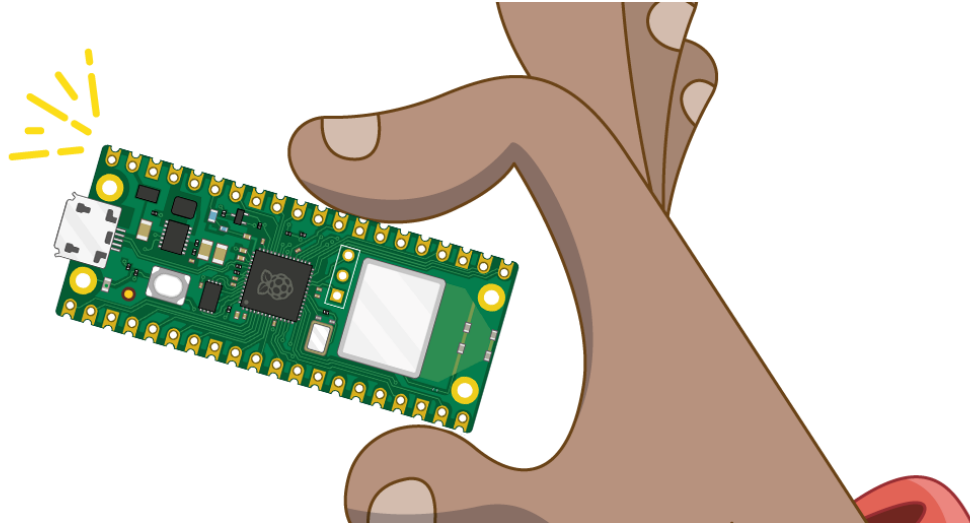


Primeros pasos con su Raspberry Pi Pico W

Frambuesa Pi Pico Pitón



Contents

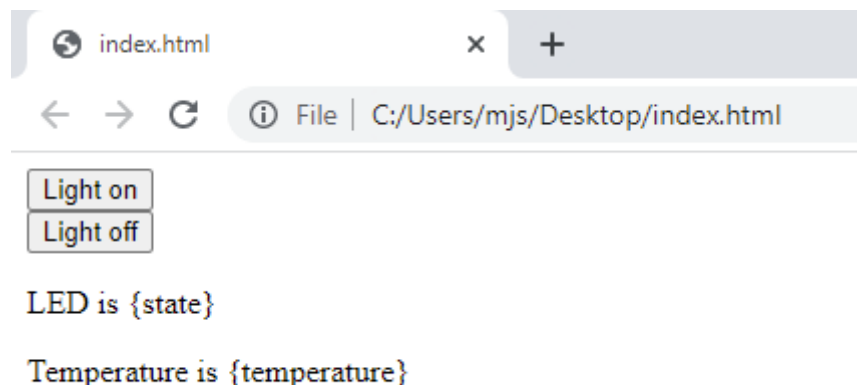
¡Guarda tu progreso!

Si desea volver a este proyecto más tarde, puede crear una cuenta de Raspberry Pi para guardar su progreso hasta el momento. En tu cuenta también verás todos los proyectos que completes.

Iniciar sesión o registrarse

Crear una página web

En este paso, creará una página web que el servidor web, que se ejecuta en su Raspberry Pi Pico W, puede enviar al navegador web de un cliente. Sin embargo, primero probará la página web en su computadora para asegurarse de que se muestre como debería. En el siguiente paso, puede agregar el código a su secuencia de comandos de Python, para que su



Raspberry Pi Pico W pueda servir la página web.

Una página web puede ser tan simple como un texto, formateado de tal manera que un navegador web lo represente y proporcione algo de interactividad. Aunque Thonny no está diseñado para escribir HTML, puede usarse para este propósito. Sin embargo, puede usar su editor de texto preferido si lo desea, ya sea VSCode, TextEdit o Notepad.



En su editor de texto o en Thonny, cree un nuevo archivo. Puede llamarlo como quiera, pero `index.html` es el nombre estándar para la primera página con la que interactúa un usuario. Asegúrese de agregar la `.html` extensión de archivo. Si usa Thonny, asegúrese de guardar en **Esta computadora**.



Hay un código HTML estándar que deberá incluir para empezar.

índice.html

```
<!DOCTYPE html>
<html>
<body>
</body>
</html>
```



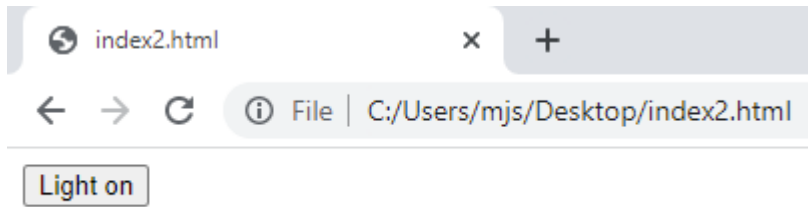
A continuación, puede crear un botón que se utilizará para encender o apagar el LED integrado.

índice.html

```
<!DOCTYPE html>
<html>
<body>
<form action="./lighton">
<input type="submit" value="Light on" ⤴
</form>
</body>
</html>
```



Guarde su archivo y luego búsquelo en su administrador de archivos. Cuando hace doble clic en el archivo, debería abrirse en su navegador web predeterminado. Así es como se ve la página web en Google Chrome.



Agregue un segundo botón para apagar el LED.

índice.html

```
<!DOCTYPE html>
<html>
<body>
<form action="./lighton">
<input type="submit" value="Light on" ↵
</form>
<form action="./lightoff">
<input type="submit" value="Light off" ↵
</form>
</body>
</html>
```

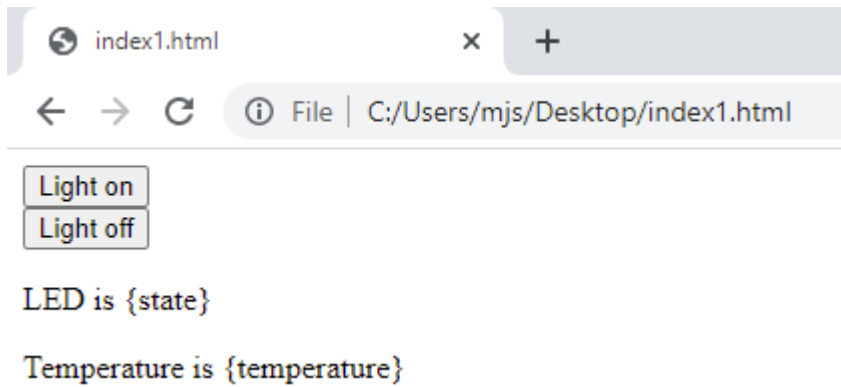


Para finalizar la página web, puedes añadir algunos datos extra, como el estado del LED y la temperatura de tu Raspberry Pi Pico W.

índice.html

```
<!DOCTYPE html>
<html>
<body>
<form action="./lighton">
<input type="submit" value="Light on" ↵
</form>
<form action="./lightoff">
<input type="submit" value="Light off" ↵
</form>
<p>LED is {state}</p>
<p>Temperature is {temperature}</p>
</body>
</html>
```

Su página web debería verse así:



Ahora que tiene una página web en funcionamiento, puede agregar este código a su secuencia de comandos de Python. Primero deberá volver a su código Python en Thonny.



Cree una nueva función llamada `webpage`, que tenga dos parámetros. Estos son `temperature` y `state`.

`servidor_web.py`

```
def webpage(temperature, state):
    #Template HTML
```



Ahora puede almacenar todo su código HTML que ha escrito y probado en una variable. El uso de **fstrings** para el texto significa que los marcadores de posición que tiene en el HTML para `temperature` y `state` se pueden insertar en su cadena.

`servidor_web.py`

```
def webpage(temperature, state):
    #Template HTML
    html = f"""
        <!DOCTYPE html>
        <html>
        <form action="./lighton">
        <input type="submit" value="Light on" ↗
        </form>
        <form action="./lightoff">
        <input type="submit" value="Light off" ↗
        </form>
        <p>LED is {state}</p>
        <p>Temperature is {temperature}</p>
        </body>
        </html>
        """
```



Por último, puede devolver la `html` cadena desde su función.

servidor_web.py

```
def webpage(temperature, state):
    #Template HTML
    html = f"""
        <!DOCTYPE html>
        <html>
        <form action="./lighton">
        <input type="submit" value="Light on" ↵
        </form>
        <form action="./lightoff">
        <input type="submit" value="Light off" ↵
        </form>
        <p>LED is {state}</p>
        <p>Temperature is {temperature}</p>
        </body>
        </html>
        """

    return str(html)
```

Guarda tu proyecto

No puede probar este código todavía, ya que su programa aún no está sirviendo el HTML. Eso se abordará en el siguiente paso.

El código HTML simple que acaba de escribir se almacenará en su secuencia de comandos de MicroPython y se enviará al navegador de cualquier computadora que se conecte a través de su red, al igual que una página web almacenada en cualquier otro servidor del mundo. Una diferencia importante es que solo los dispositivos conectados a su red WiFi pueden acceder a la página web o controlar su Raspberry Pi Pico W. Esta página es una demostración muy simple de lo que es posible. Para obtener más información sobre la codificación HTML y la creación de sitios web, ¡vea algunos de nuestros otros proyectos en este sitio!



Sirve tu página web

¿Detectó un error? ¿Disfrutando del proyecto? ¿Alguna opinión sobre la web? ¡Háznos saber!

Enviar comentarios

Publicado por [Raspberry Pi Foundation](#) bajo una [licencia Creative Commons](#) . [Ver proyecto y licencia en GitHub](#)

