LEARN CSS ANIMATION BY BUILDING A FERRIS WHEEL

Introduction:

You can use CSS animation to draw attention to specific sections of your webpage and make it more engaging.

In this course, you'll build a Ferris wheel. You'll learn how to use CSS to animate elements, transform them, and adjust their speed.

Step 1:

Begin with the standard boilerplate. Add your `DOCTYPE` declaration, your `html` element with the language set to English, your `head` and `body` elements.

Add your `meta` element for the correct `charset`, your `title` element, and a `link` element for the `./styles.css` file.

Set the `title` to `Ferris Wheel`.

Step 2:

Add a `div` within your `body` element and give it a `class` of `wheel`.

Inside your new `div`, add six `span` elements with a `class` set to `line`, and six `div` elements with a `class` set to `cabin`.

Step 3:

Create a selector for your `.wheel` element. Start by setting the `border` to `2px solid black`, the `border-radius` to `50%`, and the `margin-left` to `50px`.

Step 4:

Set the `position` property of the `.wheel` selector to `absolute`. Set the `height` and `width` both to `55vw`.

Step 5:

Give your `.wheel` selector a `max-height` and `max-width` property both set to `500px`.

Step 6:

Create a selector for your `.line` elements. Start by setting the `background-color` to `black`, the `width` to `50%`, and the `height` to `2px`.

Step 7:

Set the `.line` selector's `position` property to `absolute`, the `left` property to `50%`, and the `top` property to `50%`.

Step 8:

The `transform-origin` property is used to set the point around which a CSS transformation is applied. For example, when performing a `rotate` (which you will do later in this project), the `transform-origin` determines around which point the element is rotated.

Give the `.line` selector a `transform-origin` property of `0% 0%`. This will offset the origin point by `0%` from the left and `0%` from the top, setting it to the top left corner of the element.

Step 9:

The `transform-origin` property is used to set the point around which a CSS transformation is applied. For example, when performing a `rotate` (which you will do later in this project), the `transform-origin` determines around which point the element is rotated.

Give the `.line` selector a `transform-origin` property of `0% 0%`. This will offset the origin point by `0%` from the left and `0%` from the top, setting it to the top left corner of the element.

Step 10:

Using the same pattern, create a separate selector for the third `.line`, the fourth `.line`, the fifth `.line`, and the sixth `.line`.

Set the `transform` property for the third `.line` to `rotate(120deg)`, the fourth to `rotate(180deg)`, the fifth to `rotate(240deg)`, and the sixth to `rotate(300deg)`.

Step 11:

Create a `.cabin` selector. Set the `background-color` to `red`, the `width` to `20%`, and the `height` to `20%`.

Step 12:

Give the `.cabin` a `position` of `absolute`, and a `border` of `2px solid`.

Step 13:

Set the `.cabin` to have a `transform-origin` property of `50% 0%`. This will set the origin point to be offset `50%` from the left and `0%` from the top, placing it in the middle of the top edge of the element.

Step 14:

Time to position the cabins around the wheel. Select the first `.cabin` element. Set the `right` property to `-8.5%` and the `top` property to `50%`.

Step 15:

Continuing the pattern, select the following `.cabin` elements and apply the specific rules to them:

- The second `.cabin` should have the `right` property set to `17%` and the `top` property set to `93.5%`.
- The third `.cabin` should have the `right` property set to `67%` and the `top` property set to `93.5%`.
- The fourth `.cabin` should have the `left` property set to `-8.5%` and the `top` property set to `50%`.
- The fifth `.cabin` should have the `left` property set to `17%` and the `top` property set to `7%`.
- The sixth `.cabin` should have the `right` property set to `17%` and the `top` property set to `7%`.

Step 16:

The `@keyframes` at-rule is used to define the flow of a CSS animation. Within the `@keyframes` rule, you can create selectors for specific points in the animation sequence, such as `0%` or `25%`, or use `from` and `to` to define the start and end of the sequence.

`@keyframes` rules require a name to be assigned to them, which you use in other rules to reference. For example, the `@keyframes freeCodeCamp { }` rule would be named `freeCodeCamp`.

Time to start animating. Create a `@keyframes` rule named `wheel`.

Step 17:

You now need to define how your animation should start. To do this, create a `0%` rule within your `@keyframes wheel` rule. The properties you set in this nested selector will apply at the beginning of your animation.

As an example, this would be a `12%` rule:

Example Code:

```
@keyframes freecodecamp {
  12% {
    color: green;
  }
}
```

Step 18:

Give the `0%` rule a `transform` property set to `rotate(0deg)`. This will start the animation with no rotation.

Step 19:

Now give the `@keyframes wheel` rule a `100%` selector. Within that, set the `transform` to `rotate(360deg)`. By doing this, your animation will now complete a full rotation.

Step 20:

The `animation-name` property is used to link a `@keyframes` rule to a CSS selector. The value of this property should match the name of the

@keyframes rule. Give your .wheel selector an animation-name property set to wheel.

The animation-duration property is used to set how long the animation should sequence to complete. The time should be specified in either seconds (s) or milliseconds (ms). Set your .wheel selector to have an animation-duration property of 10s.

Step 21:

The animation-iteration-count property sets how many times your animation should repeat. This can be set to a number, or to infinite to indefinitely repeat the animation. Your Ferris wheel should never stop, so set the .wheel selector to have an animation-iteration-count of infinite.

The animation-timing-function property sets how the animation should progress over time. There are a few different values for this property, but you want the Ferris wheel animation to run at the same rate from start to finish. Set the animation-timing-function to linear in your .wheel selector.

Step 22:

Create another @keyframes rule with the name cabins. Use the same properties as your @keyframes wheel, copying both the 0% and 100% rules, but set the transform property of the 100% selector to rotate(-360deg).

Step 23:

With your .wheel selector, you created four different properties to control the animation. For your .cabin selector, you can use the animation property to set these all at once.

Set the `animation` property of the `.cabin` rule to `cabins 10s linear infinite`. This will set the `animation-name`, `animation-duration`, `animation-timing-function`, and `animation-iteration-count` properties in that order.

Step 24:

To make your cabin animation seem more like a natural swinging motion, you can use the `ease-in-out` timing function. This setting will tell the animation to start and end at a slower pace, but move more quickly in the middle of the cycle.

Replace `linear` to `ease-in-out` in the `.cabin` selector.

Step 25:

You can use `@keyframes` rules to control more than just the transformation of an element. In the `0%` selector of your `@keyframes cabins`, set the `background-color` to `yellow`.

Step 26:

Between the `0%` and `100%` selectors, add a `50%` selector. This will apply in the middle of the animation cycle. Set the `background-color` to `purple`.

Step 27:

Because the animation is on an infinite loop and the start and end colors are not the same, the transition appears jerky when it switches back to yellow from red.

To start fixing this, remove the `background-color` from your `0%` selector.

Step 28:

Create a new `25%` selector between your `0%` and `50%` selectors. Give this new selector the `background-color` property set to `yellow`.

Step 29:

Finally, create a new `75%` selector between your `50%` and `100%` selectors. Give this new selector a `background-color` property set to `yellow`.

With that, your animation is much smoother and your Ferris wheel is complete.