

5:52

- Making Your First Chart
- LESSON 2: (5:51) Loading the weather dataset
- LESSON 3: (8:35) Setting up our line chart
- LESSON 4: (9:15) Drawing our chart
- LESSON 5: (3:24) Creating our workspace
- LESSON 6: (4:28) Adding an SVG element
- LESSON 7: (4:42) Creating our bounding box
- LESSON 8: (10:06) Creating our scales
- LESSON 9: (6:27) Drawing the line
- LESSON 10: (5:08) Drawing the axes
- LESSON 11: Week 1: Exercise
- APPENDIX 1: Datasets

MODULE 1: MAKING A SCATTERPLOT

- LESSON 1: (1:28) Making a Scatterplot
- LESSON 2: (2:03) Steps in drawing any chart
- LESSON 3: (1:57) Step 1: Access data
- LESSON 4: (7:24) Step 2: Create chart dimensions
- LESSON 5: (2:59) Step 3: Draw canvas
- LESSON 6: (4:20) Step 4: Create scales
- LESSON 7: (15:27) Step 5: Draw data
- LESSON 8: (17:29) Step 6: Draw peripherals
- LESSON 9: (1:55) Looking at our chart
- LESSON 10: (5:47) Extra credit: adding a color scale
- LESSON 11: Week 2: Exercise

MODULE 2: MAKING A BAR CHART

- LESSON 1: (3:31) Making a Bar Chart
- LESSON 2: (2:03) Access data
- LESSON 3: (2:49) Create dimensions
- LESSON 4: (2:25) Draw canvas
- LESSON 5: (10:25) Create scales
- LESSON 6: (8:35) Draw data
- LESSON 7: (7:11) Adding Labels
- LESSON 8: (14:28) Draw peripherals
- LESSON 9: (8:58) Extra credit
- LESSON 10: (13:13) Accessibility
- LESSON 11: Week 3: Exercise

MODULE 3: ANIMATIONS AND TRANSITIONS

- LESSON 1: (3:00) Animations and Transitions
- LESSON 2: (8:47) CSS transitions
- LESSON 3: (8:14) CSS transitions with a chart
- LESSON 4: (22:12) d3.transition
- LESSON 5: (11:41) Lines
- LESSON 6: Week 4: Exercise

MODULE 4: INTERACTIONS

- LESSON 1: (0:55) Interactions
- LESSON 2: (7:09) d3 events
- LESSON 3: (3:09) Destroying d3 event listeners
- LESSON 4: (18:51) Bar chart
- LESSON 5: (25:04) Scatter plot
- LESSON 6: (24:19) Line chart
- LESSON 7: Week 5: Exercise

MODULE 5: DATA VISUALIZATION BASICS

- LESSON 1: (4:17) Data Visualization Basics
- LESSON 2: (4:52) Types of data
- LESSON 3: (4:49) Ways to visualize a metric
- LESSON 4: (2:58) Chart design
- LESSON 5: (7:43) Example redesign
- LESSON 6: (5:26) Color scales
- LESSON 7: (4:48) Custom color scales
- LESSON 8: (9:33) Color spaces
- LESSON 9: (4:36) Color tips
- LESSON 10: Week 6: Exercise

MODULE 6: ADVANCED EXAMPLE

- LESSON 1: (2:23) Radar Weather Chart
- LESSON 2: (7:16) Getting set up
- LESSON 3: (34:54) Adding gridlines
- LESSON 4: (2:17) Adding freezing
- LESSON 5: (12:55) Adding the temperature area
- LESSON 6: (8:26) Adding the UV index marks
- LESSON 7: (7:38) Adding the cloud cover bubbles
- LESSON 8: (7:34) Adding the precipitation bubbles
- LESSON 9: (20:09) Adding annotations
- LESSON 10: (30:51) Adding the tooltip
- LESSON 11: (0:52) Wrapping up
- LESSON 12: Week 7: Exercise

MODULE 7: D3 + JAVASCRIPT FRAMEWORKS

- LESSON 1: (2:48) D3 + Javascript Frameworks
- LESSON 2: (3:11) React.js
- LESSON 3: (4:48) Digging in
- LESSON 4: (4:03) Creating dimensions in React
- LESSON 5: (6:12) Drawing our canvas in React
- LESSON 6: (3:39) Creating our scales in React
- LESSON 7: (5:26) Drawing our data in React
- LESSON 8: (7:34) Drawing our peripherals in React
- LESSON 9: (15:11) Drawing our peripherals in React, take two
- LESSON 10: (4:27) Setting up interactions in React, and wrapping up
- LESSON 11: (7:41) Using d3 with Angular.js
- LESSON 12: (7:30) Using d3 with Svelte.js

MODULE 8: INTERVIEWS

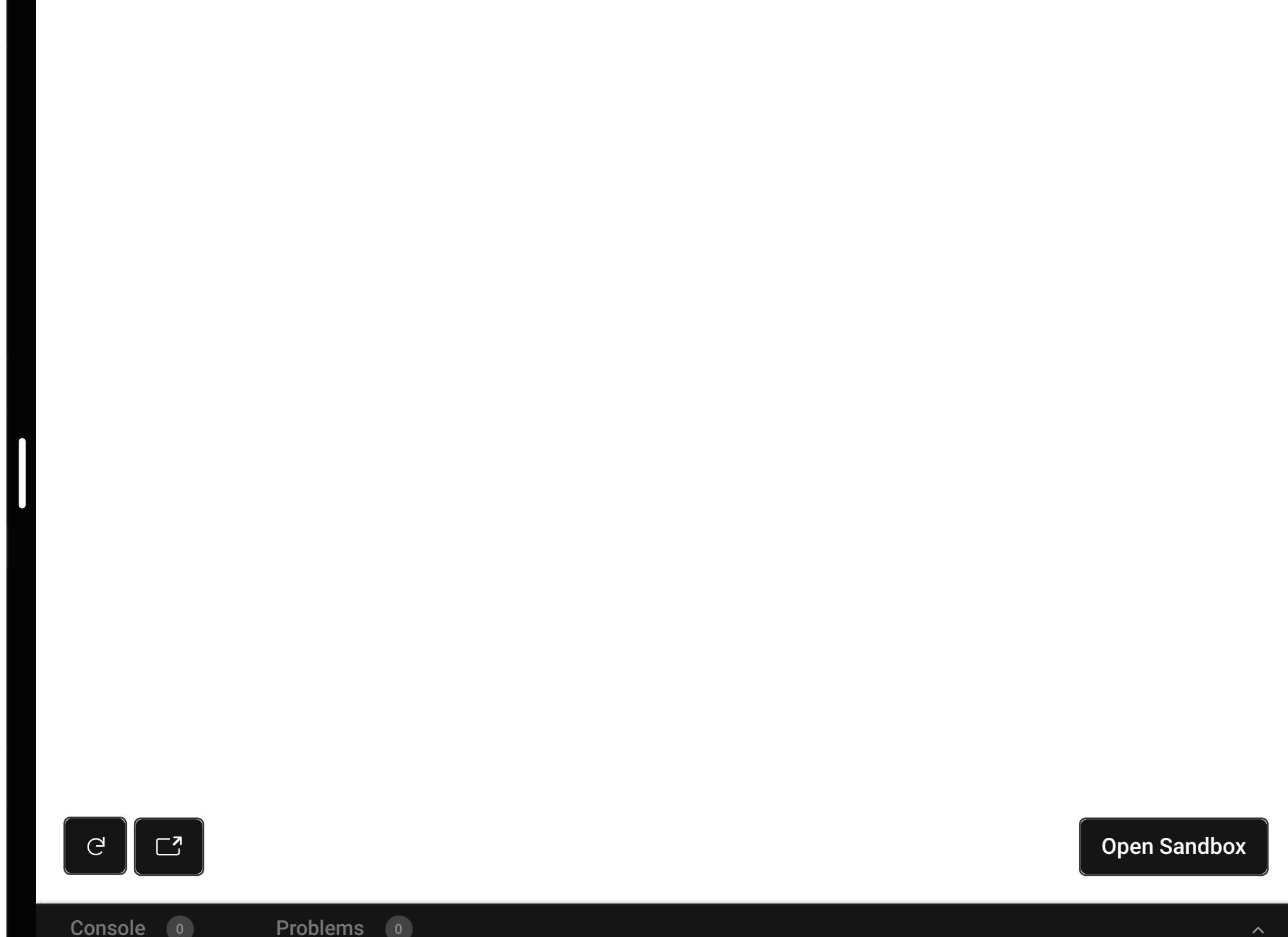
- LESSON 1: Interviews
- LESSON 2: (18:01) Shirley Wu
- LESSON 3: (18:07) Ian Johnson
- LESSON 4: (14:45) Russell Goldenberg
- LESSON 5: (28:42) Will Chase

Fullstack D3 Masterclass > Animations and Transitions

CSS transitions

We learn about CSS transitions and the different CSS transition properties, then run through a concrete example and look at how to debug them.

LESSON DISCUSSION 1



Many of our chart changes can be transitioned with the CSS `transition` property. When we update a `<rect>` to have a fill of **red** instead of **blue**, we can specify that the color change take 10 seconds instead of being instantaneous. During those 10 seconds, the `<rect>` will continuously re-draw with intermediate colors on the way to **red**.

Let's try out an example! In our example, you'll see a blue box that moves and turns green on hover.



Let's open up the

`styles.css` file to take a look at what's going on. We can see our basic styles for the box.

And our styles that apply to our box when it is hovered (change the background color and move it 30 pixels to the right).

To create CSS a transition, we need to specify how long we want the animation to take with the `transition-duration` property. The property value accepts **time CSS data types** — a number followed by either `s` (seconds) or `ms` (milliseconds).

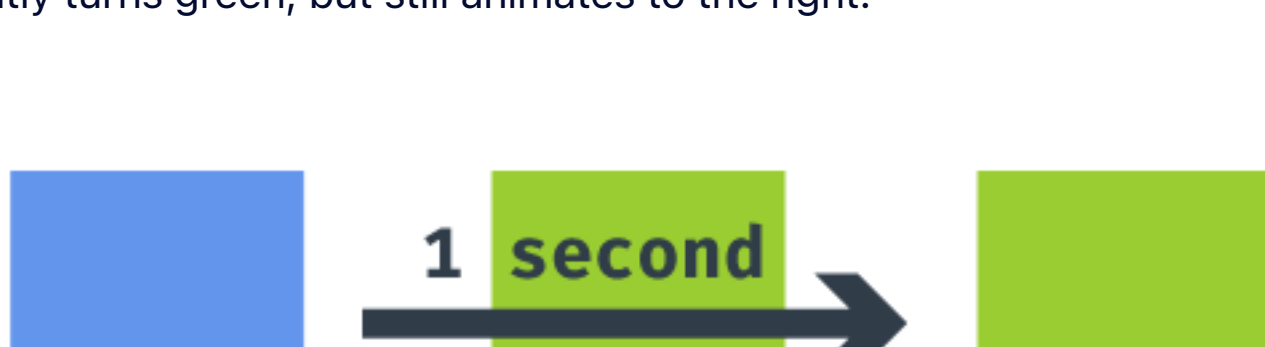
Let's make our box changes animate over one second.

Now when we hover over the box, we can see it slowly move to the right and turn green. Smooth!



Now let's say that we only want to animate our box's movement, but we want the color change to happen instantaneously. This is possible by specifying the `transition-property` CSS property. By default, `transition-property` is set to `all`, which animates all transitions. Instead, let's override the default and specify a specific CSS property name (`transform`).

Now our box instantly turns green, but still animates to the right.



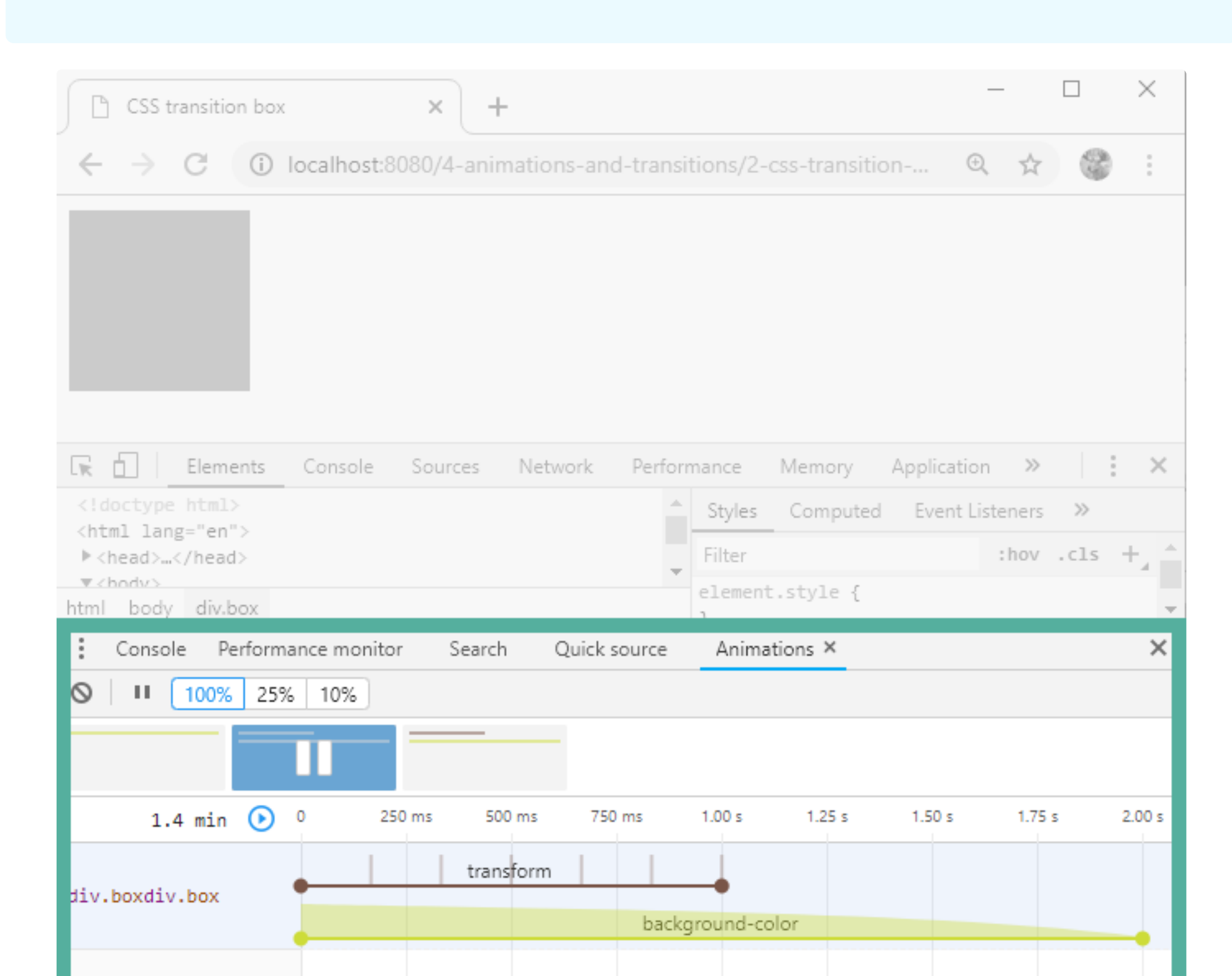
Instead of setting `transition-duration` and `transition-property` separately, we can use the shorthand property: `transition`. Shorthand CSS properties let you set multiple properties in one line. When we give `transition` a CSS property name and duration (separated by a space), we are setting both `transition-duration` and `transition-property`. Let's try it out.

`transition` will accept a third property (`transition-timing-function`) that sets the acceleration curve for the animation. The animation could be linear (the default), slow then fast (`ease-in`), in steps (`steps(6)`), or even a custom function (`cubic-bezier(0.1, 0.7, 1.0, 0.1)`), among other options. Let's see what `steps(6)` looks like — it should break the animation into 6 discrete steps.

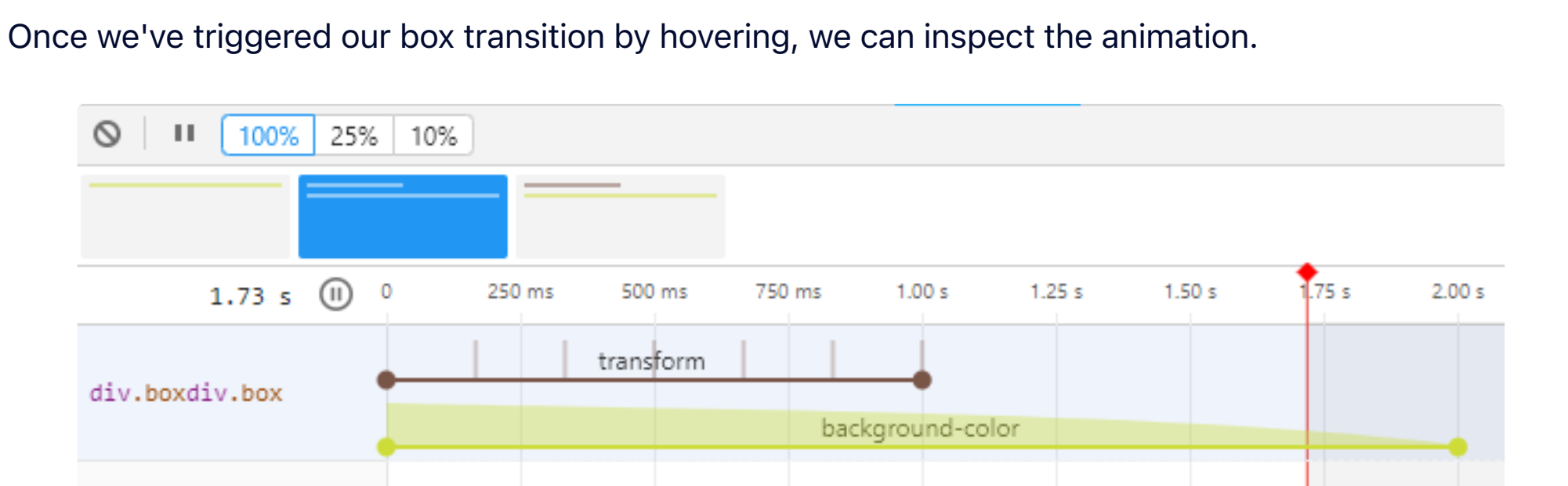
What if we wanted to animate the color change, but finish turning green *while* our box is shifting to the right? `transition` will accept multiple transition statements, we just need to separate them by a comma. Let's add a transition for the `background-color`.

Nice! Now our box transitions by stepping to the right, while turning green over two seconds. Chrome's dev tools have a great way to visualize this transition. Press `esc` when looking at the **Elements** panel to bring up the bottom panel. In the bottom panel, we can open up the **Animations** tab.

If you don't see the **Animations** tab, click on the kebab menu on the left and select it from the dropdown options.



Once we've triggered our box transition by hovering, we can inspect the animation.



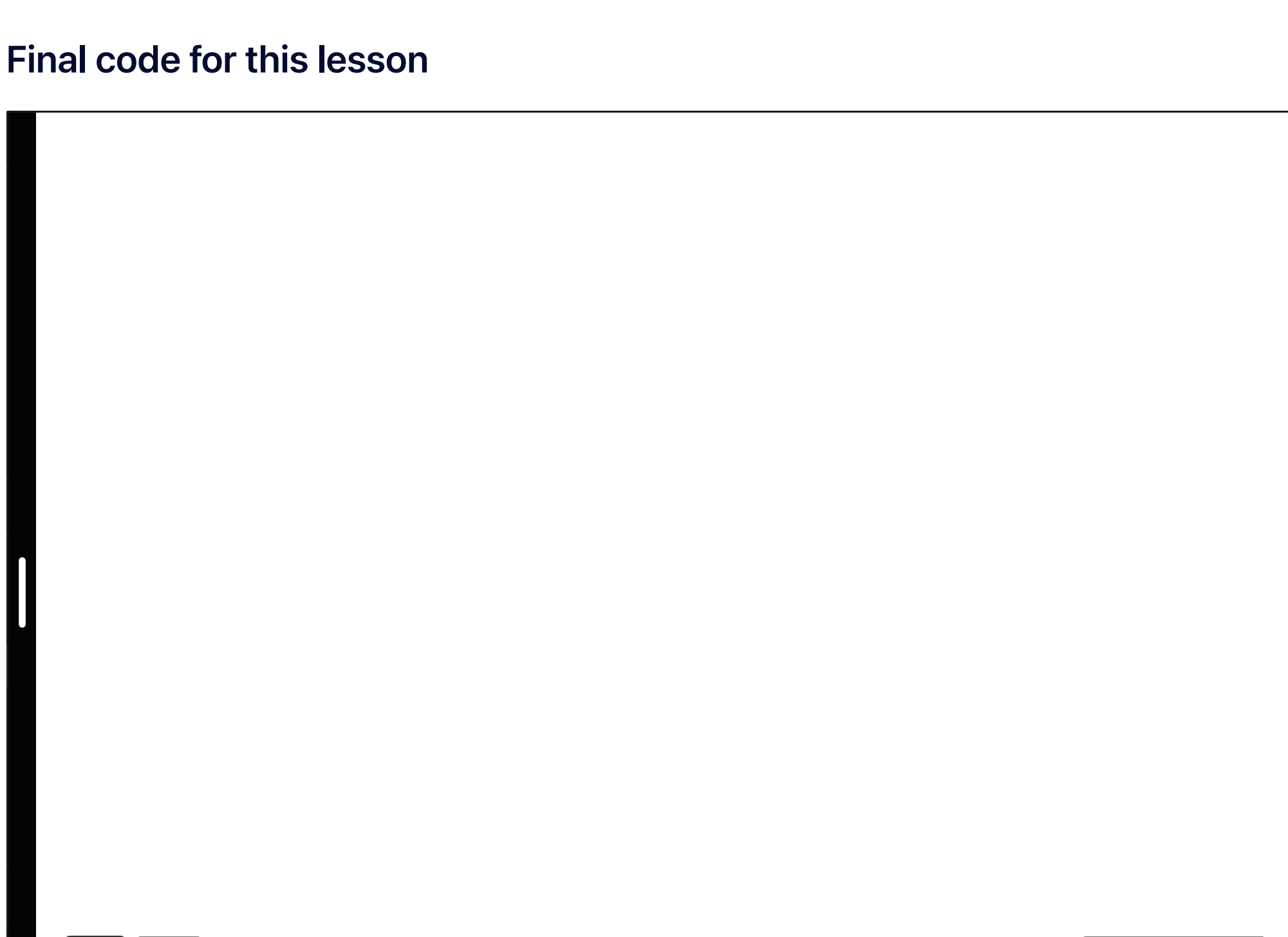
We can see the `transform` transition on top, with six discrete changes, and the `background-color` animation on the bottom, easing gradually from one color to the next. The `background-color` transition diagram is twice as wide as the `transform` transition diagram, indicating that it takes twice as long.

This view can be very handy when inspecting, tweaking, and debugging transitions.

Now that we're comfortable with CSS `transition`, let's see how we might use it to animate our charts.

Not all properties can be animated. For example, how would you animate changing a label from **Humidity** to **Dew point**? However most properties can be animated, so feel free to operate under the assumption that a property can be animated until proven otherwise.

Final code for this lesson



Previous Lesson: Animations and Transitions

Next Lesson: CSS transitions with a chart