



Girls Who Code en casa

Excursión virtual
Aplicación web interactiva con JavaScript

Resumen de la actividad

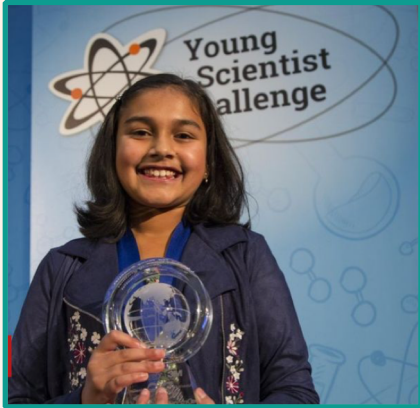
El Día Internacional de la Madre Tierra es el 22 de abril, por lo que queremos animarte a pensar en algunos de los lugares silvestres y al aire libre que te gustaría explorar. ¿Adónde irías? ¿Qué te gustaría hacer y ver? En este tutorial, aprenderás a usar **arreglos** o listas en JavaScript para crear una excursión virtual. Crearás un sitio web capaz de llevar a una persona de visita a un parque nacional, tu museo favorito u otro país. ¡Las posibilidades son ilimitadas! Antes de que comiences a trabajar en tu itinerario, te recomendamos que leas el artículo destacado de “Women in Tech” sobre Gitanjali Rao.

Nota: Cuando desarrollan un sitio web, muchos programadores utilizan HTML para crear el esqueleto del sitio, CSS para darle estilo a la página y JavaScript para añadir interactividad. En esta actividad, trabajaremos principalmente con JavaScript. Si quieres un repaso rápido de HTML y CSS, te recomendamos ver nuestra tercera actividad: [Comparte tus talentos](#), antes de comenzar esta actividad.

Materiales

- [Glitch](#) o un editor de textos de tu elección
- [Ejemplo de proyecto de excursión virtual \(con extensiones\)](#)
- [Ejemplo de proyecto de excursión virtual \(sin extensiones\)](#)
- [Código inicial de la excursión virtual](#)
- Opcional: Guía de planificación
- Opcional: Lápiz, lapicera o marcadores

Artículo destacado de “Women in Tech”: Gitanjali Rao



Fuente de la imagen: [BBC](#)

En 2014, hubo una crisis de salud pública en Flint, Michigan. El suministro de agua de Flint contenía plomo, que es muy peligroso si se consume, en especial para niños pequeños. Después de enterarse de la crisis de salud, Gitanjali puso manos a la obra y creó Tethys, un aparato que detecta si hay contaminación en el agua.


Gitanjali tenía solo 11 años cuando creó Tethys. Ganó el [Reto para Científicos Jóvenes 3M de Discovery Education](#) 2017 y fue invitada como oradora de TEDx.

Su dispositivo utiliza un nanotubo de carbono para detectar el flujo de electrones en plomo, además de usar una señal basada en Arduino que se conecta a una aplicación de teléfono inteligente mediante Bluetooth.

Ve el [video](#) sobre Gitanjali Rao y su invento, Tethys. Si tienes tiempo, te recomendamos que aprendas más sobre Gitanjali explorando [este artículo](#) de Bustle, [este artículo](#) de Young Scientist Lab o [este artículo](#) de NPR. También puedes ver su discurso de TEDx [aquí](#).

Reflexión

Ser una experta informática significa mucho más que simplemente ser buena programando. Toma unos minutos para reflexionar sobre cómo Gitanjali y su trabajo reflejan las características que todos los verdaderos expertos informáticos deben desarrollar : valentía, resistencia, creatividad y propósito.

 PROPÓSITO	A Gitanjali le interesa el ámbito de la medicina, pero también aprendió a programar y a crear tecnología. ¿Qué otros ámbitos te interesan donde quizá pudieras usar las ciencias de la computación?
---	---

Comparte tus respuestas con un familiar o amigo. Anima a otros para que lean sobre Gitanjali y se unan al debate.

Primer Paso: Explora (5 minutos)

En este tutorial, crearás un sitio web que presenta una serie de imágenes para guiar al usuario por una excursión virtual. Toma 5 minutos para explorar las características de este ejemplo de [sitio web](#) que creamos sobre sitios naturales en Japón.

- **Tema del proyecto:** Naturaleza.
- **Audiencia:** Personas que quieren conocer más sobre lugares donde pueden admirar la naturaleza en Japón.
- **Objetivo:** Compartir una experiencia fuera de casa y planear un itinerario por Japón centrado en la naturaleza.

Al explorar este sitio web, piensa en las siguientes preguntas:

- ¿Qué sucederá al hacer clic en cada botón?
- ¿Las fotografías seleccionadas se relacionan con el tema y el objetivo? ¿Qué fotografías te gustaría incorporar?
- ¿Cuáles crees que son los pasos al hacer clic en el botón “Next” (Siguiente)? Intenta dividir los pasos en pasos más pequeños.
- ¿Cuándo deja de funcionar el botón “Next” (Siguiente)? ¿Cuándo deja de funcionar el botón “Back” (Atrás)? ¿Por qué podría suceder esto?

Segundo Paso: Genera ideas sobre las características y planifica el proyecto (10 minutos)

Ya que has tenido la oportunidad de explorar un ejemplo de proyecto, deberías tomar un poco de tiempo para elaborar un plan de juego. Aprovecha este tiempo para determinar qué quieres que haga tu proyecto y cuál será tu meta. Tal vez quieras usar el documento de planificación al final de esta actividad para captar y organizar tus ideas.

1. Selecciona un tema, una audiencia y una meta para tu excursión virtual.

En honor del Día Internacional de la Madre Tierra (22 de abril de 2020), te aconsejamos que selecciones un lugar al aire libre que te gustaría explorar, pero puedes seleccionar cualquier tema que gustes. Si te sientes sin inspiración, ve aquí algunas ideas alternativas para tu proyecto:

- Una galería virtual de actividades o pasatiempos que te gusta hacer en casa.
- Una caminata virtual por tu museo favorito.
- Una excursión virtual a tus lugares favoritos.

Al seleccionar la audiencia de tu excursión virtual, piensa en quién será tu público objetivo. ¿A quién le emocionaría usar tu sitio web? Piensa en lo que les interesaría conocer y cómo captarás su atención.

2. Escoge por lo menos tres imágenes.

Vas a querer incluir por lo menos tres imágenes para mostrarlas en tu recorrido virtual. Puedes usar imágenes propias o que encuentres en línea. Al escoger las imágenes, piensa en el título o texto breve que querrás mostrar con cada imagen.

Tercer Paso: Comenzar en Glitch (10 minutos)

Glitch es una herramienta sencilla para la creación de aplicaciones web. Viene con un editor de texto que te permite ver los cambios en tu página web en vivo. También te permite publicar tu proyecto fácilmente, para que todo el mundo lo pueda ver. Si encuentras un proyecto genial creado por otra persona, puedes ver su código y hacer un remix con él.

1. Inicia sesión en [Glitch](#) usando tu cuenta de Google, Facebook o GitHub.

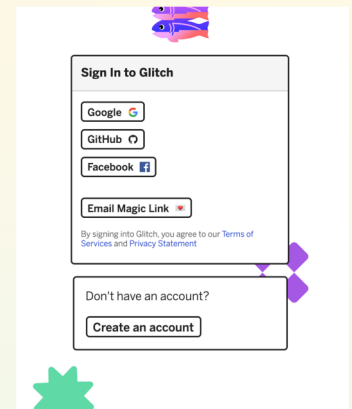
- Para guardar y compartir tu trabajo en Glitch, tendrás que iniciar sesión. Glitch te permite iniciar sesión usando tu cuenta de Google, Facebook o GitHub. Si eres menor de 13 años, necesitarás la autorización y la dirección de correo electrónico de tu padre o tutor para registrarte.

2. Adapta este [código inicial](#) para tu sitio web de excursión virtual. Ve este [video](#) para obtener instrucciones más detalladas.

3. Explora la interfaz de Glitch.

Lo primero que verás será el contenido del archivo **README.md**. Los archivos README (Léeme) son documentos que verás de forma habitual en los proyectos de otras personas. Normalmente contienen información sobre cómo están estructurados los archivos del proyecto y cómo ejecutar el programa. Ve este [video](#) para conocer más sobre la interfaz de Glitch.

4. Arma la vista de tu proyecto. Haz clic en el botón **Show** (Mostrar) y luego selecciona la opción **Next To The Code** (Junto al código). Esto te permitirá ver lado a lado el código y el aspecto del proyecto en el sitio web. Ve este [video](#) para obtener instrucciones más detalladas.



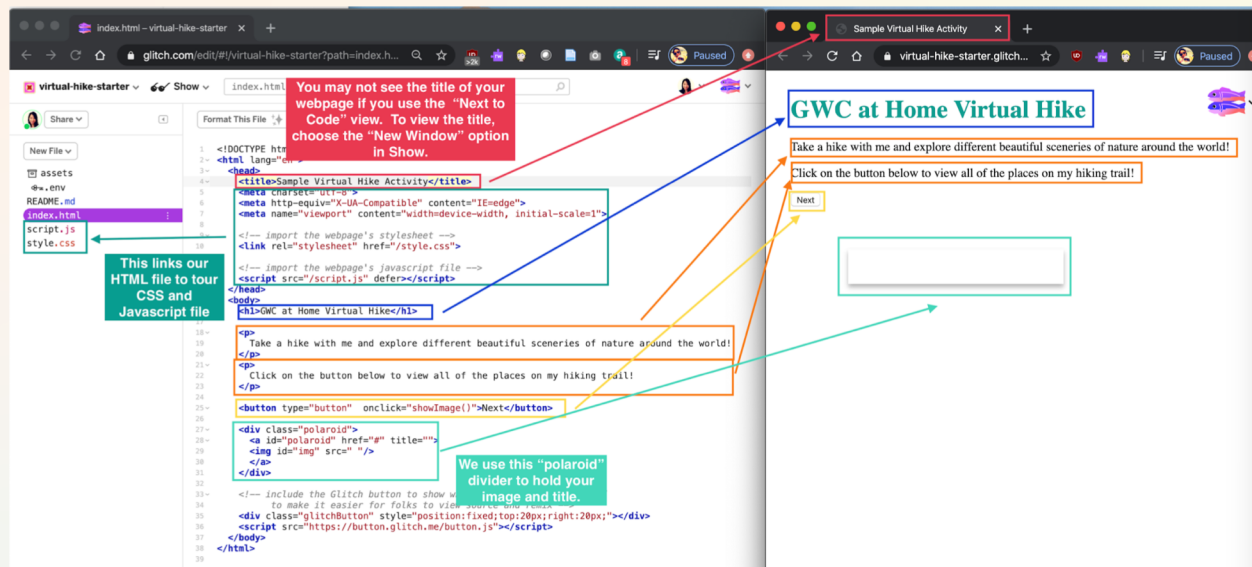
Cuarto Paso: Explora el código HTML del proyecto inicial (10 minutos)

Veamos primero el archivo `index.html`. Haz clic en el archivo `index.html` en el menú de navegación de la izquierda. **HTML**, que significa “Hypertext Markup Language” (lenguaje de marcas de hipertexto), utiliza **etiquetas** para organizar el contenido de un sitio web. Todo el contenido que aparece en tu ventana de vista previa online está incluido dentro de una etiqueta de apertura `<body>` y otra etiqueta de cierre `</body>`. Las etiquetas son parte de la anatomía de un elemento HTML, que normalmente está compuesto por la etiqueta de apertura, el contenido y la etiqueta de cierre. Observa las siguientes imágenes para ver algunos ejemplos de etiquetas HTML para diversos tipos de elementos. La etiqueta `<p>` se usa para indicar “párrafo”, mientras que la etiqueta `<a>` es una más genérica que por lo general se asigna a enlaces.



Cuarto Paso: Explora el código HTML del proyecto inicial (continuación)

La imagen siguiente relaciona las etiquetas que ves en el código inicial con el aspecto que tendrán en el sitio web.



Dedicamos tiempo a la personalización de la información de tu sitio web.

¡Intenta programar tú misma en HTML!

- 1. Cambia el título del sitio web.** El título del sitio web es el nombre que aparece al cargar el sitio. Ten presente que esto no es lo mismo que cambiar la dirección (URL) del sitio web. Si quieres comprobar que el título de tu página web esté correcto, haz clic en **Show** (Mostrar) y selecciona la opción **New Window** (Ventana nueva) en el menú.
 - Localiza la etiqueta `<title>` en el archivo HTML. (Se señala con el cuadro rojo en la imagen de arriba).
 - Reemplaza el texto entre la etiqueta de apertura `<title>` y la etiqueta de cierre `</title>` por el nombre de tu página web.
Ejemplo: `<title> Nombre de mi nueva página web </title>`
- 2. Cambia el encabezamiento del sitio web.** Un encabezamiento es el texto que usualmente aparece en la parte superior de la página. A diferencia del título de la página web, el encabezado será visible en la página.
 - Localiza la etiqueta `<h1>` en el archivo HTML. (Se señala con el cuadro azul en la imagen de arriba).
 - Reemplaza el texto entre la etiqueta de apertura `<h1>` y la etiqueta de cierre `</h1>` por el nombre de tu proyecto. Podría ser lo mismo que escribiste para el título de la página web.
Ejemplo: `<h1> Nombre de mi nueva página web </h1>`

Cuarto Paso: Explora el código HTML del proyecto inicial (continuación)

3. **Cambia el texto del sitio web.** El ejemplo de código tiene dos párrafos. El primero contiene una descripción breve del proyecto, mientras que el segundo le presenta al usuario instrucciones para interactuar con el sitio web.

- Localiza la etiqueta `<p>` en el archivo HTML. (Se señala con el cuadro anaranjado en la imagen de arriba).
- Reemplaza el texto entre el **primer conjunto** de etiqueta de apertura `<p>` y etiqueta de cierre `</p>` por una descripción breve de tu proyecto. Consulta lo que escribiste en tu documento de planificación.

Ejemplo: `<p>`

Descripción de texto de mi nueva página web

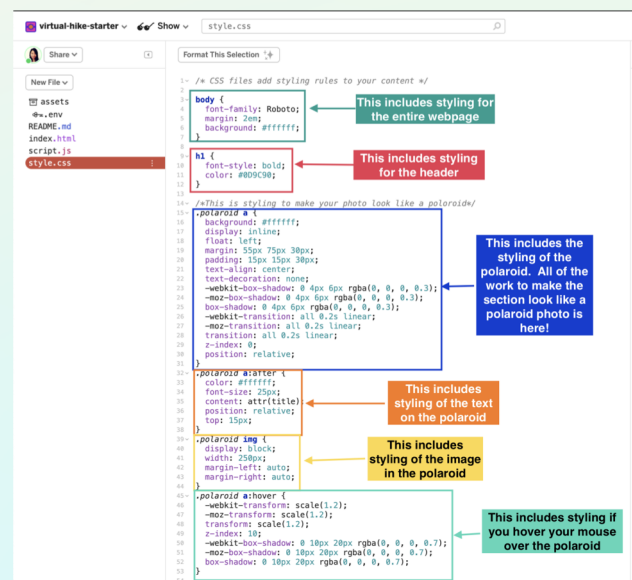
`</p>`

- **(Opcional)** Personaliza las instrucciones en el segundo conjunto de etiquetas `<p>`.

Quinto Paso: Explora el código CSS del proyecto inicial (5 minutos)

CSS, o **Cascading Style Sheets** (hojas de estilo en cascada), describe las reglas (o los estilos) de presentación que deben aplicarse a los elementos HTML. CSS te permite cambiar el aspecto del contenido de tu sitio web, incluido el color o tamaño del texto, las fuentes usadas, los colores o imágenes de fondo y mucho más. En este tutorial, no profundizaremos en CSS. Te recomendamos que veas [este](#) recurso o que realices nuestra actividad [Comparte tus talentos](#) si te interesa aprender más sobre CSS.

En este proyecto inicial, ya configuramos varias reglas básicas de estilos CSS para que puedas enfocarte en la personalización y el aprendizaje de JavaScript. Incluimos estilos para el título, el encabezamiento y polaroid. Nombramos uno de los elementos **polaroid** porque añadimos reglas de estilo en CSS para que tus fotos parecieran tomadas con una cámara Polaroid. Los estilos de polaroid se tomaron de [este](#) tutorial. Al final de este proyecto, también tendrás tiempo de personalizar el sitio web con los colores y los tipos de letra que prefieras.



Fuente de la imagen: [Creative Market](#)

Sexto Paso: Introducción a JavaScript (2 minutos)

¡Ahora viene lo divertido! Hasta ahora, has construido el esqueleto de tu sitio web, pero este aún no hace nada. Es aquí donde **JavaScript** desempeña un papel. JavaScript se usa para que un sitio web sea interactivo. Veamos el archivo `script.js`. Hemos añadido algunas cosas para ayudarte a comenzar, así como varios comentarios que explican qué hacen las líneas de código.

```
1  /* If you're feeling fancy you can add interactivity
2  to your site with Javascript */
3
4  //This is an array that will hold the file names! Add at least 4 images to the images array.
5  var images = new Array();
6  images[0] = "";
7  //This is an array that holds the names of each location shown in your pictures.
8  //Make sure that the index of each location is the same as the index of the picture in the images array
9  var locations = new Array();
10 locations[0] = "";
11
12 //This is your starting index. First we start off at Home!
13 var index = 0;
14
15 //This function will help us toggle through our photos. Everytime the button is pressed it will call this function!
16 function showImage() {
17
18 }
19
```

Nota: La computadora no lee los comentarios - describen el código para otras personas y para ti cuando vuelvas a leerlo en el futuro. Puedes añadir un comentario en una línea tecleando `//` y un mensaje.

Septimo Paso: Introducción a las variables y los arreglos (2 minutos)

Para guardar información (como las fotos y sus descripciones), necesitamos usar una **variable**. Una variable es un contenedor que almacena todo tipo de datos. Puede contener palabras, número e incluso una lista de datos. Veamos algunos de los símbolos clave (o sintaxis) para el uso de variables:

`var miVariable = valor;`

- **var** : Esta palabra clave se usa para declarar un tipo de variable.
- **miVariable** : El nombre de nuestra variable. Para los nombres por lo general se utiliza lo que se conoce como estilo de camello: sin espacio y con letras mayúsculas para indicar palabras nuevas.
- **=** : el signo de igualdad muestra la asignación o reasignación de un valor a la variable
- **valor** : Puede ser cualquier valor: número, palabras o incluso una lista de datos.
- **;** : todos las declaraciones de JavaScript deben terminar con dos puntos. De esta manera, la computadora sabe que ha finalizado el comando.

Un **arreglo** es una estructura de datos ordenados que puede contener varios tipos de información. Puedes pensar en un arreglo como un tocador con varios cajones, cada uno de los cuales puede contener datos. Veamos los símbolos clave (o sintaxis) que usamos para crear un arreglo **vacío** (como en el código inicial):

`var miArreglo = new Array();`

- **var** : Usamos la palabra clave var para guardar el arreglo en una variable.
- **new** : Esta palabra clave se usa para indicar que queremos crear un objeto vacío. En nuestro caso, un arreglo vacío.
- **Array** : Esta palabra clave le indica a la computadora que cree un objeto de arreglo.
- **()** : Los paréntesis se usan para invocar la **función** de arreglo. JavaScript ya tiene una función (o serie de comandos) integrada que ayuda a crear el objeto de arreglo. Aprenderemos más sobre las funciones y cómo se usan en una parte posterior de este proyecto.

Octavo Paso: Cómo usar arreglos (10 minutos)

Los arreglos son una de las estructuras de datos más utilizadas, ya que son estructuras *ordenadas*. Un arreglo le asigna un número a cada elemento que guarda y nos referimos a este número como índice. La asignación de un valor de índice a cada elemento facilita acceder, eliminar o reemplazar valores guardados en el arreglo. En ciencias de la computación, comenzamos el recuento del índice con 0. Al primer elemento del arreglo se le asignará el índice 0 y este índice se incrementa en 1 para cada elemento subsiguiente. Veamos un ejemplo:

```
var programadoras = ["Ada", "Grace", "Katherine", "Roya"];
```

En este ejemplo, tenemos el arreglo llamado `programadoras`, que contiene los nombres de programadoras famosas: `"Ada"`, `"Grace"`, `"Katherine"`, `"Roya"`. En total, este arreglo tiene una longitud de 4. Los corchetes `[]` se usan para indicar donde comienza y termina el contenido del arreglo.

0	1	2	3
"Ada"	"Grace"	"Katherine"	"Roya"

Digamos que queremos acceder al nombre "Grace" en el arreglo. ¿Cómo lo haríamos?

Podemos usar el índice asociado con "Grace" para obtener esta información. Solo tendríamos que escribir la declaración: `programadoras[1]` y nos devolvería el nombre "Grace". Primero (1) escribimos el **nombre** del arreglo que queremos acceder, (2) añadimos corchetes `[]` y (3) colocamos dentro de los corchetes el número de índice que queremos acceder; como queremos "Grace", ponemos el número 1.

Si escribiéramos `programadoras[3]`, nos devolvería el nombre "Roya". Si escribiéramos `programadoras[4]`, obtendríamos un **error**. Los arreglos son **indexados a 0**, lo que significa que el índice comienza en 0. Aunque hay 4 nombres en la lista, el último nombre tiene un índice que es *uno menos que la longitud de la lista*.

También podemos usar el índice para añadir o cambiar valores en el arreglo. Digamos que quisiéramos añadir el nombre "Gitanjali" al arreglo. Podemos escribir la siguiente sintaxis:

```
programadoras[4] = "Gitanjali";
```

Noveno Paso: Añade tu primera imagen y lugar (10 minutos)

Observa el archivo `script.js`. Verás que utilizamos la sintaxis presentada arriba para crear dos arreglos, uno llamado `images` (imágenes) y otro llamado `locations` (lugares).

```
/* If you're feeling fancy you can add interactivity
   to your site with Javascript */

//This is an array that will hold the file names! Add at least 4 images to the images array.
var images = new Array();
images[0] = "";
//This is an array that holds the names of each location shown in your pictures.
//Make sure that the index of each location is the same as the index of the picture in the images array
var locations = new Array();
locations[0] = "";

//This is your starting index. First we start off at Home!
var index = 0;

//This function will help us toggle through our photos. Everytime the button is pressed it will call this function!
function showImage() {
}
```

- `images` será un arreglo con las direcciones de las imágenes, es decir, los enlaces de las fotos que quieres incluir.
- `locations` será un arreglo con la ubicación geográfica de cada imagen (o la descripción, si eliges un tema alternativo).

Al añadir cada foto y su título a los arreglos `images` y `locations`, queremos asegurarnos de que los valores de índice estén alineados en los dos arreglos. Esto significa que si colocamos una imagen de la aurora boreal en el índice 0 del arreglo `images`, queremos incluir el texto del título “Aurora boreal” en el índice 0 del arreglo `locations`.

Para incluir una foto en el arreglo, colocamos la **dirección URL de la imagen**.

- Si usarás imágenes en línea, te recomendamos anotar la dirección URL de la imagen en tu **guía de planificación**. Para obtener la dirección de una imagen, sigue las indicaciones de este [video](#).
- Si usarás tus propias imágenes, asegúrate de que estén guardadas *digitalmente* en tu computadora. Para cargar imágenes en Glitch, ve este [video](#).

Ahora que tienes las direcciones URL de las imágenes, agrégalas al arreglo `images`. Navega de vuelta al archivo `script.js`. Como es tu primera imagen, agregaremos su dirección en el índice 0. Pega la dirección URL de la imagen entre los símbolos “ ” en la línea 6 del código inicial.

```
1~ /* If you're feeling fancy you can add interactivity
2   to your site with Javascript */
3
4 //This is an array that will hold the file names! Add at least 4 images to the images array.
5 var images = new Array();
6 images[0] = "https://cdn.glitch.com/5b718117-0dff-4b23-9a99-5b6ff3f4ee3d%2FNorthern%20Lights.jpg?v=1586357262455";
7 //This is an array that holds the names of each location shown in your pictures.
8 //Make sure that the index of each location is the same as the index of the picture in the images array
9 var locations = new Array();
10 locations[0] = "";
11
12 //This is your starting index. First we start off at Home!
13 var index = 0;
14
15 //This function will help us toggle through our photos. Everytime the button is pressed it will call this function!
16~ function showImage() {
17
18 }
19
```

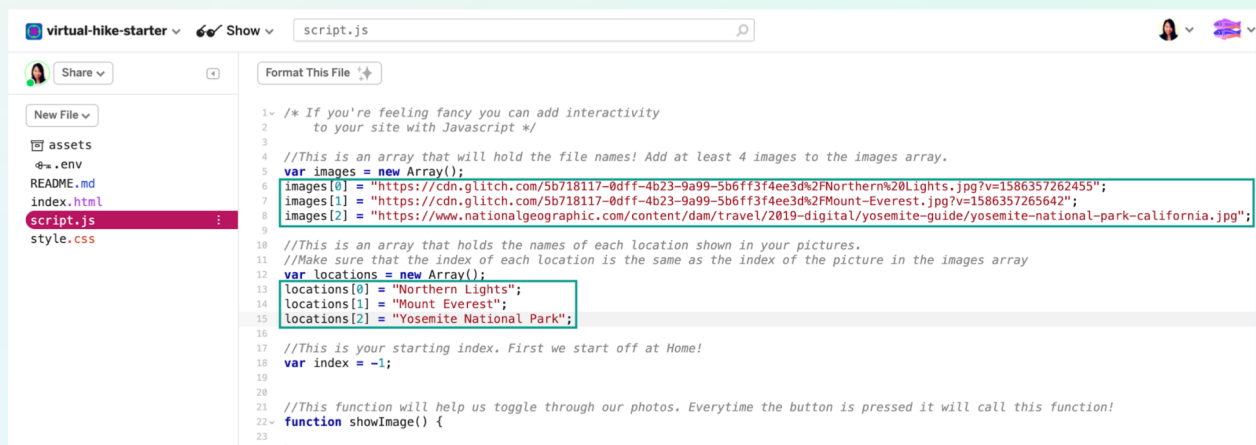
Noveno Paso: Añade tu primera imagen y lugar (continuación)

A continuación, queremos añadir el lugar o título de la imagen que acabas de cargar. Agrega el título de la imagen que añadiste entre las comillas dobles `" "` en el índice 0 del arreglo `locations`.

```
1~ /* If you're feeling fancy you can add interactivity
2   to your site with Javascript */
3
4 //This is an array that will hold the file names! Add at least 4 images to the images array.
5 var images = new Array();
6 images[0] = "https://cdn.glitch.com/5b718117-0dff-4b23-9a99-5b6ff3f4ee3d%2FNorthern%20Lights.jpg?v=1586357262455";
7 //This is an array that holds the names of each location shown in your pictures.
8 //Make sure that the index of each location is the same as the index of the picture in the images array
9 var locations = new Array();
10 locations[0] = "Northern Lights";
11
12 //This is your starting index. First we start off at Home!
13 var index = 0;
14
15 //This function will help us toggle through our photos. Everytime the button is pressed it will call this function!
16~ function showImage() {
17
18 }
19
```

Décimo Paso: Añade más imágenes y lugares (10 minutos)

Ahora que has añadido el primer par de imágenes y títulos de lugar, es hora de añadir las demás imágenes. Sigue las instrucciones del noveno paso para añadir más imágenes y texto. Al añadir la dirección URL de la imagen al arreglo `images`, recuerda que debes añadir su título con el **mismo índice** en el arreglo `locations`. Después de añadir todas tus imágenes y lugares, tu código debe parecerse al siguiente imagen.



Ya has añadido las imágenes, pero ¡no sucede nada al presionar el botón “Next” (Siguiente)! Aunque tenemos una manera de guardar estas imágenes, aún no hemos configurado un comando para acceder a ellas. Cambiemos esta situación.

Undécimo Paso: Introducción a las funciones (5 minutos)

Comencemos por ver la etiqueta del botón. Navega hasta el archivo `index.html` y encuentra esta línea de código.

```
<button type="button" onclick="showImage()">Next</button>
```

Observa que el atributo `onclick` es igual a `showImage()`. El atributo `onclick` se usa para indicarle a la computadora qué debe hacer cuando se haga clic en el botón. En este caso, el botón debe invocar o ejecutar la **función** `showImage()`. La función `showImage()` está escrita en nuestro archivo de JavaScript. Pero antes de describir qué hace `showImage()`, veamos qué es una función.

Una **función** es una sección de código, con un nombre asociado, que lleva a cabo una serie de instrucciones (o línea de código). Por ejemplo, si te dijeran “poner la mesa”, ya sabes que esto significa colocar en la mesa un mantel individual para cada una de las personas en tu hogar, luego colocar los utensilios (tenedores, cucharas, cuchillos o palillos) para cada lugar y luego poner una servilleta en cada lugar. **Poner la mesa** es una **función** y las tareas individuales necesarias para realizarla están incluidas dentro de la función. Por supuesto, las instrucciones para poner la mesa podrían ser diferentes en tu casa. Como programadoras, podemos *especificar* las tareas escritas en una función.

Veamos la función `showImage()` del archivo `script.js`.

```
1~ /* If you're feeling fancy you can add interactivity
2   to your site with Javascript */
3
4 //This is an array that will hold the file names! Add at least 4 images to the images array.
5 var images = new Array();
6 images[0] = "https://cdn.glitch.com/5b718117-0dff-4b23-9a99-5b6ff3f4ee3d%2FNorthern%20Lights.jpg?v=1586357262455";
7 images[1] = "https://cdn.glitch.com/5b718117-0dff-4b23-9a99-5b6ff3f4ee3d%2FMount-Everest.jpg?v=1586357265642";
8 images[2] = "https://www.nationalgeographic.com/content/dam/travel/2019-digital/yosemite-guide/yosemite-national-park-california.jpg";
9
10 //This is an array that holds the names of each location shown in your pictures.
11 //Make sure that the index of each location is the same as the index of the picture in the images array
12 var locations = new Array();
13 locations[0] = "Northern Lights";
14 locations[1] = "Mount Everest";
15 locations[2] = "Yosemite National Park";
16
17 //This is your starting index. First we start off at Home!
18 var index = -1;
19
20
21 //This function will help us toggle through our photos. Everytime the button is pressed it will call this function!
22 function showImage() {
23
24 }
```

La **palabra clave** `function` le indica al programa que estás declarando una función. Esa palabra clave está seguida por el nombre de la función y los **parámetros**, elementos de información que deben pasarse a la función como entrada, que se indican entre los paréntesis `()`. Tal vez hayas notado que no hay nada entre los paréntesis de la función. Esto se debe a que los parámetros son opcionales. Todos los pasos que se realizarán al **invocar** la función están contenidos en un **bloque de código** entre las llaves `{}`, puntualizados por un `;` final.

Queremos que la función haga lo siguiente:

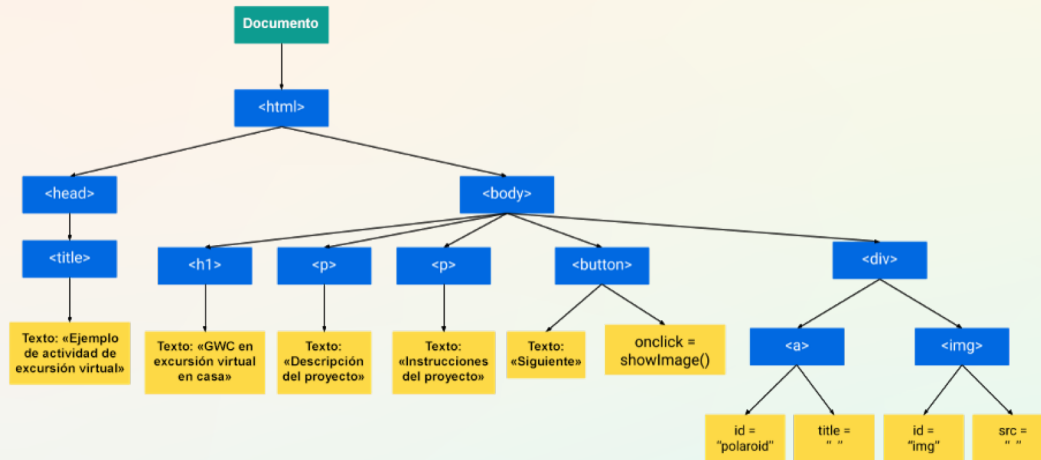
1. Actualizar la imagen en el polaroid
2. Actualizar el texto en el polaroid
3. Avanzar a la siguiente imagen

En los tres pasos siguientes, escribiremos código para añadir interacción a nuestra excursión virtual.

Duodécimo Paso: Actualiza la imagen en el polaroid (10 minutos)

Para actualizar la imagen en el polaroid, necesitamos usar el DOM. Un sitio web utiliza un **DOM** (modelo de objetos del documento) para organizar todo el contenido de la página web. El DOM refleja la estructura escrita en nuestro archivo HTML. Aquí está el modelo de objetos del documento (DOM) de nuestro sitio web hasta ahora.

```
<div class="polaroid">
  <a id="polaroid" href="#" title="">
    <img id="img" src="" />
  </a>
</div>
```



Notarás que los cuadros **azules** representan las etiquetas en el archivo HTML. La información representada en los cuadros **amarillos** son **atributos** para cada tipo de elemento. Los atributos son información única para cada elemento. Así como tu color de cabello, color de ojos y nombre son atributos tuyos. Para que nuestro archivo de JavaScript se comunique con el esqueleto HTML, usamos el DOM para conectar los elementos de los archivos `script.js` e `index.html`. Podría decirse que el DOM es un mapa para que nuestro código de JavaScript localice los elementos que queremos cambiar. El uso de la **identificación (id)** de un elemento es la manera más sencilla de invocar un elemento en tu archivo HTML. Observa que al elemento que hace referencia a la fotografía polaroid se le asignó el nombre de identificación (id) **polaroid** y que la imagen dentro de la fotografía polaroid tiene una etiqueta `` y un nombre de identificación **img**.

Nota: Las siguientes líneas de código deben escribirse *dentro* de la función `showImage()`. Todo el código debe escribirse entre las llaves `{}`.

1. Usa el DOM para crear una referencia a la etiqueta de imagen con la identificación "img".

Necesitamos indicarle a nuestro archivo de JavaScript qué elemento del archivo HTML queremos cambiar. Es aquí donde se verdaderamente luce el DOM.

```
var img = document.getElementById("img");
```

Usamos el objeto `document` para indicarle a JavaScript que queremos realizar una acción en *nuestra* página web. Se usa el método `getElementById()` para encontrar un elemento en nuestra página web mediante la identificación de una etiqueta. Usamos la identificación de etiqueta de imagen `"img"` para hacer referencia a la etiqueta ``. Por último, guardamos la referencia a la etiqueta de la imagen en una variable a la que le asignamos el nombre `img`.

Duodécimo Paso: Actualiza la imagen en el polaroid (continuación)

2. **Actualiza la fuente de una de las imágenes en el arreglo `images`.** Ya creamos una variable llamada `index` que comienza en 0. Esta variable hará un seguimiento de la imagen que quieres mostrar.

```
img.src = images[index];
```

Tenemos nuestro objeto `img`, que encontramos a través del DOM. Ahora queremos adjuntarle un atributo de fuente, usando la palabra clave `src`. Aquí, usamos `images[index]` para obtener la imagen con el valor de índice guardado en la variable `index`. Usamos el signo de igualdad `=` para asignar la fuente de la imagen a la nueva imagen.

```
//This function will help us toggle through our p  
function showImage() {  
  var img = document.getElementById("img");  
  img.src = images[index];  
}
```

3. Prueba que el código funcione correctamente. En tu sitio web en vivo, presiona el botón "Next" (Siguiente). Debe aparecer la primera imagen. De lo contrario, revisa que haya puntos y comas al final de cada línea de código de la función y que las imágenes tengan la dirección URL correcta.

Decimotercero Paso: Actualiza el texto en el polaroid (5 minutos)

Ahora, continuaremos programando *dentro* de la función `showImage()`, añadiendo código para actualizar el texto que se muestra con la imagen. Recuerda: todos estos pasos deben escribirse entre las llaves `{}`.

1. **Usa el DOM para crear una referencia a la etiqueta del polaroid, usando la identificación "polaroid".**

```
var polaroid = document.getElementById("polaroid");
```

Aquí, invocamos el DOM usando la palabra clave `document` y hacemos referencia a la etiqueta de la imagen por su identificación, "polaroid". Como estamos usando la identificación para obtener la etiqueta, utilizamos el método `getElementById()`. Por último, guardamos esto en una variable a la que le asignamos el nombre `polaroid`.

2. **Actualiza el título para una de las ubicaciones en el arreglo `locations`.** Ya creamos una variable llamada `index` que comienza en 0. Se usa para ayudarte a mantener seguimiento de cual descripción quieres mostrar.

```
polaroid.title = locations[index];
```

Aquí, usamos `locations[index]` para obtener la imagen usando el valor de índice guardado en la variable `index`. Queremos asegurarnos de usar el mismo índice para obtener el lugar correcto para cada imagen. Después, obtenemos el atributo de título de la imagen usando la propiedad `title`. Usamos el signo de igualdad `=` para asignar el texto al lugar correcto.

```
function showImage() {  
  var img = document.getElementById("img");  
  img.src = images[index];  
  
  var polaroid = document.getElementById("polaroid");  
  polaroid.title = locations[index];  
}
```

Decimocuarto Paso: Avanza a la siguiente imagen (2 minutos)

Cada vez que haces clic en el botón “Next” (Siguiente), debe aparecer una imagen. Sin embargo, en este momento solo aparece una de tus imágenes. ¿Por qué? Para obtener la imagen que queremos, usamos la variable `index` para obtener los datos guardados en los arreglos. Hasta ahora, a `index` se le asignó un valor de 0, pero nunca lo cambiamos.

Después de actualizar la imagen y el título, queremos incrementar el valor del índice por 1, para que avancemos por cada una de las fotos y de los lugares.

```
index = index + 1;
```

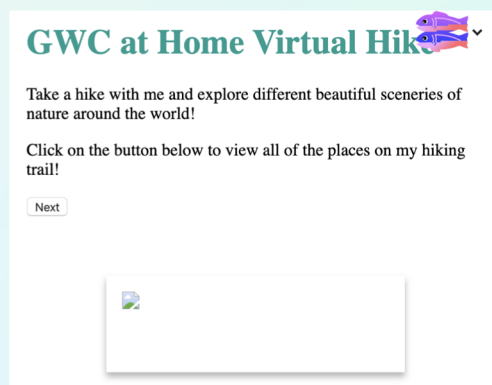
Aquí, reasignamos el índice para que sea igual al valor más 1. Queremos añadir esto como la *última* declaración de la función.

```
function showImage() {  
  var img = document.getElementById("img");  
  img.src = images[index];  
  
  var polaroid = document.getElementById("polaroid");  
  polaroid.title = locations[index];  
  index = index+1;  
}
```

Probemos rápidamente si funciona nuestro código. En el sitio web en vivo, presiona el botón “Next” (Siguiente). Cada vez que hagas clic en el botón “Next” (Siguiente), deberás ver la siguiente imagen de la presentación.

Decimoquinto Paso: ¡Ay caray! ¡Has ido demasiado lejos! (5 minutos)

Tal vez hayas notado que aparece un error si haces clic en “Next” (Siguiente) más allá de la última imagen.



Como incrementamos el valor de `índice` cada vez que hacemos clic en el botón “Next” (Siguiente), llegará el momento en el que el programa intente obtener datos inexistentes del arreglo. Por lo tanto, necesitamos una manera de determinar si el índice es demasiado grande. Una **declaración condicional** revisa si se cumple una serie de reglas (o una declaración) y decide qué acciones realizar si las reglas o la declaración son verdaderas o falsas. En nuestro caso, queremos saber si el `index` es demasiado grande; es decir, si es mayor que la longitud o el tamaño del arreglo.

Decimoquinto Paso: ¡ Ay caray! ¡Has ido demasiado lejos! (Continuación)

Antes de escribir la declaración condicional, debemos planear cuáles son los casos.

1. Si `index` no es demasiado grande (menor que la longitud del arreglo)
Podemos incrementar el índice por 1, como antes.
2. Si `index` es demasiado grande
Queremos conservar `index` con el mismo valor. Al no cambiar el valor de `índice`, dejamos la última imagen visible para los usuarios y no avanzamos más por las imágenes.

Sin embargo, ¿cómo determinamos si el `índice` es demasiado grande? Podemos determinarlo por el tamaño del arreglo. Piensa: ¿cuál es el último valor de índice en el arreglo? ¿Cómo se relaciona con el tamaño? De hecho, el último índice del arreglo siempre es uno menos que el tamaño o la longitud del arreglo. La razón por esto es que los arreglos tienen índice 0. Sabemos que el último índice puede representarse por `images.length - 1`.

Al escribir declaraciones condicionales, usamos las palabras clave `if` y `else` para dividir la serie de reglas y usamos el bloque de código `{}` para indicarle a la computadora cuales líneas de código debe ejecutar si la condición es verdadera.

Queremos reemplazar nuestra declaración anterior, `index=index+1`; por las siguientes líneas de código para determinar si el índice es demasiado grande.

```
function showImage() {  
  var img = document.getElementById("img");  
  img.src = images[index];  
  
  var polaroid = document.getElementById("polaroid");  
  polaroid.title = locations[index];  
  
  if (index < images.length - 1) {  
    index = index + 1;  
  }  
}
```

Para ver si el `índice` es demasiado grande, lo comparamos con la `longitud` del arreglo. Ya mencionamos que el último índice del arreglo es uno menos que la longitud, por lo que usamos este valor en nuestra condición (o regla). Sabemos que, si el índice no es demasiado grande, incrementaremos el valor del `índice` por 1. No añadimos más código para capturar el segundo caso, si el `índice` es demasiado grande, porque realmente no queremos actualizar el valor del `índice` en esta situación. En otras palabras, si el `índice` es demasiado grande, no hacemos nada.

Ahora, tómate un par de minutos para probar tu código. Asegúrate de **hacer clic** en el botón Next (Siguiente) varias veces para probar si el botón se detiene en la última imagen.

Decimosexto Paso: Extensiones (5 a 20 minutos)

Añadir fuentes bonitas (5 a 10 minutos)

En este proyecto, seleccionamos “Roboto” como fuente, pero puedes cambiarla para tu proyecto. Importar y usar fuentes divertidas en tu sitio web es muy fácil usando [Google Fonts](#). Ve este [video](#) para aprender cómo cambiar fuentes en tu sitio web.

Cambia el color de fondo de tu sitio web (5 a 10 minutos)

Cambiamos los colores de fondo por unos más interesantes. A los colores se les asignan valores numéricos para que la computadora pueda distinguirlos fácilmente. Pueden representarse por su valor hex, rgb o hsl. Recomendamos seleccionar un tipo para todos los colores para mantener consistencia. W3Schools tiene un excelente [selector de colores](#) que te ayudará a encontrar los valores de los colores que quieres añadir a tu sitio web. ¿Quieres añadir un fabuloso efecto degradado, como el del ejemplo de nuestro proyecto? Usa este [sitio web](#) para ayudarte a planear y generar el código CSS. Ve este [video](#) para obtener instrucciones detalladas para cambiar colores.

Añade más reglas de estilo CSS (10 a 20 minutos)

Con CSS, puedes hacer muchas cosas para que tu sitio web sea único. Hemos recopilado una lista de recursos que puedes explorar para conocer más consejos de estilo con CSS para tu sitio web.

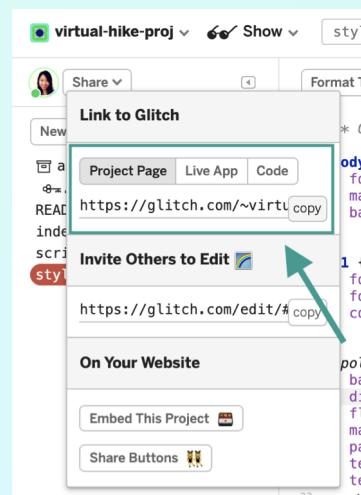
- W3Schools tiene una fantástica sección de [referencias](#) para CSS.
- Mozilla tiene mucha información útil en su módulo CSS, entre otras cosas:
 - [Cómo dar estilo al texto](#)
 - [Diseño CSS](#)
 - [Cómo usar CSS para resolver problemas comunes](#)
 - [Depuración de tu código CSS](#)

Decimoctavo Paso: Comparte tu proyecto de Girls Who Code en casa (5 minutos)

Nos encantaría ver tus proyectos de excursión virtual. Se te olvide etiquetar @girlswhocode #codefromhome e incluso podrías ser destacada en nuestra cuenta.

Para compartir tu proyecto de Glitch:

1. Haz clic en el botón **Share** (Compartir) en la parte superior izquierda, junto al icono de perfil.
2. Puedes optar por compartir el enlace de tu página de proyecto (lo que permitirá que otros vean tu código) o solo la aplicación en vivo. Haz clic en la opción que prefieras para compartir y copia el enlace provisto.



Hoja de trabajo de planificación para el proyecto de excursión virtual

Planificación general del proyecto

Tema: ¿Qué tema abarcará tu excursión virtual?

Audiencia: ¿A quién servirá este sitio web? ¿Qué grupo de personas estarían interesadas en tu producto?

Objetivo: ¿Qué quieres que logre tu sitio web? ¿Por qué le interesa esto a tu audiencia?

Planificación de imágenes

Selecciona por lo menos tres imágenes que quieras destacar en tu excursión virtual. Asegúrate de considerar el **orden** de las imágenes. Completa la siguiente tabla e incluye la imagen, la dirección URL de la imagen y el título.

Número	Imagen	Fuente de la imagen <i>(Indica si será una fotografía propia o de otro sitio web. Si usarás una fotografía de un sitio web, documenta un enlace al sitio).</i>	Título <i>(¿Qué texto quieres mostrarle a la audiencia con esta imagen?)</i>
0.			
1.			
2.			
3.			

Planificación de imágenes (continuación)

Número	Imagen	Fuente de la imagen <i>(Indica si será una fotografía propia o de otro sitio web. Si usarás una fotografía de un sitio web, documenta un enlace al sitio).</i>	Título <i>(¿Qué texto quieres mostrarle a la audiencia con esta imagen?)</i>
4.			
5.			
6.			
7.			