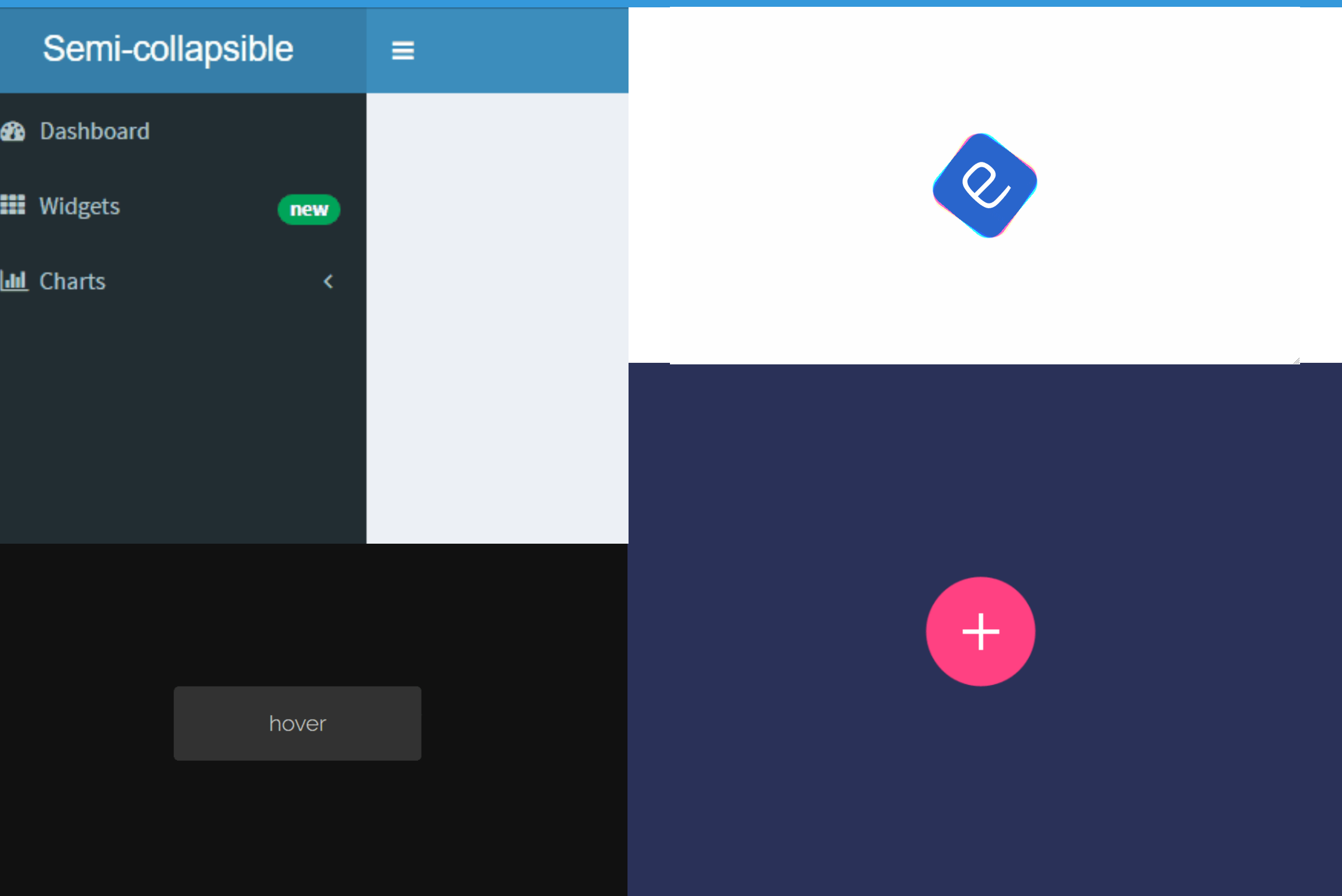


The background of the image is a grayscale photograph of a workspace. In the upper left, a portion of a laptop screen is visible, showing a file explorer with folders like 'apl.', 'composer', 'cron.sql', 'dump.php', 'index.php', 'pdf.php', and 'sitemap.php'. Below the screen is a cup of coffee with a thick, brown, frothy top. To the right of the cup is a laptop keyboard with black keys and white lettering. In the lower right corner, a smartphone is partially visible, displaying a social media profile page. The profile shows a name, a bio, and statistics: '16.8 k followers' and '191 following'. The text 'Transitions & Transforms' is centered over the image in a large, white, serif font, flanked by two horizontal red lines.

Transitions & Transforms

WEEK 6

MOTION IN DESIGN



CSS animation, video and other motion media can improve design not only aesthetically, but also be a practical tool as visual cues and interactivity feedback.

CSS ANIMATABLE

Property

[background](#)

[background-color](#)

[background-position](#)

[background-size](#)

[border](#)

[border-bottom](#)

[border-bottom-color](#)

[border-bottom-left-radius](#)

[border-bottom-right-radius](#)

[border-bottom-width](#)

[border-color](#)

[border-left](#)

[\[FULL ANIMATABLE >> \]](#)

2D TRANSFORMS

`translate()`

`rotate()`

`scaleX()`

`scaleY()`

`scale()`

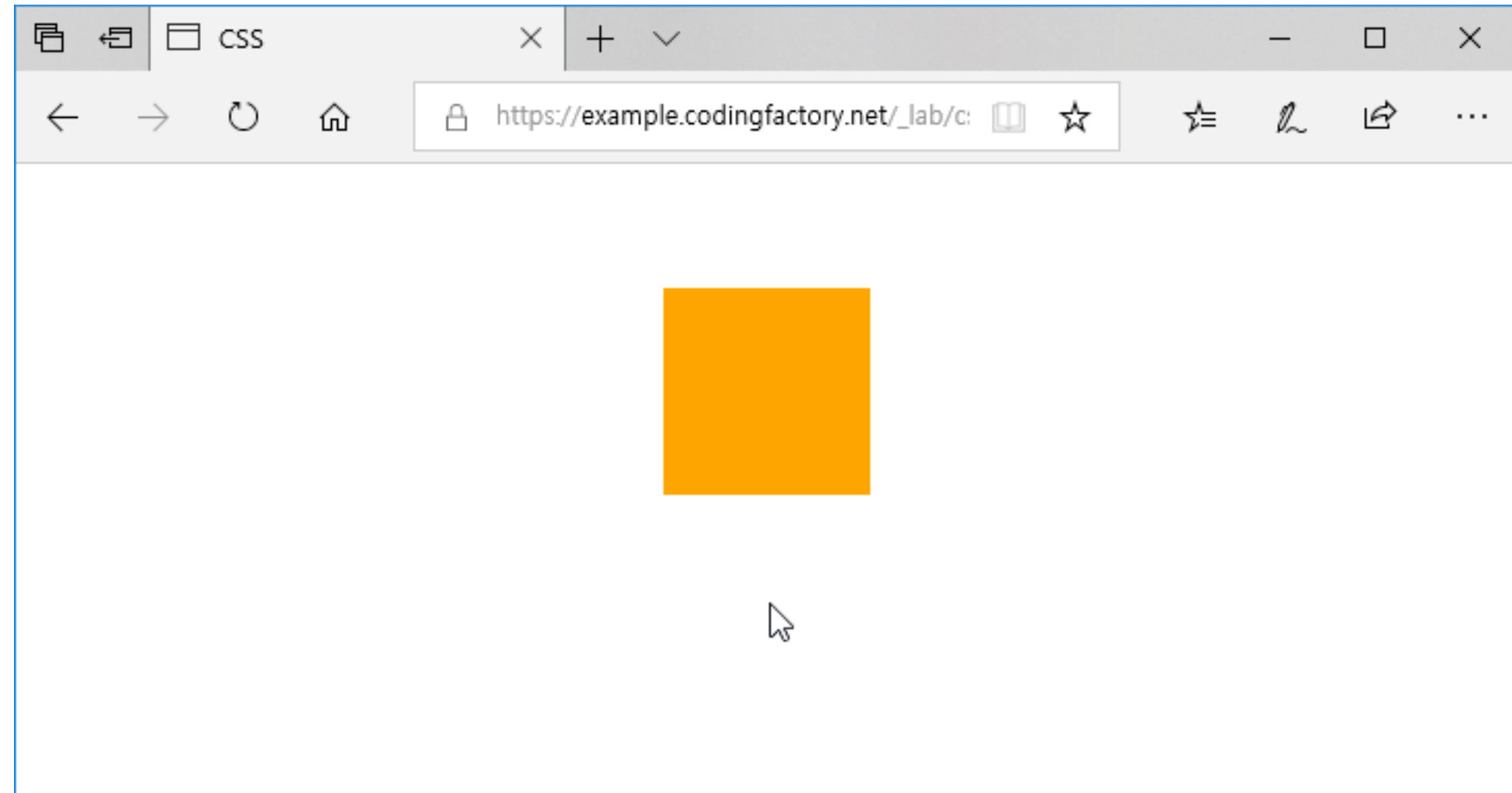
`skewX()`

`skewY()`

`skew()`

`matrix()`

[\[W3 Reference >> \]](#)



2D TRANSFORMS

`translate()`

`rotate()`

`scaleX()`

`scaleY()`

`scale()`

`skewX()`

`skewY()`

`skew()`

`matrix()`



[\[W3 Reference >> \]](#)

2D TRANSFORMS

```
div {  
  transform:  
    translate(50px,100px); /*  
    Standard syntax */  
}
```

This div element is moved 50 pixels to the right, and 100 pixels down from its current position.

[\[W3 Example >> \]](#)

2D TRANSFORMS

```
div {  
  transform: rotate(20deg);  
}
```

This a normal div element.

This div element is rotated clockwise 20 degrees.

[\[W3 Example >> \]](#)

2D TRANSFORMS

```
div {  
  transform: scale(2,3);  
}
```

This div element is two times of its original width, and three times of its original height.

[\[W3 Example >> \]](#)

2D TRANSFORMS

```
div {  
  transform: skewX(20deg);  
}
```

This a normal div element.

This div element is skewed 20 degrees along the X-axis.

[\[W3 Example >> \]](#)

3D TRANSFORMS



Move in 3D

Layer elements along the Z-axis by adding space between them.



Rotate in 3D

Rotate an element along its horizontal and / or vertical axes.



Apply perspective

Play with perspective to intensify your 3D effects.

[\[W3 Reference >> \]](#)

3D TRANSFORMS

rotateX()

rotateY()

rotateZ()

```
div {  
    transform: rotateY(150deg);  
    transform: rotateX(10deg);  
    transform: rotateZ(50deg);  
}
```

[\[W3 Reference >> \]](#)

ANIMATION

{transition}

```
div {  
  width: 100px;  
  height: 100px;  
  background: red;  
  transition: width 2s;  
}  
  
div:hover {  
  width: 300px;  
}
```

[\[W3 Reference >> \]](#)

ANIMATION

{transition}

```
div {  
  width: 100px;  
  height: 100px;  
  background: red;  
  transition: width 2s;  
}
```

```
div:hover {  
  width: 300px;  
}
```

Exercise

Apply a 2 second transition to the width property of the div. Define an ease-in-out timing function for the transition.



Hover over the div element above.

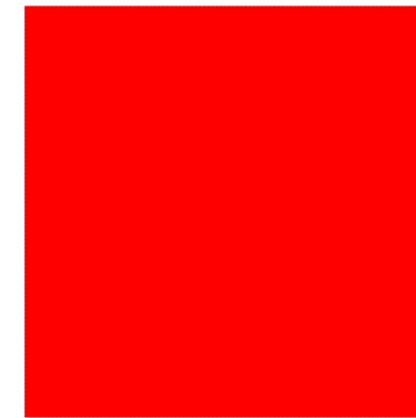
[\[W3 Reference >> \]](#)

ANIMATION

@keyframes { }

```
@keyframes example {  
  from {background-color: red;}  
  to {background-color: yellow;}  
}
```

Note: This example does not work in Internet Explorer 9 and earlier versions.



Note: When an animation is finished, it changes back to its original style.

[\[W3 Reference >> \]](#)

ANIMATION

[\[W3 Reference >> \]](#)

@keyframes { }

```
@keyframes example {  
  0%    {background-color: red;}  
  25%   {background-color: yellow;}  
  50%   {background-color: blue;}  
  100%  {background-color: green;}  
}
```

Note: This example does not work in Internet Explorer 9 and earlier versions.

