



Girls Who Code At Home

Passeggiata virtuale
App Web interattiva con Javascript

Panoramica delle attività

In occasione della commemorazione della giornata della Terra, questa settimana, vogliamo invitarti a pensare a un luogo selvaggio o all'aria aperta, che vorresti esplorare. Dove vorresti andare? Che cosa vorresti fare e vedere? In questo tutorial imparerai come usare gli **array**, o le liste su JavaScript per creare una passeggiata virtuale! Creerai un sito web che porterà le persone in giro, per esempio in un parco naturale, nel tuo museo preferito o le condurrà in un altro Paese! Le possibilità sono infinite! Prima di cominciare con il tuo itinerario ti consigliamo di dare un'occhiata alla Woman in Tech Spotlight del momento, Gitanjali Rao.

Nota: Quando si sviluppa un sito web, molti programmatori utilizzano l'HTML per costruire lo scheletro del sito web, il CSS per creare lo stile della pagina, e Javascript per aggiungere interattività. Durante questa attività lavoreremo principalmente su Javascript. Se hai bisogno di rinfrescare le tue conoscenze su HTML e CSS, ti consigliamo caldamente di dare un'occhiata alla nostra terza attività: [Condividi le tue competenze](#) prima di cominciare questa attività.

Materiali

- [Glitch](#) o editor di testo che preferisci
- [Esempio del progetto passeggiata virtuale \(con estensioni\)](#)
- [Esempio del progetto passeggiata virtuale \(senza estensioni\)](#)
- [Codice di avvio passeggiata virtuale](#)
- Facoltativo: Guida alla pianificazione
- Facoltativo: Penna/matita/pennarelli

Women in Tech Spotlight: Gitanjali Rao



Fonte dell'immagine: [BBC](#)

Nel 2014, a Flint, in Michigan, c'è stata una crisi della sanità pubblica. Del piombo, che è pericolosissimo se ingerito, soprattutto da bambini piccoli, è stato ritrovato nella rete idrica di Flint. Dopo aver appreso la notizia di questa crisi sanitaria, Gitanjali ha deciso di fare qualcosa e ha creato Tethys, un dispositivo che permette di rivelare se l'acqua è contaminata.

All'epoca, quando creò Tethys, Gitanjali aveva solo 11 anni. Non si è fermata, e nel 2017 ha vinto la gara [Discovery Education 3M Young Scientist Challenge](#) ed è stata invitata come relatore a TEDx!

Il suo dispositivo utilizza un nano tubo di carbonio per rilevare il flusso di elettroni in piombo, utilizzando un segnale basato su Arduino che lo collega a uno smartphone, tramite un'app Bluetooth.

Guarda il [video](#) su Gitanjali Rao e sulla sua invenzione, Tethys. Se hai del tempo, ti consigliamo di scoprire di più su Gitanjali dando un'occhiata a [questo articolo](#) di Bustle, [questo articolo](#) di Young Scientist Lab o [questo articolo](#) apparso su NPR. Se vuoi, puoi ritrovarla anche nel discorso da lei tenuto in occasione del TEDx [qui](#).

Riflettere

Essere un informatico è molto più che essere semplicemente bravo nella programmazione. Prenditi del tempo per riflettere su come Gitanjali e il suo lavoro hanno a che fare con i punti di forza su cui i grandi informatici puntano: coraggio, resilienza, creatività e motivazione.



SCOPO

Gitanjali è particolarmente interessata al campo medico, ma nonostante tutto ha imparato a fare coding e a incorporare la tecnologia. A quali campi in cui potresti utilizzare le tue conoscenze informatiche sei interessata?

Condividi le tue risposte con un familiare o un amico. Incoraggia gli altri a ottenere maggiori informazioni su Gitanjali per partecipare alla discussione!

Fase 1: Esplora (5 min)

In questo tutorial creerai un sito web dove vi saranno un serie di foto che permetteranno all'utente finale di immergersi in una passeggiata virtuale. Prenditi 5 minuti per dare un'occhiata ed esplorare alcune delle caratteristiche di questo esempio di [sito web](#) che abbiamo creato. Si tratta di un sito che parla di spazi naturali in Giappone!

- **Tematica del progetto:** Natura
- **Pubblico:** Persone interessate a imparare di più su luoghi naturali in Giappone
- **Obiettivo:** Condividere un'esperienza al di fuori della propria abitazione e progettare un itinerario concentrato sulla natura in Giappone

Mentre esplorerai il sito web pensa alle domande seguenti:

- Che cosa accade quando clicchi su ciascun pulsante?
- Le foto che sono state scelte si ricollegano alla tematica e all'obiettivo? Che genere di foto sceglieresti di integrare al sito?
- Che cosa ti aspetti che ci sia quando fai clic sul pulsante "Avanti"? Cerca di suddividere i vari passaggi in passaggi più brevi.
- Quando non funzionerà più il pulsante "Avanti"? Quando non funzionerà più il pulsante "Indietro"? Perché potrebbe succedere?

Fase 2: Fai un brainstorming sulle caratteristiche del tuo progetto e pianificalo (10 min)

Adesso che hai avuto la possibilità di visionare un esempio di progetto, potrebbe essere una buona idea prenderti del tempo per creare innanzitutto un piano d'azione. Utilizza questo tempo per cercare di capire che cosa vuoi ottenere con il tuo progetto e quale sarà il tuo obiettivo. Perché non utilizzi i documenti sulla pianificazione che si trovano in fondo a questa attività per raccogliere e organizzare le idee?

1. Scegli una tematica, un pubblico e l'obiettivo della tua passeggiata virtuale.

In occasione della commemorazione della giornata della Terra (22 aprile 2020), vogliamo invitarti a pensare a un luogo che vorresti esplorare, ma se vuoi puoi scegliere qualsiasi argomento tu preferisca. Se ti senti bloccata, qui di seguito troverai alcune idee alternative per il tuo progetto:

- Una galleria virtuale di attività/hobby che vorresti fare a casa.
- Una passeggiata virtuale nel tuo museo preferito.
- Una passeggiata virtuale nei tuoi posti del cuore.

Quando scegli un pubblico per la tua passeggiata virtuale pensa a quale potrebbe essere il tuo pubblico target. Chi potrebbe essere contento di usare il tuo sito web? Pensa a quali sono i loro interessi, che cosa vorrebbero sapere e in che modo potrai catturare la loro attenzione.

2. Raccogli almeno tre immagini.

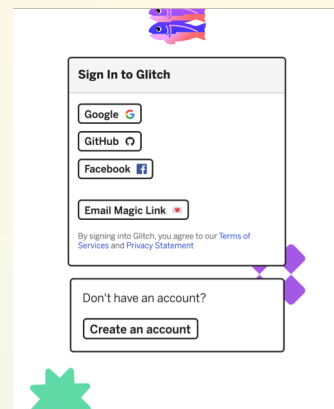
Ti ci vorranno almeno tre immagini che mostrerai durante il tuo viaggio virtuale. Puoi utilizzare tue immagini o altrimenti immagini che troverai online. Mentre recupererai le tue immagini, rifletti anche al titolo o al titolo pubblicitario che vorrai apporre ad ogni immagine.

Fase 3: Primi passi su Glitch (10 min)

Glitch è uno strumento semplice per la creazione di app web. Viene fornito con un editor di testo che ti consente di visualizzare le modifiche alla tua pagina web in tempo reale. Ti permette inoltre di pubblicare facilmente il tuo progetto per farlo vedere al mondo! Se trovi un bel progetto realizzato da qualcun altro, puoi guardarne il codice e remixarlo.

1. **Accedi a [Glitch](#) utilizzando il tuo account Google, Facebook o GitHub.**
 - Per salvare e condividere il tuo lavoro su Glitch, devi effettuare l'accesso. Glitch ti consente di accedere utilizzando un account Google, Facebook o GitHub. Se hai meno di 13 anni avrai bisogno del permesso e dell'indirizzo e-mail del tuo tutore legale per registrarti.
2. **Remixa questo [codice d'avvio](#) per il sito web della tua passeggiata virtuale.** Guarda questo [video](#) per ricevere delle istruzioni più dettagliate.
3. **Esplora l'interfaccia di Glitch.**

La prima cosa che vedrai sono i contenuti del file [README.md](#). I file README sono documenti che vedrai costantemente nei progetti di altre persone. Di solito contengono informazioni su come navigare tra i file del progetto e su come eseguire il programma. Guarda questo [video](#) per imparare di più sull'interfaccia di Glitch.
4. **Imposta la modalità di visualizzazione tuo progetto.** Fai clic sul pulsante **Mostra** e scegli l'opzione **Accanto al codice**. Questo ti permetterà di visualizzare il tuo codice e di vedere come appare il progetto dall'altra parte del web. Guarda questo [video](#) per ricevere delle istruzioni più dettagliate.



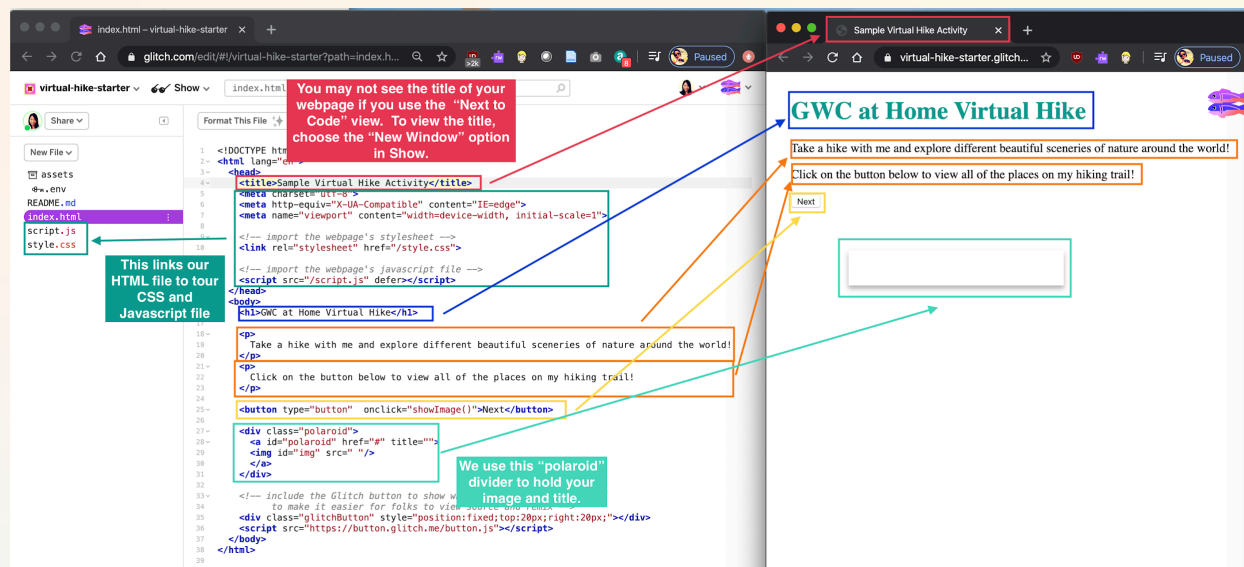
Fase 4: Scopri il Progetto HTML iniziale (10 min)

Innanzitutto, dai un'occhiata al file [index.html](#). Fai clic sul file [index.html](#) dal menù di navigazione a sinistra. **HTML**, acronimo per Hypertext Markup Language, utilizza **tag** per strutturare i contenuti di un sito web. Tutti i contenuti mostrati nell'anteprima dal vivo sono inclusi all'interno di un `<body>` tag di apertura e un `</body>` tag di chiusura. I tag fanno parte dell'anatomia di un elemento HTML, e di solito sono costituiti dal tag di apertura, dai contenuti e dal tag di chiusura. Dai un'occhiata alle immagini qui sotto per vedere alcuni esempi di tag HTML per diversi tipi di elementi. Il tag `<p>` viene utilizzato per indicare il "paragrafo", mentre il tag `<a>` è piuttosto un tag più generico, che viene utilizzato per i link.



Fase 4: Scopri il Progetto HTML iniziale (continua)

L'immagine qui sotto mostra i tag che vedi sul tuo codice d'avvio e come i tag appaiono direttamente sul tuo sito web.



Adesso, prendiamo un po' di tempo per personalizzare le informazioni per il *tuo* sito web!

Provaci tu a fare coding in HTML!

- Cambia il titolo del tuo sito web.** Il titolo del tuo sito corrisponde al nome che appare quando carichi il tuo sito web. Nota che questo non equivale a cambiare l'URL del tuo sito web. Se vuoi verificare che il titolo del tuo sito web sia corretto, fai clic su **Mostra** e seleziona l'opzione **Nuova finestra** dal menu.
 - Localizza il tag del `<titolo>` nel file HTML. (Questo è sottolineato di rosso nella casella dell'immagine qui sopra).
 - Sostituisci il testo fra il tag di apertura `<titolo>` e il tag di chiusura del `</title>` inserendo il nome del tuo sito web.
Es. `<titolo> Nome della Mia nuova pagina web </titolo>`
- Cambia l'intestazione del tuo sito web.** L'intestazione è un testo che normalmente appare in cima alla tua pagina. Diversamente dal titolo della tua pagina web, l'intestazione sarà visibile sulla pagina!
 - Individua il tag `<h1>` nel file HTML. (Questo è sottolineato in blu nella casella dell'immagine qui sopra).
 - Sostituisci il testo fra il tag di apertura `<h1>` e il tag di chiusura `</h1>` inserendo il nome del tuo progetto. Potrebbe corrispondere allo stesso di quello che hai scritto nel titolo della tua pagina web.
Es. `<h1> Nome della mia nuova pagina web </h1>`

Fase 4: Scopri il Progetto HTML iniziale (Continua)

3. **Cambiare il testo sul tuo sito web.** In questo esempio di codice ci sono due paragrafi. Il primo include una breve descrizione del progetto, mentre il secondo fornisce all'utente le istruzioni su come interagire con il sito web.

- Individua il tag `<p>` nel file HTML (è sottolineato in arancione nella casella dell'immagine sottostante).
- Sostituisci il testo fra **la prima serie** di tag di apertura `<p>` e di tag di chiusura `</p>` con una breve descrizione del tuo progetto. Torna indietro e dai un'occhiata a che cosa avevi scritto sul tuo documento sulla pianificazione.

Ex. `<p>`

La mia nuova pagina web Descrizione del testo

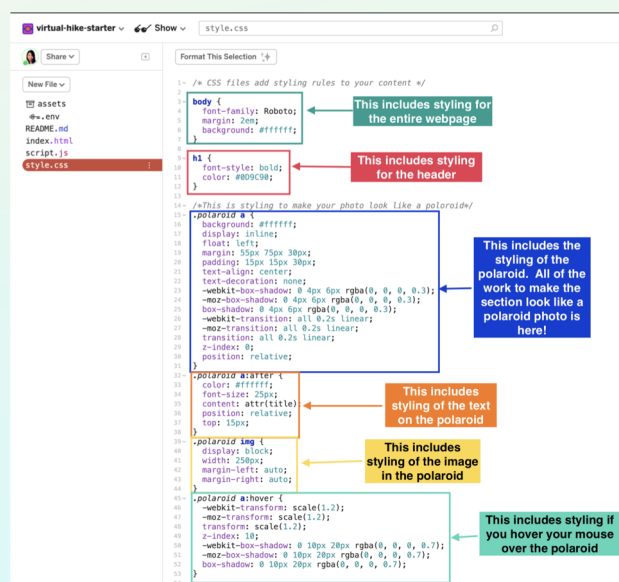
`</p>`

- **(Facoltativo)** Personalizza le istruzioni nella seconda serie di `<p>` tag.

Fase 5: Scopri il Progetto CSS iniziale (5 min)

Il **CSS**, acronimo per **Cascading Style Sheets**, descrive quali tipi di regole di presentazione, o stili, devono essere applicati agli elementi HTML. CSS ti consente di cambiare l'aspetto dei contenuti del tuo sito web, compreso il colore del testo, la dimensione del testo, i caratteri utilizzati, i colori o le immagini di sfondo e molto altro. In questo tutorial non andremo troppo nei dettagli del CSS. Ti invitiamo a guardare [questa](#) risorsa o di provare la nostra attività [Condividi le tue competenze](#) se ti interessa saperne di più sul CSS!

In questo progetto iniziale abbiamo già visto diverse regole di base dello stile CSS, in modo da poterci concentrare sulla personalizzazione e sull'apprendimento di JavaScript. Abbiamo inoltre incluso lo styling per il tuo titolo, la tua intestazione e la polaroid. Abbiamo chiamato uno di questi elementi **polaroid** perché abbiamo aggiunto delle regole di styling su CSS per rendere le tue foto simili a delle foto scattate con una polaroid! Gli stili per la polaroid sono stati presi in prestito da [questo](#) tutorial. Inoltre, avrai del tempo alla fine di questo progetto per personalizzare il tuo sito web, utilizzando i colori e i caratteri che più ti piacciono!



Fonte dell'immagine: [Creative Market](#)

Fase 6: Introduzione a Javascript (2 min)

Siamo finalmente arrivati alle cose divertenti! Fino ad ora abbiamo costruito lo scheletro del tuo sito web, ma questo sito non è ancora in grado di fare nulla! Ecco dove entra in gioco **Javascript**. Javascript viene utilizzato per rendere i siti web interattivi. Diamo un'occhiata al file `script.js`. Abbiamo aggiunto alcune cose per iniziare, così come alcuni commenti per spiegare esattamente di cosa si occupa ogni stringa del codice.

```
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:** I commenti non sono letti dal computer: descrivono il codice per altri o per il sé futuro. Puoi aggiungere un commento su una stringa, semplicemente scrivendo `//` seguito da un messaggio.*

Fase 7: Introduzione sulle Variabili e sugli Array (2 min)

Per poter immagazzinare informazioni (per esempio le tue foto e le descrizioni) abbiamo bisogno di utilizzare una **variabile**. Una variabile è un contenitore che immagazzina qualsiasi genere di dato. Può contenere parole, numeri e addirittura una lista di dati. Diamo un'occhiata ai simboli chiave (o sintassi) per l'utilizzo delle variabili:

`var myVariable = valore;`

- **var** : Questa parola chiave viene utilizzata per stabilire di che variabile si tratta.
- **myVariable** : Questo è il nome della nostra variabile. I nomi di solito seguono [la notazione a cammello](#) senza che vi siano degli spazi o delle lettere maiuscole a indicare l'inizio di una nuova parola.
- **=** : Il simbolo dell'uguale rappresenta l'assegnazione o la riassegnazione di un valore a una variabile
- **valore** : Può trattarsi di qualsiasi valore - numero, parola o addirittura di una lista di dati.
- **;** : Tutte le istruzioni di Javascript devono terminare con un punto e virgola. In questo modo il computer riconosce che il comando è concluso.

Un **array** corrisponde a una struttura ordinata di dati che possono contenere diversi tipi di informazioni. Puoi pensare a un array come se fosse un cassetto con molti cassetti, dove ogni cassetto contiene dei dati. Vediamo adesso di suddividere i simboli chiave (o sintassi) utilizzati per creare un array **vuoto** (come nel codice di avvio):

`var mioArray = nuovo Array();`

- **var** : Utilizziamo la parola chiave **var** per immagazzinare l'array in una variabile.
- **nuovo** : Questa parola chiave è utilizzata per mostrare che vogliamo creare un oggetto vuoto. Nel nostro caso, per creare un array vuoto!
- **Array** : Questa parola chiave indica al computer di creare un oggetto dell'array.
- **()** : Le parentesi vengono utilizzate per chiamare l'array **funzione**. Javascript include già una funzione integrata (una serie di comandi) che aiutano a creare l'oggetto dell'array. Impareremo di più sulle funzioni e su come vengono usate, più avanti in questo progetto.

Fase 8: Come utilizzare gli Array (10 min)

Gli array sono fra le strutture di dati più largamente utilizzate perché rappresentano una struttura *ordinata* di dati. Un array assegna un numero a ogni elemento che immagazzinano. Questo numero viene chiamato indice. Il fatto di assegnare a ogni elemento un indice rende l'accesso, l'eliminazione o la sostituzione dei valori immagazzinati nell'array, un compito molto semplice. In informatica, l'indice comincia a partire da 0! Al primo elemento presente nell'array verrà assegnato l'indice 0. Tale indice incrementerà di 1 per ogni elemento successivo. Diamo un'occhiata a questo esempio:

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

In questo esempio, l'array è stato chiamato `programmatori`. Contiene i nomi di famose programmatrici: "Ada", "Grace", "Katherine", "Roya". In totale questo array è lungo 4.

Le parentesi quadre `[]` sono utilizzate per indicare l'inizio e la fine dei contenuti di un array.

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

Mettiamo che volessimo accedere al nome "Grace" dall'array. Come dovremmo fare?

Potremmo semplicemente utilizzare l'indice associato a "Grace" per ottenere questa informazione. Dovremmo semplicemente scrivere l'istruzione: `programmatori[1]`. Questo ci permetterà di accedere al nome di "Grace". Innanzitutto (1) scriveremo il *nome* dell'array al quale vogliamo accedere, poi (2) aggiungeremo delle parentesi quadre `[]`, poi (3) indicheremo il numero dell'indice al quale vogliamo accedere, all'interno delle parentesi quadre. Poiché vogliamo accedere a "Grace" metteremo il numero 1!

Se avessimo scritto `programmatori[3]`, ci avrebbe portato a "Roya". Se avessimo scritto `programmatori[4]` avremmo ricevuto un messaggio di **errore**! Gli Array **sono indicizzati a partire da 0**. Ciò significa che l'indicizzazione parte da 0. Anche se ci sono 4 nomi sulla lista, il nome che si trova alla fine della stessa ha un indice che *è uno in meno rispetto alla lunghezza*.

Possiamo inoltre utilizzare l'indice per aggiungere o per modificare il valore dell'array. Mettiamo il caso che volessimo aggiungere il nome "Gitanjali" al nostro array. Possiamo scrivere la seguente sintassi:

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


Fase 9: Aggiungere la tua prima immagine e posizione (10 min)

Dai di nuovo un'occhiata al file `script.js`. Si noti che utilizziamo la sintassi indicata in precedenza per creare due array diversi, uno che verrà chiamato `immagini` e l'altro `posizioni`.

```
/* 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() {
}
```

- `immagini` corrisponderà a un array che contiene gli indirizzi delle immagini, o i link delle immagini di tutte le foto che vorrai includere.
- `posizioni` corrisponderà a un array che contiene le posizioni geografiche di ogni immagine (o descrizione se decidi di scegliere una tematica diversa).

A poco a poco che aggiungiamo foto e titolo agli array `immagini` e `posizioni` dobbiamo assicurarci che il valore dell'indice sia allineato con entrambi gli array. Ciò significa che se posiziono un'immagine delle Luci del Nord sull'array `immagini` all'indice 0, voglio anche includere il testo del titolo "Luci del Nord" sull'array delle `posizioni` a indice 0.

Per includere una foto nell'array, metteremo piuttosto l'indirizzo URL dell'immagine.

- Se utilizzerai delle immagini online, ti consigliamo di salvare l'indirizzo URL dell'immagine sulla tua guida alla pianificazione. Per ottenere l'indirizzo di un'immagine guarda questo [video](#).
- Se stai utilizzando immagini tue, assicurati che siano salvate *in versione digitale* sul tuo computer. Per caricare le tue immagini su Glitch, guarda questo [video](#).

Adesso che hai ottenuto gli indirizzi URL delle tue immagini, aggiungile al tuo array `immagini`!

Torna indietro al file `script.js`. Poiché si tratta della tua prima immagine, aggiungiamo l'URL dell'immagine con indice 0. Copia l'indirizzo URL della tua immagine fra le "" sulla stringa numero 6 del tuo codice di avvio.

```
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
```

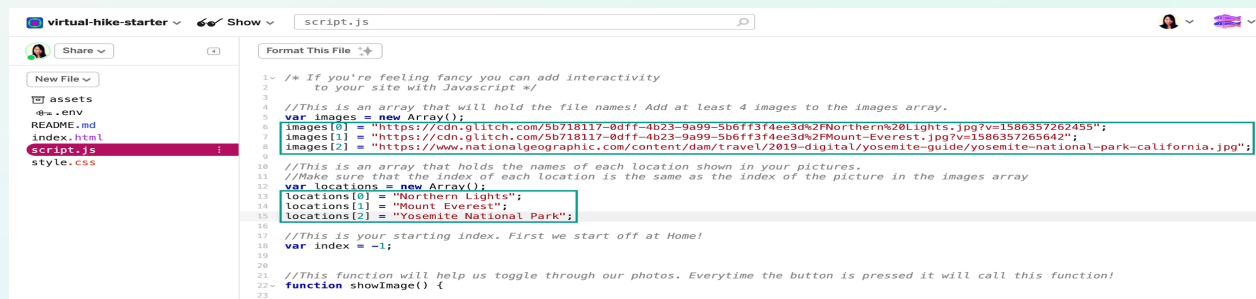
Fase 9: Aggiungere la tua prima immagine e posizione (Continua)

Adesso, vogliamo aggiungere la posizione, o il titolo dell'immagine che hai appena caricato. Aggiungi il titolo dell'immagine che hai aggiunto, tra le doppie virgolette "" all'indice 0 dell'array **posizioni**.

```
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/5b718117-0dffa4b23-9a99-5b6ff3f4ee3d%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
```

Fase 10: Aggiungere più immagini e posizioni (10 min)

Adesso che hai aggiunto le prime due immagini e il titolo della posizione, è arrivato il momento di aggiungere le foto rimanenti! Segui le istruzioni presentate alla fase 9 per aggiungere più foto e testo. Ricordati che, avendo aggiunto l'indirizzo URL dell'immagine all'array di **immagini**, devi aggiungere il titolo corrispondente all'array **posizioni** *allo stesso indice*. Il tuo codice dovrebbe essere simile a quello che appare nell'immagine qui sotto dopo che avrai aggiunto tutte le tue immagini e tutte le tue posizioni.



Adesso hai aggiunto tutte le immagini ma non succede nulla quando clicchi sul pulsante "Avanti"! Anche se adesso abbiamo un modo per immagazzinare queste immagini, non abbiamo ancora impostato un comando che ci permetta di accedervi! Cambiamolo subito.

Fase 11: Introduzione alle funzioni (5 min)

Innanzitutto diamo un'occhiata al pulsante del tag. Vai al file [index.html](#) e cerca questa stringa di codice.

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

Nota che l'attributo `onclick` corrisponde all'attributo `mostralmimmagine()`. L'attributo `onclick` è utilizzato per indicare al computer cosa deve fare quando si clicca sul pulsante. In questo caso, il nostro pulsante dovrebbe chiamare o eseguire la **funzione** `mostralmimmagine()`. La funzione `mostralmimmagine()` è scritta sul nostro file Javascript. Ma prima di inoltrarci nella spiegazione di `mostralmimmagine()`, spieghiamo innanzitutto che cosa sia una funzione!

Una **funzione** è una sezione del codice che svolge una serie di istruzioni (o stringhe di codice). Per esempio, se ti viene chiesto di "apparecchiare", sai già in partenza che ciò significa che dovrai apparecchiare la tavola per un certo numero di persone che sono a casa tua, dovrai mettere le posate (forchette, cucchiari, coltelli e/o bacchette) per ogni posto, e poi dovrai aggiungervi un tovagliolo. **Apparecchiare** corrisponde a una **funzione** e l'attività che deve essere svolta dall'individuo per completare l'apparecchiatura corrisponde a ciò che è incluso all'interno della funzione. Va da sé, che le istruzioni che vengono seguite per apparecchiare a casa tua possono essere diverse. In quanto programmatori possiamo *specificare* i compiti scritti dentro la funzione.

Diamo ora un'occhiata alla funzione `mostralmimmagine()` che si trova nel file `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 }
```

Con l'utilizzo della **parola chiave** `funzione`, indichi al programma che stai dichiarando una funzione. Quella parola chiave è seguita da un nome di una funzione e da qualsiasi **parametro** – o informazioni che richiedono di essere trasferite alla funzione come input – elencate tra `()`. Avrai sicuramente notato che non c'è niente tra le parentesi della nostra funzione. Questo perché i parametri sono facoltativi! Tutte le fasi che devono essere svolte quando una funzione è **chiamata** sono contenute in un **blocco di codice** tra le `{ }` che viene concluso con un `;`.

Vogliamo che la funzione si occupi delle cose seguenti:

1. Aggiornare l'immagine nella polaroid
2. Aggiornare il testo nella polaroid
3. Procedere all'immagine successiva

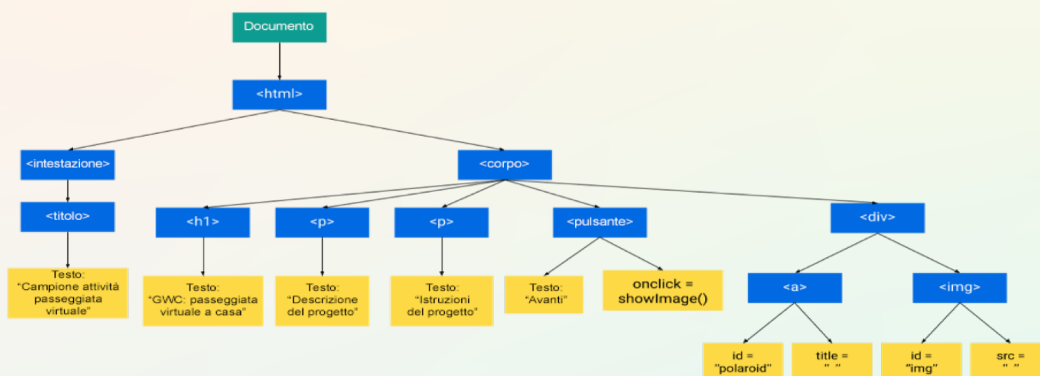
Nelle prossime tre fasi scriveremo un codice per rendere interattiva la nostra passeggiata virtuale.

Fase 12: Aggiornare l'immagine nella polaroid (10 min)

Per poter aggiornare l'immagine nella polaroid, dovremo usare il DOM! Un sito internet utilizza un **DOM** (Document Object Model) per organizzare tutti i contenuti della pagina web. Il DOM riflette i

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

della Polaroid scritti nel nostro file HTML! Questo è stato fino ad ora il DOM del tuo sito web!



Avrai notato che le caselle **blu** rappresentano i tag presenti nel file HTML. Le informazioni presenti nelle caselle **gialle** corrispondono agli **attributi** per ogni genere di elemento. Gli attributi sono informazioni uniche a ogni elemento. Proprio come il colore dei nostri capelli, il colore dei nostri occhi e il nostro nome sono per noi degli attributi! Per permettere al nostro file Javascript di comunicare con lo scheletro HTML, utilizziamo il DOM per aiutare a collegare gli elementi sui file [script.js](#) e [index.html](#). Si potrebbe affermare che il DOM preveda una mappa per il nostro codice JavaScript per localizzare gli elementi che vogliamo modificare. Utilizzare l'**id** di un elemento rappresenta il modo più semplice per cercare un elemento nel tuo file HTML. Non ti dimenticare che all'elemento che si riferisce alla foto polaroid era stato dato l'id **polaroid** e che l'immagine all'interno della foto polaroid aveva come tag **** e come id nome **img**.

Nota: Le prossime stringhe di codice dovrebbero essere scritte *all'interno* della funzione `mostrImmagine()`. Tutti i codici dovrebbero essere scritti all'interno delle `{}`.

1. **Utilizza il DOM per creare una referenza al tag dell'immagine avente come id l'id seguente: "img".** Dobbiamo dire al nostro file JavaScript quali elementi vogliamo modificare nel file HTML. ed è qui che entra in gioco il DOM.

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

Utilizziamo l'oggetto del `documento` per ordinare a JavaScript che vogliamo fare qualcosa sul *nostro* sito web. Il metodo `getElementById()` è utilizzato per cercare un elemento sulla nostra pagina web secondo un determinato tag id. Utilizziamo il tag id **"img"** per fare riferimento al tag ****. Per concludere, registriamo la referenza del tag dell'immagine in una variabile che chiameremo **img**.

Fase 12: Aggiornare l'immagine nella polaroid (Continua)

2. **Aggiornare la fonte dell'immagine a una delle foto presenti nell'array `immagini`.** Abbiamo già creato una variabile chiamata `indice` che inizia a partire da 0. Questo ci permetterà di verificare l'immagine che vorrai mostrare!

```
img.src = immagini[indice];
```

Abbiamo l'oggetto della nostra `img` che abbiamo ritrovato attraverso il DOM. Adesso vogliamo allegarvi una fonte di attributo utilizzando la parola chiave `src`. In questo caso, utilizziamo `immagini[indice]` per trovare la foto che vogliamo usare, utilizzando l'indice che è salvato sull'`indice` variabile. Utilizzeremo il simbolo uguale `=` per assegnare la fonte dell'immagine alla nuova foto!

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

3. Verifica che i tuoi codici funzionino! Clicca sul pulsante "avanti" della versione live del tuo sito web! La tua prima immagine dovrebbe apparire. Nel caso non dovesse apparire, verifica che alla fine di ogni stringa di codice, all'interno della funzione, ci sia il punto e virgola e che le tue immagini presentino l'indirizzo URL corretto.

Fase 13: Aggiornare il testo nella polaroid (5 min)

Adesso continueremo a fare coding *all'interno* della funzione `mostraImmagine()`, aggiungendo codici per aggiornare il testo mostrato con l'immagine. Ricordati che tutte queste fasi dovranno essere scritte all'interno delle `{}`.

1. **Crea una referencia utilizzando DOM per il tag polaroid, applicando l'id "`polaroid`".**

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

In questo modo richiamiamo il DOM utilizzando la parola chiave `document` e facendo riferimento al tag dell'immagine attraverso il suo id, "`polaroid`". Poiché stiamo utilizzando l'id per ottenere il tag, utilizziamo il metodo `getElementById()`. Per concludere, salviamo il tutto in una variabile che chiameremo `polaroid`.

2. **Aggiornare il titolo di una delle posizioni sull'array `posizioni`.** Abbiamo già creato una variabile chiamata `indice` che inizia a partire da 0. Questo viene utilizzato per aiutarti a controllare quale descrizione vuoi che appaia!

```
polaroid.title = posizioni[indice];
```

In questo caso, utilizziamo `posizioni[indice]` per trovare la foto che vogliamo usare, utilizzando l'indice che è salvato sull'`indice` variabile. Vogliamo assicurarci di utilizzare lo stesso indice per ottenere la posizione corretta per ogni foto. Successivamente, tramite il `titolo` della proprietà otterremo il titolo dell'attributo dell'immagine. Utilizziamo il simbolo `=` per assegnare il testo alla posizione corretta!

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

Fase 14: Procedere all'immagine successiva (2 min)

Ogni volta che fai clic sul pulsante “Avanti” dovrebbe apparire un'immagine, ma adesso te ne appare solo una. Perché? Per vedere l'immagine che vuoi, abbiamo usato l'**indice** variabile per ottenere i dati salvati nei nostri array. Fino ad ora il nostro **indice** è stato configurato come uguale a 0, ma non lo abbiamo mai cambiato!

Dopo che aggiorniamo un'immagine e un titolo, quello che dobbiamo fare è di aumentare il valore dell'indice di 1. Quindi dobbiamo analizzare una foto e una posizione dopo l'altra.

indice = indice + 1;

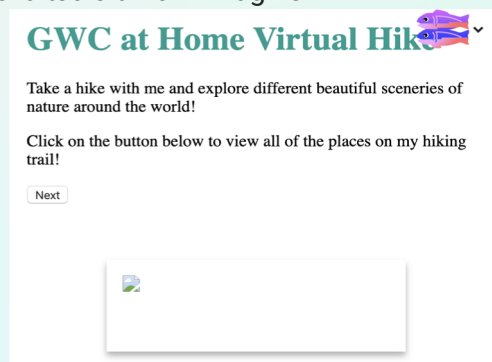
In questo caso, riassegniamo l'indice dei valori semplicemente al valore precedente più 1. Vogliamo aggiungerlo come *ultima* istruzione della funzione.

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

Facciamo un test per vedere se il nostro codice funziona! Sulla versione live del tuo sito web, fai clic sul pulsante “Avanti”. Ogni volta che farai clic sul pulsante “Avanti” dovresti vedere apparire l'immagine successiva della tua presentazione.

Fase 15: Perché no? Andare oltre! (5 min)

Avrai sicuramente notato che cliccando più volte sul pulsante “Avanti” riceverai un messaggio di errore dopo che avrai visualizzato la tua ultima immagine.



Poiché ogni volta che clicchiamo su “Avanti” aumentiamo il valore del nostro **indice**, è possibile che il nostro programma provi ad ottenere dei dati dal nostro array, che non esistono! Per questo, dobbiamo trovare qualcosa che ci permetta di raccogliere i dati nonostante il nostro indice sia troppo ampio. Un'istruzione **condizionale** verifica se una serie di regole (o di istruzioni) sono state rispettate e determina quali azioni debbano essere fatte nel caso in cui tali regole o istruzioni siano vere o false. Nel nostro caso, vogliamo sapere se l'**indice** è troppo ampio. In poche parole, vogliamo sapere se è più grande rispetto alla lunghezza/dimensione dell'array.

Fase 15: Perché no? Andare oltre! (Continua)

Prima di cominciare a scrivere un'istruzione condizionale vogliamo stabilire quali possono essere le casistiche.

1. Se l'**indice** non è troppo ampio (meno ampio rispetto alla lunghezza dell'array)
Allora possiamo aumentare il nostro indice di 1 come prima!
2. Se l'**indice** è troppo ampio
Allora in questo caso manteniamo il nostro **indice** tale e quale. Lasciando il valore dell'**indice** tale e quale, lasceremo l'ultima immagine visibile a tutti i nostri utenti e non avanza con le nostre immagini.

Ma in che modo possiamo determinare se un **indice** è troppo ampio? Possiamo affermarlo in base alle dimensioni di un array! Pensa di nuovo all'ultimo valore dell'indice dell'array. In che modo questo ti può aiutare a capirne la lunghezza? Di fatto, l'ultimo indice dell'array è sempre *meno* uno rispetto alle dimensioni/lunghezza dell'array. Questo avviene perché gli array sono indicizzati a 0! Sappiamo che l'ultimo indice può essere rappresentato dalla **lunghezza.immagini - 1**.

Quando si scrivono condizionali, utilizziamo le parole chiave **if** e **else** per spezzare le nostre serie di regole e per utilizzare il blocco di codice **{ }** per indicare al computer quali stringhe di codice devono essere eseguite se la condizione si rivela vera!

Vogliamo sostituire la nostra precedente istruzione **indice = indice + 1;** con le seguenti stringhe di codice per raccogliere quando l'indice è troppo largo!

```
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;  
  }  
}
```

Per verificare se l'**indice** è troppo ampio, dobbiamo paragonarlo alla **lunghezza** dell'array. Poiché abbiamo affermato che l'ultimo indice dell'array è sempre meno uno rispetto alla lunghezza, utilizziamo questo valore nella nostra istruzione condizionale (o regola). Sappiamo che se l'indice non è troppo ampio, allora incrementeremo il valore dell'**indice** di 1. Non aggiungeremo altro codice per recuperare il secondo caso se l'**indice** è troppo ampio perché non vogliamo veramente aggiornare il valore dell'**indice** se ci troviamo in questo caso. In altre parole, se l'**indice** è troppo ampio, allora non faremo niente!

16

Adesso, prenditi due minuti per fare una prova del tuo codice. Assicurati di fare **click** sul pulsante Avanti diverse volte per provare se il tuo pulsante si arresta all'ultima immagine!

Fase 16: Estensioni (5-20 min)

Aggiungere dei caratteri divertenti (5-10 min)

Per questo progetto abbiamo impostato il carattere "Roboto", ma se preferisci lo puoi modificare per il tuo progetto! Puoi facilmente importare e utilizzare caratteri fantasiosi nel tuo sito web da [Google Fonts](#). Guarda questo [video](#) per imparare a modificare il carattere del tuo sito web!

Cambiare il colore dello sfondo del tuo sito web (5-10 min)

Adesso cambieremo i colori dello sfondo con dei colori un po' più accattivanti! Dei valori numerici vengono attribuiti ai colori per facilitare il compito del computer nel distinguere i vari colori. Possono essere rappresentati con i loro valori hex, rgb, o hsl. Ti consigliamo di utilizzare lo stesso tipo per tutti i colori, per essere coerenti. W3Schools offre un'ottima [tavolozza di colori](#) per aiutarti a scegliere i valori dei colori che potresti volere aggiungere al tuo sito web! Vuoi aggiungere un bell'effetto sfumato come sul nostro progetto? Vai su questo [sito web](#) per aiutarti a progettare e generare il codice CSS. Guarda questo [video](#) per ottenere informazioni dettagliate su come cambiare i colori.

Aggiungere più regole di stile CSS (10-20 min)

Puoi fare così tanto con CSS per rendere il tuo sito unico! Abbiamo compilato una lista di risorse che potrebbero esserti utili per trovare ulteriori consigli su come impostare lo stile del tuo sito web con CSS.

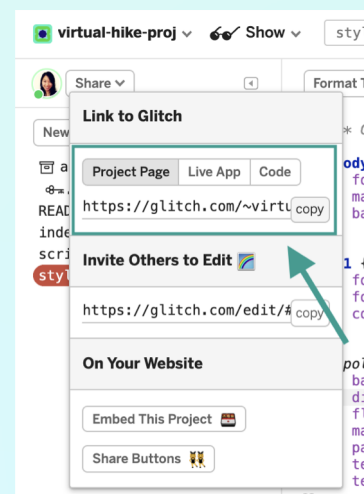
- W3Schools ha un'ottima sezione di [riferimento](#) per il CSS.
- Mozilla ha molte informazioni utili nel suo modulo CSS, tra cui:
 - [Come modificare lo stile del testo](#)
 - [Layout CSS](#)
 - [Come usare i CSS per risolvere problemi comuni](#)
 - [Debug del tuo CSS](#)

Fase 18: Condividi il tuo progetto Your Girls Who Code at Home! (5 min)

Ci farebbe piacere vedere i tuoi progetti su questa passeggiata virtuale! Ricordati di taggare @girlswhocode #codefromhome e potremmo anche nominarti sul nostro account!

Per condividere il tuo progetto Glitch:

1. Fai clic sul pulsante **Condividi** in alto a sinistra dello schermo, vicino all'icona del tuo profilo.
2. Puoi decidere di condividere il link della pagina del tuo Progetto (questo permetterà agli altri di vedere i tuoi codici) o se preferisci puoi condividere la tua Live App. Fai clic su qualsiasi opzione che desideri condividere e copia il link che apparirà.



Foglio di pianificazione del progetto passeggiata virtuale

Progettare panoramica pianificazione

Tematica: Di che argomento tratterai durante la tua passeggiata virtuale?

Pubblico: A chi servirà il tuo sito web? Che gruppo di persone sarà interessato al tuo chatbot?

Obiettivo: Che cosa vuoi ottenere con il tuo sito web? Perché è interessante per il tuo pubblico?

Pianificazione delle immagini

Scegli almeno tre immagini che utilizzerai durante la tua passeggiata virtuale. Assicurati di prendere in considerazione l'**ordine** delle tue immagini. Compila la tabella qui sotto e includi l'immagine, l'indirizzo URL dell'immagine e il titolo.

Numero	Immagine	Fonte dell'immagine <i>(Specifica se si tratterà di una tua foto o di una foto presa da un altro sito web. Se utilizzerai una foto di un altro sito web, ricordati di riportare il link del sito).</i>	Titolo <i>(Che testo desideri che appaia insieme a questa foto e che possa essere visto dal tuo pubblico?)</i>
0.			
1.			
2.			
3.			

Pianificazione delle immagini (continua)

Numero	Immagine	Fonte dell'immagine <i>(Specifica se si tratterà di una tua foto o di una foto presa da un altro sito web. Se utilizzerai una foto di un altro sito web, ricordati di riportare il link del sito).</i>	Titolo <i>(Che testo desideri che appaia insieme a questa foto e che possa essere visto dal tuo pubblico?)</i>
4.			
5.			
6.			
7.			