Programación web de CS50 con Python y JavaScript

OpenCourseWare

Donar (https://cs50.harvard.edu/donate)

Brian Yu (https://brianyu.me)
brian@cs.harvard.edu

David J. Malan (https://cs.harvard.edu/malan/)
malan@harvard.edu
(https://www.facebook.com/dmalan) (https://github.com/dmalan)
(https://www.instagram.com/davidjmalan/) (https://www.linkedin.com/in/malan/)
(https://www.reddit.com/user/davidjmalan)
(https://www.threads.net/@davidjmalan) (https://twitter.com/davidjmalan)

Buscar

> Las conferencias 1 y 2 no son necesarias para completar este proyecto, pero pueden resultar útiles si tienes dificultades para *enviarlo* , ya que enseñan Git, entre otros temas.

Diseñe una interfaz para la Búsqueda de Google, la Búsqueda de imágenes de Google y la Búsqueda avanzada de Google.

Fondo

Recuerde de la conferencia que podemos crear un formulario HTML usando una `<form>` etiqueta y podemos agregar `<input>` etiquetas para crear campos de entrada para ese formulario. Más adelante en el curso, veremos cómo escribir aplicaciones web que puedan aceptar datos de formulario como entrada. Para este proyecto, escribiremos formularios que envíen datos a un servidor web existente: en este caso, el de Google.

Cuando realiza una búsqueda en Google, como escribir una consulta en la página de inicio de Google y hacer clic en el botón "Búsqueda de Google", ¿cómo funciona esa consulta? Intentemos averiguarlo.

Navegue a google.com (https://www.google.com/) , escriba una consulta como "Harvard" en el campo de búsqueda y haga clic en el botón "Búsqueda de Google".

Como probablemente esperaba, debería ver los resultados de búsqueda de Google para "Harvard". Pero ahora, eche un vistazo a la URL. Se debe comenzar por `https://www.google.com/search`, la ruta en el sitio web de Google que permite realizar búsquedas. Pero seguir la ruta es un `?`, seguido de información adicional.

Esa información adicional en la URL se conoce como cadena de consulta. La cadena de consulta consta de una secuencia de parámetros GET, donde cada parámetro tiene un nombre y un valor. Las cadenas de consulta generalmente tienen el formato

```
field1=value1&field2=value2&field3=value3...
```

donde `=` separa el nombre del parámetro de su valor y el `&` símbolo separa los parámetros entre sí. Estos parámetros son una forma para que los formularios envíen información a un servidor web, codificando los datos del formulario en la URL.

Take a look at the URL for your Google search results page. It seems there are quite a few parameters that Google is using. Look through the URL (it may be helpful to copy/paste it into a text editor), and see if you can find any mention of “Harvard,” our query.

If you look through the URL, you should see that one of the GET parameters in the URL is `q=Harvard`. This suggests that the name for the parameter corresponding to a Google search is `q` (likely short for “query”).

It turns out that, while the other parameters provide useful data to Google, only the `q` parameter is required to perform a search. You can test this for yourself by visiting `https://www.google.com/search?q=Harvard`, deleting all the other parameters. You should see the same Google results!

Using this information, we can actually re-implement a front end for Google’s homepage. Paste the below into an HTML file called `index.html`, and open it in a browser. You can alternatively download the `index.html` file directly from the “Getting Started” section below.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Search</title>
    </head>
    <body>
        <form action="https://www.google.com/search">
            <input type="text" name="q">
            <input type="submit" value="Google Search">
        </form>
    </body>
</html>
```

When you open this page in a browser, you should see a (very simple) HTML form. Type in a search query like "Harvard" and click "Google Search", and you should be taken to Google's search results page!

How did that work? In this case, the `action` attribute on the `form` told the browser that when the form is submitted, the data should be sent to `https://www.google.com/search`. And by adding an `input` field to the form whose `name` attribute was `q`, whatever the user types into that input field is included as a GET parameter in the URL.

Your task in this project is to expand on this site, creating your own front end for Google Search, as by exploring Google's interface to identify what GET parameters it expects and adding the necessary HTML and CSS to your website.

Getting Started

- Download the distribution code from https://cdn.cs50.net/web/2020/spring/projects/0/search.zip (https://cdn.cs50.net/web/2020/spring/projects/0/search.zip) and unzip it. You can skip this step if you manually created the `index.html` file by following the steps outlined in the "Background" section above.

Specification

Your website must meet the following requirements:

- Your website should have at least three pages: one for regular Google Search (which must be called `index.html`), one for Google Image Search, and one for Google Advanced Search.
    - On the Google Search page, there should be links in the upper-right of the page to go to Image Search or Advanced Search. On each of the other two pages, there should be a link in the upper-right to go back to Google Search.
- On the Google Search page, the user should be able to type in a query, click "Google Search", and be taken to the Google search results for that page.
    - Like Google's own, your search bar should be centered with rounded corners. The search button should also be centered, and should be beneath the search bar.
- On the Google Image Search page, the user should be able to type in a query, click a search button, and be taken to the Google Image search results for that page.
- On the Google Advanced Search page, the user should be able to provide input for the following four fields (taken from Google's own advanced search (https://www.google.com/advanced_search) options)
    - Find pages with… "all these words:"
    - Find pages with… "this exact word or phrase:"

    - Find pages with… "any of these words:"
    - Find pages with… "none of these words:"
- Like Google's own Advanced Search page, the four options should be stacked vertically, and all of the text fields should be left aligned.
    - Consistent with Google's own CSS, the "Advanced Search" button should be blue with white text.
    - When the "Advanced Search" button is clicked, the user should be taken to the search results page for their given query.
- Add an "I'm Feeling Lucky" button to the main Google Search page. Consistent with Google's own behavior, clicking this link should take users directly to the first Google search result for the query, bypassing the normal results page.
    - You may encounter a redirect notice when using the "I'm Feeling Lucky" button. Not to worry! This is an expected consequence of a security feature implemented by Google.
- The CSS you write should resemble Google's own aesthetics.

Hints

- To determine what the parameter names should be, you're welcome to experiment with making Google searches, and looking at the resulting URL. It may also be helpful to open the "Network" inspector (accessible in Google Chrome by choosing View -> Developer -> Developer Tools) to view details about requests your browser makes to Google.
    - Any `<input>` element (whether its `type` is `text`, `submit`, `number`, or something else entirely) can have `name` and `value` attributes that will become GET parameters when a form is submitted.
    - You may also find it helpful to look at Google's own HTML to answer these questions. In most browsers, you can control-click or right-click on a page and choose "View Page Source" to view the page's underlying HTML.
- To include an input field in a form that users cannot see or modify, you can use a "hidden" (https://www.w3schools.com/tags/att_input_type_hidden.asp) input field.

How to Submit

1. Visit this link (https://submit.cs50.io/invites/89679428401548238ceb022f141b9947), log in with your GitHub account, and click **Authorize cs50**. Then, check the box indicating that you'd like to grant course staff access to your submissions, and click **Join course**.
2. Install Git (https://git-scm.com/downloads) and, optionally, install `submit50` (https://cs50.readthedocs.io/submit50/).

Cuando envíe su proyecto, el contenido de su `web50/projects/2020/x/search` rama debe coincidir con la estructura de archivos del código de distribución descomprimido tal como se recibió originalmente. Es decir, sus archivos no deben estar anidados dentro de ningún otro directorio de su propia creación (`search` o `project0`, por ejemplo). Su rama tampoco debe contener ningún código de ningún otro proyecto, solo este. Si no cumple con esta estructura de archivos, es probable que su envío sea rechazado.

A modo de ejemplo, para este proyecto eso significa que si el personal de calificación lo visita `https://github.com/me50/USERNAME/blob/web50/projects/2020/x/search/index.html` (donde `USERNAME` está su propio nombre de usuario de GitHub como se proporciona en el formulario a continuación), su envío para `index.html` este proyecto debe ser lo que aparece. Si no es así, reorganice su repositorio según sea necesario para que coincida con este paradigma.

3. Si lo ha instalado `submit50`, ejecute

   ```
   submit50 web50/projects/2020/x/search
   ```

   De lo contrario, usando Git, envíe su trabajo a `https://github.com/me50/USERNAME.git`, donde `USERNAME` está su nombre de usuario de GitHub, en una rama llamada `web50/projects/2020/x/search`.
4. Grabe un screencast (https://www.howtogeek.com/205742/how-to-record-your-windows-mac-linux-android-or-ios-screen/) que no exceda los 5 minutos de duración (y no lo cargue más de un mes antes de enviar este proyecto), en el que demuestre la funcionalidad de su proyecto. **Su barra de URL debe permanecer visible durante la demostración del proyecto.** Asegúrese de que cada elemento de la especificación anterior se demuestre en su video. No es necesario mostrar su código en este video, solo su aplicación en acción; Revisaremos su código en GitHub. Sube ese video a YouTube (https://www.youtube.com/upload) (como no listado o público, pero no privado) o en otro lugar. En la descripción de su video, debe marcar la hora en la que su video demuestra cada uno de los siete (7) elementos de la especificación. **Esto no es opcional** , los videos sin marcas de tiempo en su descripción serán rechazados automáticamente.
5. Envíe este formulario (https://forms.cs50.io/31c21989-e2e4-4939-8e1a-8f4012133fa5) .

Luego puede ir a https://cs50.me/cs50w (https://cs50.me/cs50w) para ver su progreso actual.