

Taller de FutureMakers

Día 2

Inteligencia artificial (IA) y aprendizaje automático (Machine Learning) (ML)

Versión 1.0.1

Esta guía te ayudará a completar el taller de FutureMakers.
Avanza a tu propio ritmo.

Tu nombre:

No lo olvides: la programación puede ser difícil, pero es divertido - ¡puedes hacerlo!

No tengas miedo de cometer errores - los errores son la forma en que aprendemos.

No tengas miedo de hacer preguntas - los formadores están para ayudarte.

Introducción

Bienvenido al taller de FutureMakers. Vamos a repasar una serie de actividades diseñadas para introducirte en la programación, el aprendizaje automático y la inteligencia artificial. No te preocupes si no tienes mucha experiencia en codificación, o incluso si nunca has programado antes. Este taller te mostrará cómo codificar en Python (Parte A), cómo usar herramientas emocionantes como detectores de rostros o traducción automática de idiomas (Parte B), y cómo construir tu propio sitio web interactivo (Parte C).

A lo largo de este taller, recuerda que la programación puede ser difícil, pero también es muy divertida. No tengas miedo de cometer errores y hacer preguntas - cometer errores es la forma en que aprendemos, e incluso los programadores más profesionales buscan frecuentemente en la Web cosas que no entienden. La programación no consiste en conocer todas las respuestas, sino en saber cómo buscarlas, a menudo haciendo una búsqueda en la Web o preguntando a un amigo o colega.

Parte C

Creación de una verdadera aplicación web interactiva

En esta parte del taller, construiremos un sitio web interactivo y lo pondremos en la Web para que usted - y cualquier otra persona - pueda acceder a él. Usaremos tres herramientas para hacer esto:

- **La línea de comandos:** una forma anticuada, pero muy útil, de pedirle a su ordenador que haga cosas
- **Heroku:** una empresa que se encarga de ejecutar una aplicación web para ti. Todo lo que tienes que hacer es enviar a Heroku el código de tu aplicación web, y ésta estará disponible en línea a través de un servidor web.
- **Git:** un útil programa que se encarga de enviar tu código a otros ordenadores. Git es utilizado a menudo por los programadores para trabajar juntos en el software. Lo usaremos para enviar nuestro código a Heroku.

¿Qué significa esto? Un servidor web puede significar un ordenador cuyo trabajo es ejecutar sitios web. Cada sitio web necesita un servidor web; este es el ordenador con el que su navegador habla cuando usted abre un sitio web. Los grandes sitios web, como Youtube, necesitan cientos de miles de servidores para poder tratar con todos sus visitantes. Si usted no tiene suficientes servidores para ejecutar el sitio web para todos sus visitantes, su sitio web será lento, o incluso se romperá.

Ejercicio: acostumbrarse a la línea de comandos

La línea de comandos es la forma en que solíamos hablar con los ordenadores antes de windows, los archivos y las carpetas. Funciona sólo con texto. Para decirle a la computadora que haga algo, escriba un comando en la línea de comandos y presiona return. La computadora escribirá algunas líneas en la ventana de comandos para que

Recuerda: ¡ aprendemos cometiendo errores, y es bueno hacer preguntas!

puedas ver si tu comando funcionó.

Para abrir la línea de comandos, selecciona Anaconda Prompt en el menú Inicio:

Recuerda: ¡ aprendemos cometiendo errores, y es bueno hacer preguntas!

Ahora probemos algunos comandos. Si utilizas un ordenador portátil con Windows, utiliza los comandos de la columna Windows; si utilizas un ordenador Mac o Linux, utiliza los comandos de la columna OS X/Linux/Unix.

Esta es una lista de algunos comandos comunes. No escribas esto todavía.

Tarea	Comando	Ejemplo de Windows	OS X/Linux modelo
Mostrar la carpeta actual (al igual que en Windows, una ventana de línea de comandos siempre está <i>en</i> una carpeta en particular)	<code>pwd</code> o <code>cd</code>	<code>pwd</code>	<code>cd</code>
Enumere los archivos en el carpeta actual	<code>dir</code> o <code>ls</code>	<code>dir</code>	<code>ls</code>
Cambiar la corriente carpeta	<code>cd</code>	<code>cd</code> <code>/Usuarios/alice</code>	<code>cd /Usuarios/bob</code>
Crear una nueva carpeta	<code>mkdir</code>	<code>mkdir</code> <code>mi_nueva_carpeta</code>	<code>mkdir</code> <code>mi_nueva_carpeta</code>

Probemos algunos comandos.

Primero, averigüa la ruta (dirección en el sistema de archivos) de su carpeta FutureMakers. Por ejemplo, podría ser `/Users/alice/Desktop/FutureMakers` en un Mac, o `C:\bob\FutureMakers` en un PC con Windows. Pregunte a un instructor si necesita ayuda para encontrar el camino.

Escribe tu ruta de FutureMakers en algún lugar - necesitarás copiarla donde veas las rutas de ejemplo (`/Users/alice/Desktop/FutureMakers` en un Mac, o `C:\bob\FutureMakers`) **arriba**.

Si estás utilizando un ordenador con Windows:

- 1. Abre una ventana de Anaconda Prompt (Preguntar a la Anaconda).
- Usa `cd` para averiguar en qué carpeta se encuentra actualmente.
- Utiliza `dir` para mostrar los archivos de la carpeta actual.
- Utiliza `cd /Usuarios/alice/Desktop/FutureMakers` para moverse a la carpeta FutureMakers.
- Utiliza `dir` para mostrar los archivos de la carpeta FutureMakers.
- Utilice `cd /Usuarios/alice/Desktop/FutureMakers/sample-web-app` para moverse a la carpeta sample-web-app en la carpeta FutureMakers.
- Use `dir` para mostrar los archivos en la carpeta sample-web-app

Recuerda: ¡ aprendemos cometiendo errores, y es bueno hacer preguntas!

Si utilizas un ordenador Mac o Linux:

- 1. Abra una ventana de Anaconda Prompt (Preguntar a la Anaconda).
- Utilice `pwd` para saber en qué carpeta se encuentra actualmente.
- Utilice `ls` para mostrar los archivos de la carpeta actual.
- Utilice `cd /Usuarios/alice/Desktop/FutureMakers` para moverse a la carpeta FutureMakers.
- Utilice `ls` para mostrar los archivos de la carpeta FutureMakers.
- Utilice `cd /Usuarios/alice/Desktop/FutureMakers/sample-web-app` para moverse a la carpeta sample-web-app en la carpeta FutureMakers.
- Use `ls` para mostrar los archivos en la carpeta sample-web-app

Ejercicio: guardar un archivo en Texto Sublime



El icono del Texto Sublime

No usaremos el Cuaderno Jupyter para esta parte del taller: editaremos nuestros archivos en Texto Sublime, que es un programa rápido y útil para editar archivos de texto.

Consejo: no se puede programar en Word, ni en ningún editor que soporte *texto enriquecido* (fuentes, colores, cursivas, negrita, etc.). Para la programación, siempre utilizamos un editor de *texto plano* (sin fuentes, no se pueden cambiar los colores, sin *cursiva* ni **negrita**). Hay muchos editores de texto plano disponibles; Sublime Text es uno de los mejores.

Para empezar:

- Abrir Texto Sublime desde el menú Inicio
- Selecciona Proyecto -> Añadir carpeta al espacio de trabajo.
Ahora podrá ver todo en la carpeta FutureMakers en la barra de carpetas a la izquierda de Sublime Text.
- Haz un nuevo archivo y guárdelo como test.txt (la extensión.txt significa que este archivo sólo contiene texto, no código de programa).

Ejercicio: ejecutar un archivo Python desde la línea de comandos

Hasta ahora, hemos estado ejecutando código Python ya sea en CodeSkulptor o usando el Jupyter Notebook. Estos dos son métodos bastante modernos para ejecutar código. Durante los últimos 50 años, los programadores han utilizado principalmente la línea de comandos, y eso es lo que vamos a utilizar para ejecutar nuestro servidor web.

Recuerda: ¡ aprendemos cometiendo errores, y es bueno hacer preguntas!

Primero vamos a aprender a ejecutar un simple archivo Python desde la línea de comandos.

Recuerda: ¡aprendemos cometiendo errores, y es bueno hacer preguntas!

Si estás usando un ordenador con Windows:

- Ejecuta el comando `cd` para comprobar la carpeta en la que se encuentra.
- Ejecuta el `cd C:\FutureMakerspython-notebooks` para pasar a la carpeta python dentro de la carpeta FutureMakers.
- Ejecuta el comando `cd` para comprobar que estás en la carpeta python.
- Ejecuta el comando `dir` para listar los archivos en la carpeta python.
- Ejecuta el comando `python hello.py` para ejecutar el archivo `hello.py`.
Esto es exactamente lo mismo que ejecutar una celda en el Jupyter Notebook, o presionar el botón de ejecución en CodeSkulptor: le pide a la computadora que ejecute el código.
- Deberías ver un mensaje de "¡Hola, FutureMakers!"

Si utilizas un ordenador Mac o Linux:

- Ejecuta el comando `cd` para comprobar la carpeta en la que se encuentra.
- Ejecuta `/Users/alice/Desktop/FutureMakers/python-notebooks` para moverse a la carpeta python dentro de la carpeta FutureMakers.
- Ejecuta el comando `cd` para comprobar que estás en la carpeta python.
- Ejecuta el comando `ls` para listar los archivos en la carpeta python.
- Ejecuta el comando `python hello.py` para ejecutar el archivo `hello.py`.
Esto es exactamente lo mismo que ejecutar una celda en el Jupyter Notebook, o presionar el botón de ejecución en CodeSkulptor: le pide a la computadora que ejecute el código.
- Deberías ver un mensaje de "¡Hola, FutureMakers!"

Felicidades, acabas de ejecutar un archivo Python usando la línea de comandos.

Ejercicio: editar código en Sublime Text y ejecutarlo desde la línea de comandos

En el resto del taller, editaremos código en texto sublime y lo ejecutaremos en tu portátil usando la línea de comandos, o lo enviaremos a Heroku para que tu sitio web sea accesible al mundo.

¿Qué significa esto? Cuando **se despliega** el código, se envía a otra parte para que pueda ejecutarse. Cuando actualizamos un sitio web o enviamos un nuevo juego de ordenador, estamos desplegando nuestro código para que sea accesible a los usuarios.

Vamos a intentar editar algún código en Sublime Text y ejecutarlo.

- En Sublime, utilice la barra de la izquierda para encontrar la carpeta python dentro de la carpeta FutureMakers, y para encontrar el archivo `hello.py` dentro de ella.
- Use Archivo > Guardar como para hacer una copia de `hello.py`. Dale un nuevo nombre, pero asegúrate de que termina en `.py` (por ejemplo, podrías llamarlo `goodbye.py`).
- Cambiar el nuevo archivo para que diga algo diferente al usuario. Asegúrese

Recuerda: ¡ aprendemos cometiendo errores, y es bueno hacer preguntas!

de guardarlo después.

- Vuelve a la línea de comandos. Use `cd` para asegurarse de que todavía está en la carpeta `python` dentro de la carpeta `futuremakers`.

Recuerda: ¡aprendemos cometiendo errores, y es bueno hacer preguntas!

- Ejecuta tu nuevo archivo utilizando el comando python (por ejemplo, python goodbye.py) y compruebe que funciona correctamente.

Enhorabuena, acabas de crear un nuevo archivo de código Python y lo has ejecutado usando la línea de comandos.

¿Qué significa esto?

- Un servidor web es un programa que se ejecuta constantemente para entregar sitios web a los usuarios.
- Un navegador web se comunica con el servidor haciendo una **petición**. (También se conocen como solicitudes de protocolo de transferencia de hipertexto (HTTP)).
- Los servidores también pueden comunicarse entre sí haciendo peticiones.
- Hay dos tipos de petición, la petición GET (usada para pedir una página web) y la petición POST (usada para enviar algunos datos al servidor y luego recuperar una página web).
- Flask es un simple programa de servidor web que está escrito en Python. Lo usaremos para dirigir nuestro sitio web.

Consejo: podemos ejecutar nuestro servidor web en nuestro portátil (en cuyo caso podrás acceder al sitio web en tu portátil, pero no será accesible desde ningún otro lugar) o en Heroku (en cuyo caso será visible para todo el mundo). Sería posible configurar su computadora portátil para que sirva el sitio web al mundo, pero esto sería una mala idea, ya que el sitio web dejaría de estar disponible si su computadora portátil estuviera apagada.

Ejercicio: ejecución del proyecto de servidor web de ejemplo Flask

Echemos un vistazo a la aplicación de ejemplo Flask. Esta aplicación es un buen punto de partida para el resto del proyecto.

- En Texto Sublime, debe tener la carpeta FutureMakers mostrada en la barra de la izquierda. Si no está ahí, añádalo con Proyecto > Añadir carpeta al proyecto.
- En la barra izquierda de Sublime Text, busque dentro de la carpeta FutureMakers y encuentre la carpeta my_web_app.
- Echa un vistazo a los archivos Python dentro de esta carpeta. Lee los comentarios - te dicen lo que hace el código.

Ahora vamos a lanzar la aplicación.

- Abre una nueva ventana de línea de comandos seleccionando Anaconda Prompt en el menú Inicio.
- En la línea de comandos, vayamos a la carpeta sample_web_app: use el comando `cd /Users/alice/Desktop/FutureMakers/sample-web-app`.
- Comprueba que estás en la carpeta correcta: usa el comando `cd`
- Para asegurarte, lista los archivos en esta carpeta con el comando `ls`. Deberías ver los mismos archivos que vio en la carpeta my_web_app en Sublime Text.

Recuerda: ¡ aprendemos cometiendo errores, y es bueno hacer preguntas!

- Necesitamos instalar algunas librerías antes de iniciar la aplicación. Ejecute el comando `pip install -r requirements.txt`

Recuerda: ¡ aprendemos cometiendo errores, y es bueno hacer preguntas!

Deberías ver alguna salida que le diga que se han instalado las bibliotecas. Si ve un mensaje de advertencia o error, pregunte a un instructor

- Una vez que estés en la carpeta correcta, usa el siguiente comando para ejecutar la aplicación: en Mac, use `./run`

en Windows, use `run.bat`

(No usea `python server.py` ya que esto no actualizará tu código cuando lo cambies y tendrás que reiniciar el servidor para ver los cambios).

Consejo: el comando comienza con un punto y luego una barra oblicua.

Deberías ver un mensaje como el siguiente:

```
* Sirviendo la aplicación de frasco "server. py" (carga perezosa)
* Medio ambiente: desarrollo
* Modo de depuración: activado
* Se ejecuta en http://127.0.0.1:5000/ (Presione CTRL+C para salir)
* Reinicio con stat
* El depurador está activo!
* PIN del depurador: 277-273-612
```

Si ve este mensaje, significa que el comando ha funcionado y que el servidor web Flask se está ejecutando en su portátil, sirviendo a la aplicación web. No es accesible desde todo el Internet, pero puedes visitarlo en tu laptop!

Consejo: la ventana de la línea de comandos ya no escucha sus comandos (no puede escribir nada en ella), porque está ocupada ejecutando el servidor web de Flask. Si quieres apagarlo, presiona `ctrl` y `C` a la vez. Para empezar de nuevo, puede utilizar el comando `./run`.

¿Cómo visitamos la aplicación web? Cuando iniciamos Flask, nos dijo la dirección en la que estaba funcionando:

`http://127.0.0.1:5000/`

Consejo: se trata de una dirección web o URL normal. `127.0.0.1` es la dirección IP que su ordenador portátil se da a sí mismo; `5000` es el puerto en el que estamos accediendo a la aplicación.

¿Qué significa? Una URL o un localizador uniforme de recursos es sólo una dirección Web. Las URL pueden ser palabras (como `google.com`) o números (como `127.0.0.1`).

Para visitar la aplicación, abra un navegador Web (Firefox o Chrome), ingrese a `http://127.0.0.1:5000/` en la barra de direcciones y presione `return`. Si todo funciona correctamente, deberías ver la aplicación de ejemplo y el mensaje "¡Bienvenido a la aplicación de ejemplo de FutureMakers! Si tienes algún problema o pregunta, pregunta a una demostradora.

Recuerda: ¡ aprendemos cometiendo errores, y es bueno hacer preguntas!

Consejo: En lugar de introducir la URL completa, puedes introducir localhost:5000 en la barra de direcciones de su navegador. localhost es la palabra equivalente a 127.0.0.1.

Recuerda: ¡aprendemos cometiendo errores, y es bueno hacer preguntas!

Ejercicio: empezar a modificar la aplicación de ejemplo

Cambiamos el mensaje de bienvenida en la aplicación de ejemplo.

En Sublime, localice la carpeta FutureMakers en la barra izquierda, luego la carpeta python, luego la carpeta my_web_app, luego la carpeta templates y luego el archivo index.html.

En otras palabras, utilice Sublime para abrir el archivo
FutureMakers/python/my_web_app/templates/index.html

¿Qué significa esto? Una **plantilla** es un archivo que ayuda al servidor Flask a construir una página web. Flask reúne diferentes plantillas en la página web final, y también inserta datos personalizados como el nombre del usuario, un mensaje de bienvenida personalizado o la hora actual.

Las plantillas, como la mayoría de las páginas, están escritas en HTML (hypertext markup language), un lenguaje especial para definir páginas web. El *hipertexto* es un texto sobre el que se puede hacer clic (es decir, un enlace). Un lenguaje de marcado es un lenguaje destinado a describir documentos y no a programar.

HTML utiliza palabras especiales llamadas **etiquetas** para describir diferentes partes de las páginas web. Por ejemplo, algún texto en negrita está rodeado por la etiqueta ``:

<code>Este texto es en negrita </code>	Este texto está en negrita
Esta es la primera línea. <code>
</code> Esta es la segunda línea	Esta es la primera línea. Esta es la segunda línea.
Este texto es normal, pero <code><i></code> esto es cursiva <code></i></code> y <code></code> esto es negrita <code></code> .	Este texto es normal, pero <i>está en cursiva</i> y en cursiva esto es audaz
Este es un texto normal <code><h1></code> y este es un encabezado <code></h1></code>	Este es un texto normal y este es un encabezado

La Web está llena de recursos sobre HTML. Busca en Google cómo hacer lo que quieras y encontrarás muchos ejemplos. También hay una hoja de trucos disponible con los materiales de FutureMakers.

Consejo: la primera y más importante página de un sitio web se llama tradicionalmente index.html.

Veamos index.html, que ya deberías tener abierto en Sublime Text, y encontremos la parte de la página que dice "Welcome to the FutureMakers sample app".

Aquí está; podemos ver que está rodeado de etiquetas `<h1>`, lo que significa que se mostrará como un encabezado:

Recuerda: ¡aprendemos cometiendo errores, y es bueno hacer preguntas!

<h1>Bienvenido a la aplicación de ejemplo de FutureMakers!

- Cambia esto por algo más interesante.
- Ahora guarde el archivo (control - S, o Archivo > Guardar).

Recuerda: ¡aprendemos cometiendo errores, y es bueno hacer preguntas!

- Ahora vuelve a tu navegador y actualiza la página web.

Una vez que se actualice, deberías ver que la página web ha cambiado para mostrar su mensaje.

Sugerencia: cada vez que hacemos un cambio en un sitio web en el que estamos trabajando, siempre debemos guardar nuestro archivo de código en el editor de texto, luego pasar al navegador y hacer una actualización. Los programadores web hacen esto muy a menudo.

Ejercicio: implementación de la aplicación de ejemplo en Heroku

Por el momento, tu aplicación web sólo se puede usar en tu portátil: no se puede acceder a ella desde ningún otro sitio. Cambiemos eso desplegándolo en Heroku para que sea visible para todo el mundo.

Para ello utilizaremos dos herramientas: el Heroku toolbelt, que es un programa instalado en tu portátil para comunicarte con Heroku, y Git, que es un programa de propósito general para rastrear, cuidar, enviar y recibir archivos de código como los que componen tu aplicación web.

Lo primero que tenemos que hacer es tenderle una trampa a Heroku. Se te ha dado una cuenta de Heroku con su propio nombre de usuario y contraseña. Abra una nueva ventana de línea de comandos (Anaconda Prompt en el menú Inicio), ejecute el comando **heroku login**, e introduzca su nombre de usuario y contraseña. Heroku te dirá si has iniciado sesión correctamente.

Lo siguiente, tenemos que preparar a Git. Para hacer esto,

- Abra una nueva ventana de línea de comandos (Anaconda Prompt en el menú Inicio) y navegue a la carpeta FutureMakers/python/sample-web-app usando `cd /Usuarios/alice/Desktop/FutureMakers/sample-web-app` en un Mac, o `cd C:\FutureMakers - Ejemplo de aplicación web` en Windows.
- Compruebe que está en la carpeta correcta ejecutando **cd**
- Ejecuta **git init** para pedirle a Git que empiece a rastrear la carpeta sample-web-app
- Ejecute **heroku create** para crear una nueva aplicación Heroku - este será el hogar en línea de nuestra aplicación.

Estamos casi listos para desplegar nuestra aplicación web. Primero abramos los registros de Heroku para que podamos ver, en directo, lo que sucede cuando nos desplegamos - ¡y si algo sale mal!

- Mantén abierta la ventana de la línea de comandos existente a la izquierda de la pantalla.
- Abre otra ventana de línea de comandos a la derecha de la pantalla y vaya a la carpeta FutureMakers/python/sample-web-app usando `cd /Usuarios/alice/Desktop/FutureMakers/sample-web-app` en un Mac, o `cd`

Recuerda: ¡aprendemos cometiendo errores, y es bueno hacer preguntas!

C:\`FutureMakers\`sample-web-app on Windows...

- Ejecuta `cd` para comprobar que está en la carpeta correcta.
- Ejecuta el comando **heroku logs -t**
Esto le pide a Heroku que nos diga exactamente lo que está haciendo enviándonos sus archivos de registro (que mantienen un diario de todo lo que le sucede a nuestra aplicación Heroku).

Recuerda: ¡aprendemos cometiendo errores, y es bueno hacer preguntas!

Ahora deberías tener dos ventanas de línea de comandos: una a la izquierda, que está lista para aceptar comandos, y otra a la derecha, que te muestra los registros de Heroku. Asegúrese de que puede ver todas las dos ventanas (es posible que tenga que hacerlas más pequeñas o moverlas).

Ahora estamos listos para desplegarlos! Simplemente ejecute el comando **./deploy** (no olvide el punto y la barra oblicua). La implementación debería tomar alrededor de un minuto, tal vez dos minutos: puede ver los registros para ver el progreso.

Solución de problemas de Git

Si ves un mensaje de Git diciendo que tu email o nombre no ha sido configurado, intenta decirle a Git tu nombre de esta manera (no uses tus datos reales):

```
git config --global user.email "example@example.com"
git config --global user.name "Future Makers"
```

Utilízalo sólo si Git te da un mensaje de error; normalmente, sólo muestra un mensaje diciendo que ha configurado automáticamente tu nombre y tu correo electrónico.

Comprobación de si el despliegue ha funcionado

Si el despliegue funcionó correctamente, verá un mensaje (en la ventana de la izquierda) como

<https://polar-wildwood-98641.herokuapp.com/> desplegado en Heroku

Este mensaje te indica la dirección web de su aplicación. En este caso, deberás copiar y pegar <https://polar-wildwood-98641.herokuapp.com/> en la barra de direcciones de tu navegador favorito.

Visite la dirección que aparece en el mensaje de implementación y comprueba que su aplicación web funciona correctamente. Pregúntale a una demostradora si hay algún problema.

También puedes ejecutar el comando `heroku open` para visitar la aplicación directamente. Enhorabuena, acabas de desplegar una aplicación web en Heroku!

Consejo: tu aplicación Heroku, por ejemplo <https://polar-wildwood-98641.herokuapp.com/>, es ahora accesible para todo el mundo. Si quieres una URL más conveniente, puedes comprar un nombre de dominio como www.my-new-app.com, y configurarlo para que apunte a tu URL de Heroku. Tu aplicación permanecerá disponible indefinidamente, aunque bajo el plan gratuito de Heroku, puedes tomar unos segundos para que tu aplicación responda si nadie la ha usado por un tiempo.

Recuerda: ¡aprendemos cometiendo errores, y es bueno hacer preguntas!

Ejercicio: añadir una nueva página a la aplicación Flask

Añadamos otra página a la aplicación Flask. En Flask, una página se compone de dos partes:

1. La **plantilla HTML**, que controla el aspecto de la página. Por ejemplo, una plantilla HTML muy simple podría tener el siguiente aspecto

Recuerda: ¡aprendemos cometiendo errores, y es bueno hacer preguntas!

```
<h1>Este es mi título! </h1>
```

```
<Y este es mi primer y único párrafo.
```

Puedes ver esto en `sample-web-app/templates/test_template.html`

2. La **ruta**, que es un código Python que le dice a Flask qué dirección debe tener tu página y qué hacer cuando alguien solicita esa página, incluyendo qué plantilla mostrar. Por ejemplo, esta ruta mostrará la plantilla `test_template.html` cuando el usuario visite `localhost:5000/test`

```
@app.route('/test')
def test_function():
    devolver render_template('test_template.html')
```

Esto sería el equivalente, si nuestra aplicación se ejecutara en `www.myapp.com`, de visitar `www.myapp.com/test`

Una ruta es especial: es la que obtenemos cuando no solicitamos una página en particular (visitando `www.myapp.com` en la Web, o `localhost:5000` en tu laptop). Esta ruta comienza con `@app.route('/')`, así:

```
@app.route('/')
def hello_world():
    devolver render_template('index.html')
```

Mira estas dos rutas (`test_function` y `hello_world`) en `sample-web-app/server.py`, y asegúrate de entenderlas. Pregúntale a un de los profesores si tienes alguna pregunta.

Ahora vamos a añadir una nueva página. Lo llamaremos adiós.

Hagamos la plantilla primero. Crear un nuevo archivo llamado `sample-web-app/templates/goodbye.html`, y guarda lo siguiente dentro de él:

```
<h1>Adiós! </h1>
```

```
<Esperamos que nos visites de nuevo
```

```
pronto.</p> Ahora hagamos la ruta. Debería
```

```
verse así: @app.route('/adiós') def goodbye():
    devolver render_template('goodbye.html')
```

En otras palabras, se trata de una función llamada `goodbye`, que escucha a un usuario escribiendo una URL que termina con `goodbye`, y que luego muestra una plantilla llamada `goodbye.html`. No todos estos nombres tienen que ser iguales - los hemos elegido para que sean iguales aquí por simplicidad.

Recuerda: ¡aprendemos cometiendo errores, y es bueno hacer preguntas!

Ahora vamos a comprobar que todo funciona - visita localhost:5000/goodbye. Deberías ver tu nueva página web. Si algo sale mal, pregúntale a un profesor/a.

Enhorabuena, acabas de crear una nueva página web!

Tu turno: añade una nueva página propia

Sigue los pasos del último ejercicio para crear su propia plantilla, luego crea tu propia función de ruta, luego visítala en el navegador y pruébala.

Sugerencia: asegúrate de que ni su función, ni su ruta, ni su plantilla tengan el mismo nombre que las anteriores.

Ejercicio: implementa tu aplicación de nuevo Ejercicio:

añade algunos estilos visuales

Utiliza la siguiente información sobre HTML (lenguaje de marcado de hipertexto) y CSS (conexión en cascada)) para añadir una nueva página a su aplicación. Asegúrate de guardar y revisar tu página con frecuencia para que puedas ver los efectos de los cambios que has hecho. Puedes hacerlo guardando tus archivos en Texto Sublime, y luego actualizando el navegador. No hagas muchos cambios a la vez y espera para revisar su página.

CSS (Hojas de Estilo en Cascada)

Las hojas de estilo se utilizan para aplicar estilos visuales consistentes (fuentes, colores y espacios) a todas las páginas.

En este punto puedes notar que estás haciendo cambios en el CSS y que no se reflejan en el navegador. Esto puede deberse a que tu navegador no está actualizando la última versión del archivo CSS; está usando una versión en caché que está almacenando él mismo (esto hace que las actualizaciones sean mucho más rápidas, pero causa problemas si el CSS acaba de cambiar).

Puedes probar con otro navegador (Firefox o Chrome) o, para asegurarte de que todo está actualizado, haz un "hard refresh":

- En la mayoría de los navegadores Windows y Linux: Mantenga pulsada la tecla Ctrl y pulse F5.
- En Apple Safari: Mantenga pulsado ⌘ Mayúsculas y haga clic en el botón Recargar barra de herramientas.
- En Chrome y Firefox para Mac: Mantenga presionada la tecla ⌘ Cmd+ ⌘ Mayúsculas y presione R.

Tu turno: añade tú mismo algunos estilos visuales

Recuerda: ¡aprendemos cometiendo errores, y es bueno hacer preguntas!

La Web está llena de recursos sobre HTML y CSS. Busca en Google cómo hacer lo que quieras y encontrarás muchos ejemplos. También hay dos hojas de trucos disponibles con los materiales de FutureMakers. Esta tabla muestra algunas reglas comunes de CSS.

Efecto	código VCC
--------	------------

Recuerda: ¡aprendemos cometiendo errores, y es bueno hacer preguntas!

Haz que toda la página web tenga un fondo rojo y utilice una fuente Georgia de 10 puntos en azul.	<pre>cuerpo{ color de fondo: rojo; familia de fuentes: Georgia; tamaño de fuente: 10pt; color: azul; }</pre>
Poner todos los párrafos en negrita y cursiva	<pre>p{ font- weight:negrita; font-style: cursiva; }</pre>
<p>Haz que todas las posiciones con la clase "rojo" se muestren con texto rojo. Puede dar a cualquier clase HTML un tag como este:</p> <pre><p class="red">Mi Texto </p></pre> <p>o</p> <pre> Google </pre>	<pre>.rojo{ color: rojo; }</pre>
<p>Aumentar los márgenes del párrafo de cabecera.</p> <p>En HTML: <code><p class="header">Mi Texto </p></code></p>	<pre>.header{ margen: 2em }</pre> <p><i>(La em es una unidad igual al ancho de una letra M)</i></p>

Ejercicio: diseñar un servicio de IA para resolver un problema

- Trabajar en **equipo**. Hay una opción para continuar trabajando en el proyecto que comenzaste en el primer día de nuestro taller. Sólo asegúrate de que tus compañeros de equipo estén contentos con esta opción también o puedes empezar de cero.
- Si ya conoces el **Objetivo de Desarrollo Sostenible** que te gustaría abordar, puede reducir su enfoque a una de las subcategorías llamadas "objetivos" para ser más específicos: <https://www.globalgoals.org> Simplemente abra la página de metas y busque los objetivos que le gustaría abordar.
- Haga un poco de **investigación** para entender mejor el problema que está tratando de resolver, especialmente para averiguar quiénes son las personas que más sufren de este problema y cuáles son sus necesidades apremiantes.
- **Haz una lluvia de ideas** sobre cómo la IA puede ayudarle a resolver este

Recuerda: ¡aprendemos cometiendo errores, y es bueno hacer preguntas!

problema. Piensa en cómo, dónde y en qué forma se puede aplicar la IA para mejorar la vida de estas personas o para salvar nuestro planeta.

Recuerda: ¡aprendemos cometiendo errores, y es bueno hacer preguntas!

- Aplica tus **superpoderes**: creatividad, empatía, inteligencia emocional, juicio, razonamiento, comunicación, trabajo en equipo y las habilidades de IA que aprendiste hoy.
- Piensa en cualquier **alternativa** existente a su idea y por qué su idea es mejor. También puede pensar en formas de mejorar las soluciones existentes para este problema o aumentar las capacidades humanas con la IA.
- Utiliza el marco de trabajo de **Diseño Centrado en el Ser Humano** para encontrar una solución significativa que responda a las necesidades reales de estas personas. Asegúrese de que su solución no sólo sea técnicamente viable, sino también accesible a todos los grupos desfavorecidos.
- Utiliza el marco de trabajo **Ética del Código**: <https://joinfuturemakers.com/ethics-of-code/> para asegurarse de que su solución no dañe o afecte a las personas de manera negativa o aumente la desigualdad. Trate de anticipar y planear cualquier consecuencia no intencional.
- Piensa en formas **sostenibles** de hacer que tu idea sea algo real, un producto que la gente de todo el mundo usará. ¿Cómo puede hacer que sea asequible para los grupos desfavorecidos? ¿Cómo se puede escalar de un país/región a muchos?
- Piensa en los **siguientes pasos**. ¿Qué vas a hacer después de que termine el taller?

Ejercicio: presentación de tu trabajo

- **Introducción**: Por favor, empieza presentándote
- **Problema**: ¿Qué problema estás tratando de resolver? Entonces, ¿qué / por qué crees que importa?
- **Personaje de usuario**: ¿Quiénes son las personas que más sufren de este problema? ¿Qué aspecto tiene un día de tu vida ahora?
- **Impacto**: ¿Cómo de grande es el impacto? ¿Cómo cambiaría la vida de estas personas?
- **ODS**: ¿Este problema forma parte de los Objetivos de Desarrollo Sostenible de la ONU?
- **Alternativas**: ¿Cuáles son las soluciones actuales y por qué no son suficientemente eficaces?
- **Tu idea**: ¿Cuál es su solución y cómo resuelve el problema? ¿En qué se diferencia de las soluciones existentes?
- **Diseño centrado en el ser humano**: ¿Qué necesita realmente la gente? ¿Qué es financieramente viable para ellos? ¿Qué es técnica y logísticamente factible?
- **La Ética del Código**: ¿Qué riesgos éticos ha identificado y cómo los ha abordado?
- **Sostenibilidad**: ¿Cuál es el objetivo a largo plazo de tu solución de IA? ¿Es compatible con alguno de los enfoques existentes? ¿Se puede escalar a través de diferentes regiones/esferas?
- **Próximos pasos**: ¿Qué planeas hacer con tu idea a continuación?

Recuerda: ¡aprendemos cometiendo errores, y es bueno hacer preguntas!

Ir más allá después del taller

Aprender más sobre la IA y el Machine Learning

Configuración de tu entorno de desarrollo en casa

Para que vuestros ordenadores portátiles y el servidor Flask de Anaconda funcionen, instala lo siguiente:

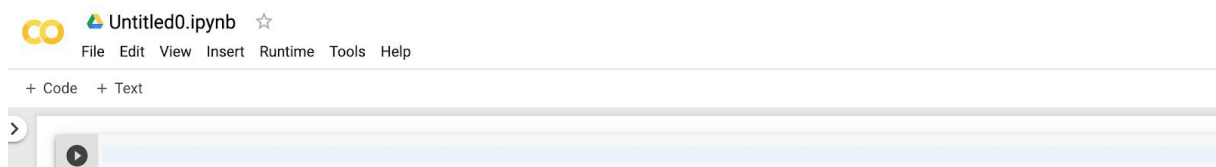
- Medio ambiente de Anaconda Python
Asegúrate de instalar la versión 3.7 de Python
<https://www.anaconda.com/download/>
- Windows para principiantes: <https://git-scm.com/download/win>
OS X: ver aquí
<https://gist.github.com/derhuerst/1b15ff4652a867391f03>
- Interfaz de línea de comandos de Heroku (CLI)
<https://devcenter.heroku.com/articles/heroku-cli>
- Asegúrate de que tanto Firefox como Chrome estén instalados
(para que puedas ejecutar tu aplicación web en una, y hacer búsquedas en Google o buscar documentación en la otra)
- <https://www.sublimetext.com/>
DIFUNDE LA PALABRA-

Uso de Google Jupyter Notebook

Google Jupyter Notebook le permite utilizar un Jupyter Notebook en línea sin necesidad de instalar nada en su ordenador.

Visite <https://colab.research.google.com/notebooks/welcome.ipynb> Cree un nuevo cuaderno en el menú 'Archivo'.

Google Jupyter Notebook tiene comandos ligeramente diferentes a los del portátil jupyter que has estado utilizando. La interfaz tiene el siguiente aspecto:



Para ejecutar la celda, haga clic en el botón Ejecutar a la izquierda de una celda:



Puedes crear una nueva celda de código haciendo clic en el botón de la parte superior izquierda:

Recuerda: ¡aprendemos cometiendo errores, y es bueno hacer preguntas!

+ Code

Observaste que el funcionamiento de este portátil requiere una conexión a Internet, por lo que podría ser más lento en comparación con el funcionamiento del portátil en tu propio ordenador. Esta es una gran herramienta si quieres compartir tu código con tus amigos y trabajar juntos en el mismo cuaderno!

Hacer que Heroku funcione

Una vez instalado el Heroku CLI:

- En Anaconda Prompt, navega hasta la carpeta del servidor. Asegúrate de que estás dentro de la carpeta del servidor y que puedes ver **server.py** cuando hagas dir.
- Ejecuta el **inicio de sesión de heroku**.
- Ejecuta **heroku apps:info** para ver tus aplicaciones y sus URLs. Luego puedes visitar las URLs para ver tus aplicaciones.
- Si has copiado correctamente su carpeta desde el taller, debería poder ejecutar **git add** .
git commit -m'deploy'
git push heroku
master
para implementar una nueva versión de su aplicación.

Próximos pasos - servidores web

Almacenamiento de datos sobre el usuario

Los servidores web **no tienen estado**: el servidor no recuerda nada entre peticiones. Una vez que ha respondido a una solicitud, se olvida de todo lo que ocurrió en esa solicitud. Esto es para que pueda mantener una flota de servidores web, cualquiera de los cuales puede atender cualquier solicitud; si uno se bloquea, no se pierden datos.

Entonces, ¿dónde almacenamos nuestros datos?

Todos los datos que una aplicación web necesita -desde cuentas de usuario hasta páginas de perfil- se almacenan normalmente en un servidor separado, el servidor de base de datos. Se trata de un servidor especializado que no responde a las peticiones web, sólo les permite almacenar y leer datos. El programa de base de datos más utilizado se llama Postgres.

En este caso, haremos "trampa" al no utilizar una base de datos: almacenaremos nuestros datos en la **sesión**, un pequeño archivo que se guarda dentro del navegador del usuario en una cookie. Así que todos los datos que el servidor necesita recordar no se almacenan en el servidor o en la base de datos, sino en el navegador del usuario.

Base de datos

Recuerda: ¡aprendemos cometiendo errores, y es bueno hacer preguntas!

Recuerda que no puedes almacenar ninguna información en el servidor web (porque muchos servidores y sistemas de archivos son efímeros). En este momento estamos almacenando información en la sesión del usuario; en su lugar deberíamos hacerlo en una base de datos.

El trabajo de la base de datos es almacenar todos los datos que son importantes para nuestra aplicación. Hablamos con la base de datos usando un lenguaje llamado SQL (Structured Query Language).

Necesitarás dos bases de datos: una local para la prueba y otra de producción con la que el servidor web pueda hablar. Es mejor usar el mismo tipo de base de datos para ambos.

Heroku viene con una base de datos gratuita llamada Postgres (que es un viejo proyecto de código abierto de Berkeley, California).

Tutorial:

<https://devcenter.heroku.com/categories/heroku-postgres>

También puedes instalar Postgres localmente. Las guías de instalación están aquí:

https://wiki.postgresql.org/wiki/Detailed_installation_guides

Flask usa una librería llamada Flask-SQLAlchemy para hablar con la base de datos.

Ya sea local o de producción, necesitará una URL de la base de datos; configure esto en Flask para asegurarse de que Flask está hablando con la base de datos correcta.

Javascript

Todo el código que hemos escrito hasta ahora se ha ejecutado en el servidor. Puedes instalar archivos Javascript en su aplicación web que se ejecutan en el navegador del usuario; pueden hacer cosas como crear ventanas emergentes y animar partes de la página.

El mejor marco de trabajo de Javascript para empezar se llama JQuery; busca cómo usar la función de preparación de documentos (una función que se llama cuando la página está lista, en la que puedes colocar tu código).

Tutorial: <https://www.tutorialspoint.com/jquery/>