## Overview

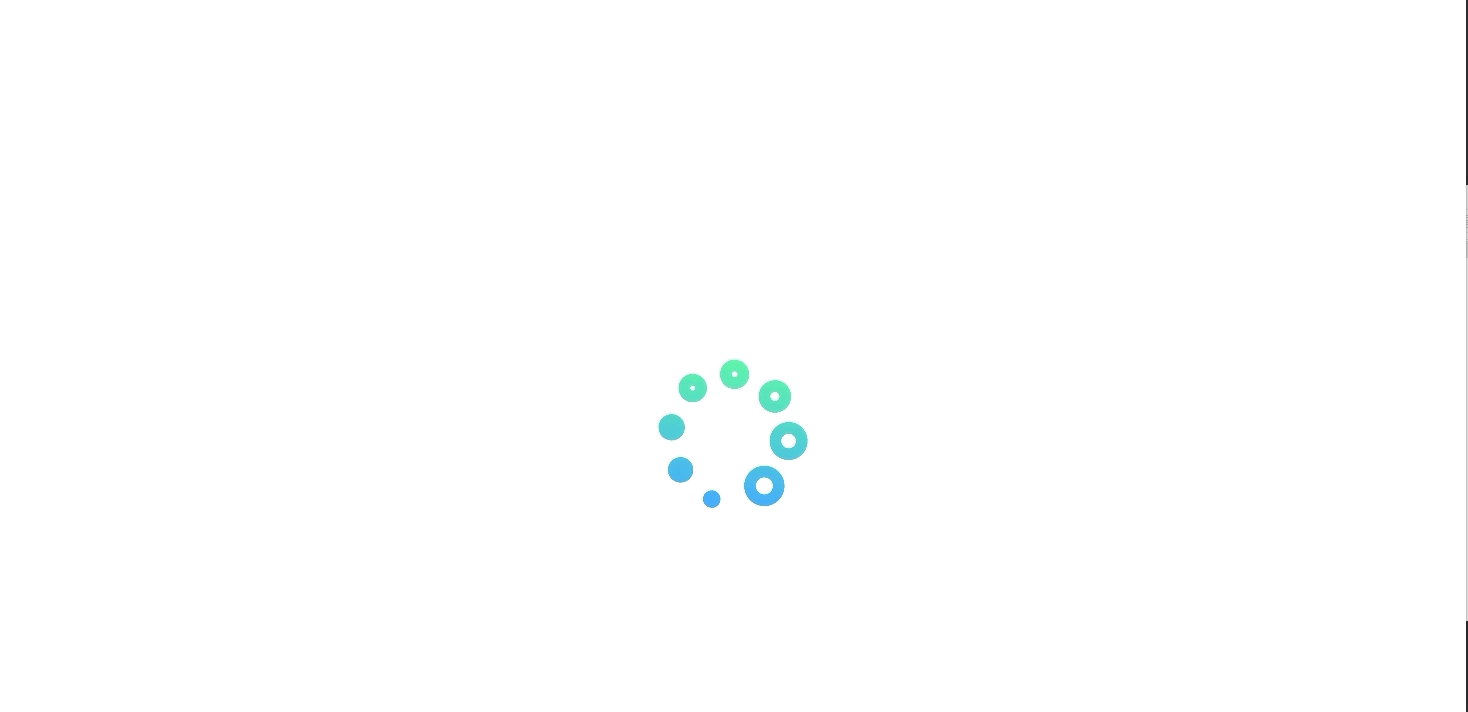
As a UX developer, you will frequently face the combination of a slow-loading and content-heavy site combined with business stakeholders that cannot cut any of the content—it’s all relevant and MUST be on the site.

In this activity, you will build on your previous front-end skills and add a preloader to a site in order to “hide” slow-loading content.

## Instructions

1. Open index.html located in Activities / Band\_Animated / unsolved.
2. Open index.css located in Activities / Band\_Animated / unsolved / css.
3. Lastly, also open index.js located in Activities / Band\_Animated / unsolved / js.

**Animation preview:**

****

1. Create a <div> with the class of preloader in index.html.
   * Add this <div> after the <footer> tag, but before the <script> tags.
2. Inside your new <div>, add an <img> tag with an src attribute that points to a loading.png in the images folder.
3. Now you will need to style your <div> so that it animates and appears vertically and horizontally centered on the page. These are all CSS properties that you have encountered before!
4. Add a selector that targets your <div>. Add the following properties to this selector:

| position: fixed;  width: 100%;  height: 100vh;  z-index: 100000;  background-color: white;  top: 0;  display: flex;  align-items: center; |
| --- |

1. Next, create a selector that targets the img **inside** of your <div>.
   * Remember, you can target elements inside of classes with the following syntax:

| .className elementType {  ...  } |
| --- |

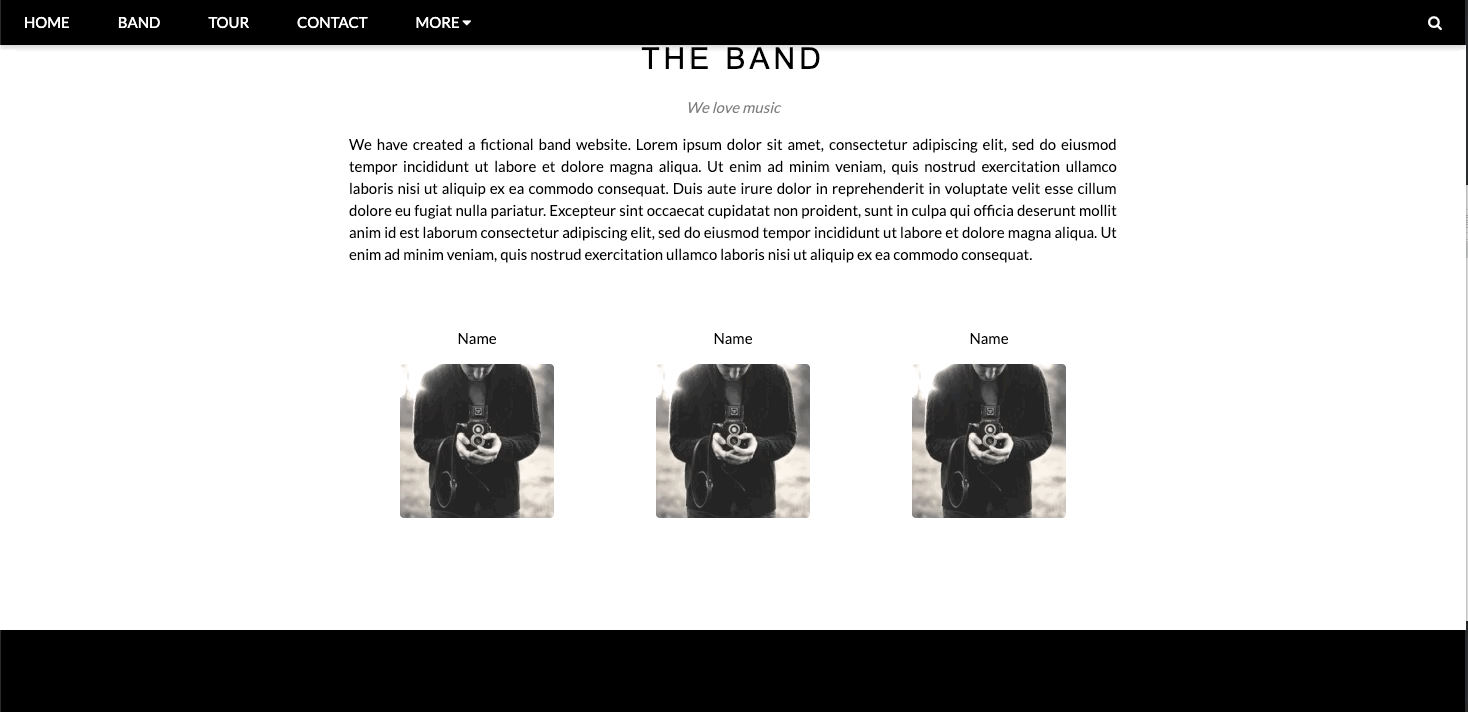
1. Add the following properties and values to this selector:
   * animation set to spin 3s ease
   * width of 150px
2. Next up, it’s time to add keyframes to create a spin effect.
   * Write a keyframe animation and name it spin (same name as above).
   * At 0% this animation should have the property transform: rotate(0deg);.
   * At 50% this animation should have the property transform: rotate(360deg);.
   * At 100% this animation should have the property transform: rotate(720deg);.
   * Reload the page and watch your animation spin!
3. You’re almost there, but there’s one problem: the preloader doesn’t fade to display our site’s content!
4. Inside index.js, create a function named fade.
   * Inside this function, target elements with the preloader class and use the built-in jQuery function fadeOut() to **slowly** fade out the selected element.
   * Remember, you can target elements in jQuery like this:

| $(element).fadeOut("slow"); |
| --- |

* + Note that you haven’t called this new function yet, so if you load your page, you will not see a result. We will be “calling” this function in the next step.

1. Call the setTimeout() function to call the fade function.
   * You can trigger one function (or many) with another function (or many others) and so on.
   * The setTimeout() function takes two parameters. **First**, the function to call. **Second**, the interval of time (in milliseconds) before the **first parameter** (the function) is called:

| setTimeout(functionName, 1000); |
| --- |

1. Put in fade and 3000 as the two parameters for setTimeout().
2. Reload your page to see the results!  
   

**Bonus**

Now that you have created your preloader, why not test out other preloader options? Create a new icon to animate or build your own effect.

* Visit <https://www.flaticon.com> to find different icons you can use (if you’re not in the mood for designing or don’t have time).