

# Python

Módulo 06

# Enviar formulario HTML vía HTTP

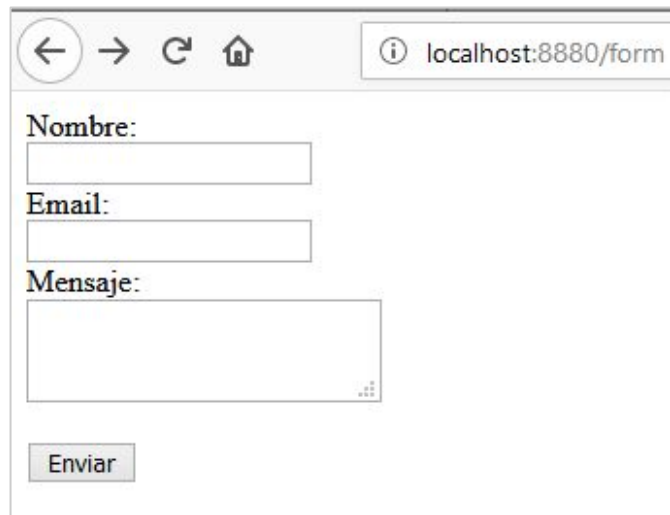
# Enviar formulario HTML vía HTTP

Los formularios son la principal forma de comunicación entre un usuario y un sitio web para enviar información. Por ejemplo, formularios de contacto o para iniciar sesión. Enviar información a través de un formulario usando Python es una operación muy común con el objetivo de automatizar tareas. La misma librería con la que hemos estado trabajando hasta ahora, ***Requests***, nos permite efectuar dicha operación. De hecho, el método HTTP que usan los formularios HTML para enviar información es ***POST***.

Lo primero que necesitamos es un sitio con un formulario. Nuevamente vamos a usar una aplicación de ejemplo que crea justamente eso. Podés descargarla desde la sección de descargas de esta clase. Luego, ejecutarla con el siguiente comando:

```
python webform.py
```

Este pequeño sitio de ejemplo expone el siguiente formulario de contacto en <http://localhost:8880/form>.



A screenshot of a web browser window. The address bar shows 'localhost:8880/form'. The page content includes a contact form with three input fields: 'Nombre:', 'Email:', and 'Mensaje:'. Below the fields is a button labeled 'Enviar'.

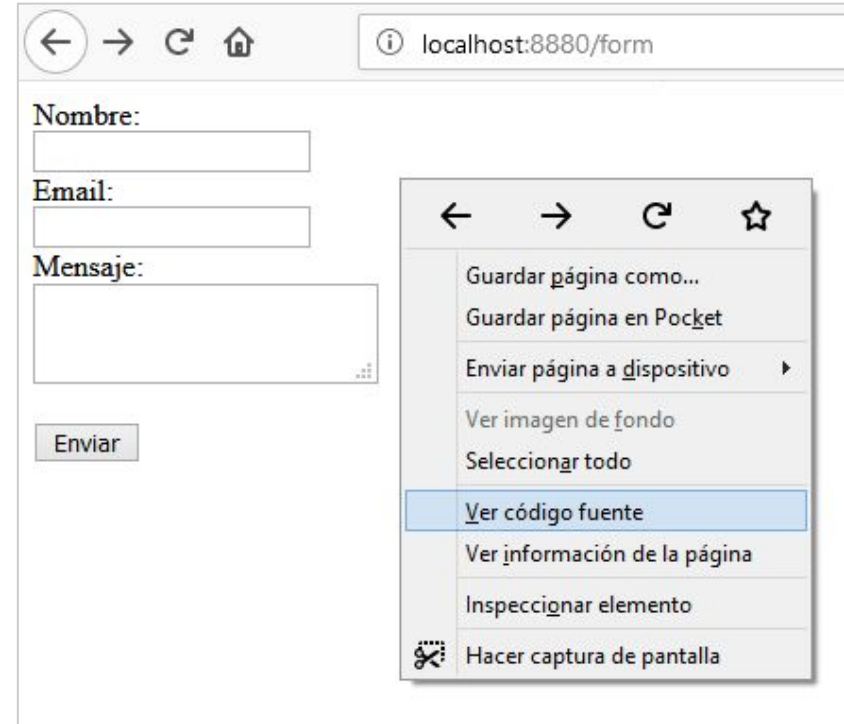
Para automatizar el envío, lo primero que tenemos que identificar es la dirección de URL. Luego, acabamos de decir que la información de un formulario HTML en un sitio web se envía a través del protocolo HTTP usando el método **POST**; entonces, nuestro código de Python comenzará siendo el siguiente:

```
import requests  
  
r = requests.post("http://localhost:8880/form")
```

El siguiente paso es pasar como argumento a **post()** los datos que queremos enviar: el nombre, el correo electrónico y el mensaje mismo. Estos datos tienen que estar contenidos como valores de un diccionario, ¿pero cuáles serán las claves?

Para ello necesitamos ver el código de fuente del formulario. Todos los navegadores contienen una opción para ello en el menú contextual (*ver imagen*). Se abrirá una nueva ventana y veremos el siguiente código:

```
<!DOCTYPE html>
<html>
<head><meta charset="utf-8"></head>
<body>
  <form method="post">
    Nombre:<br>
    <input type="text" name="name">
    <br>
    Email:<br>
    <input type="text" name="email">
    <br>
    Mensaje:<br>
    <textarea name="message"></textarea>
    <br><br>
    <input type="submit" value="Enviar">
  </form>
</body>
</html>
```



El fragmento de código que configura el formulario HTML está entre las etiquetas **<form>** y **</form>**. Lo que nos interesa dentro de él son los nombres que el sitio le ha asignado a los componentes del formulario (esto es, las cajas de texto para ingresar datos) vía el atributo *name*. Los marcamos en rojo a continuación.

Vemos que los nombres para los campos de nombre, correo electrónico y mensaje son, respectivamente, *name*, *email* y *message*.

```
<!DOCTYPE html>
<html>
<head><meta charset="utf-8"></head>
<body>
  <form method="post">
    Nombre:<br>
    <input type="text" name="name">
    <br>
    Email:<br>
    <input type="text" name="email">
    <br>
    Mensaje:<br>
    <textarea name="message"></textarea>
    <br><br>
    <input type="submit" value="Enviar">
  </form>
</body>
</html>
```

Pues bien, éstos corresponderán, entonces, a las clave de nuestro diccionario de datos:

```
datos = {  
    "name": "Mariano",  
    "email": "mariano@ejemplo.com",  
    "message": "¡Hola, mundo!"  
}  
r = requests.post("http://localhost:8880/form",  
data=datos)
```

(Aquí el argumento para **post()** es *data* y no *json*, a diferencia del apartado anterior, ya que al enviar datos en un formulario raramente se codifican en formato JSON).

¡Perfecto! Ahora, ¿cómo sabemos que el formulario se ha enviado correctamente?

Primero intentemos enviarlo manualmente usando el navegador. Luego de completar los datos veremos que la página responde con lo siguiente:



Podemos acceder a la respuesta del sitio desde Python vía ***r.content***:

```
datos = {  
    "name": "Mariano",  
    "email": "mariano@ejemplo.com",  
    "message": "¡Hola, mundo!"  
}  
r = requests.post("http://localhost:8880/form", data=datos)  
contenido = r.content.decode("utf-8")  
print(contenido)
```

**Nota:** ***r.content*** corresponde a un tipo de dato que no hemos visto, *bytes*, pero que se puede convertir a una cadena fácilmente usando la función ***decode()***. El argumento indica la codificación de caracteres de la página del sitio que estamos visitando, que por lo general es UTF-8.

Ésta se encuentra declarada dentro de la etiqueta ***<head>*** en el código HTML:

```
<!DOCTYPE html>  
<html>  
<head><meta charset="utf-8"></head>  
<body>
```



Esto imprime:

```
<!DOCTYPE html>
<html>
<head><meta charset="utf-8"></head>
<body>¡Mensaje enviado correctamente!</body>
</html>
```

Eso quiere decir que podemos hacer una comprobación desde nuestro código para determinar si el formulario se envió correctamente, igualando contenido a la cadena anterior. Pero dado que el resultado en HTML suele ser muy voluminoso, simplemente podemos buscar alguna parte de la cadena, por ejemplo, *“Mensaje enviado correctamente”*.

Para ello usamos el operador ***in***, como vimos en las colecciones:

```
datos = {
    "name": "Mariano",
    "email": "mariano@ejemplo.com",
    "message": "¡Hola, mundo!"
}
r = requests.post("http://localhost:8880/form",
    data=datos)
contenido = r.content.decode("utf-8")
if "Mensaje enviado correctamente" in contenido:
    print("¡Formulario enviado!")
else:
    print("Ocurrió un error.")
```

O bien usando la función ***find()*** según lo visto en la sección de operaciones sobre cadenas:

```
if contenido.find("Mensaje enviado correctamente") > -1:  
    print("¡Formulario enviado!")  
else:  
    print("Ocurrió un error.")
```



# ¡Muchas gracias!

¡Sigamos trabajando!