
}

#### Rust

Rust ( [*https://www.rust-lang.org*](https://www.rust-lang.org) ) es un lenguaje desarrollado originalmente por Mozilla Research (los mismos que hacen el navegador Firefox).

Un programa sencillo para imprimir "Hola Mundo" con Rust podría tener este aspecto:

fn main() {  
 println!("Hello World")  
}

#### Swift

Swift ( [*https://swift.org*](https://swift.org) ) es un lenguaje desarrollado por Apple para sus dispositivos (iOS, macOS, etc.), por lo que es el más adecuado si utilizas un producto de Apple.

Podríamos imprimir "Hola Mundo" con Swift de esta forma:

import Swift  
print("Hello World")

### Palabras finales

Tanto si sigues con Python como si decides probar otro lenguaje de programación, los conceptos que has descubierto en este libro te seguirán resultando útiles. Aunque no continúes con la programación informática, comprender algunas de las ideas fundamentales puede ayudarte en todo tipo de actividades, ya sea en la escuela o, más adelante, en el trabajo.

¡Buena suerte y diviértete programando!

