

Chapter 8

Slideshow

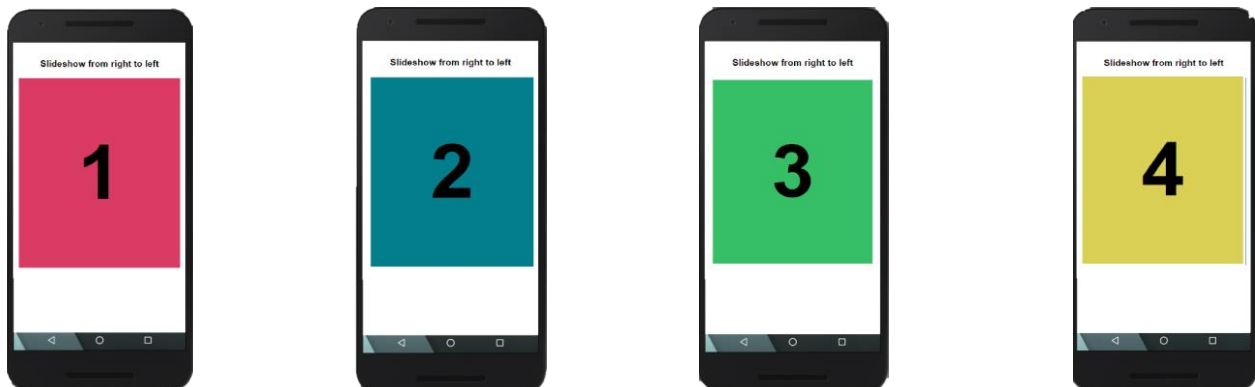
Slideshow is one of the fun features in web app. It presents to the viewer a series of information, images, or text slides looping in certain time. The purpose of a slideshow is to promote visual interest or artistic value, sometimes accompanied with description or text, or to clarify or reinforce information, ideas, comments, solutions, or suggestions.



Slideshow in web app differs from website by its simplicity of the slideshow presentation. For example, since the smartphone screen is smaller than the computer screen, some features, such as next and previous arrow or the slide indicators, are omitted in the slideshow of a web app, this will permit the user to see clearly the message or images of the slideshow.

There are different way to create slideshow, one of the way to create slideshow in CSS is to use the *animation* feature.

Example 1) Create an infinite slideshow to display a number from 1 to 4, increasing order, with different background color.



Let's set up the html structure with a link to index.css

HTML

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="index.css" rel="stylesheet" type="text/css"/>
    <title>Slideshow by Huixin Wu</title>
  </head>
  <body>

  </body>
</html>
```

In index.css, we are going to set the **body** padding to 2%. This will prevent that elements within the body sit in the border of the smartphone's screen.

CSS

```
body {
  font-family: Helvetica, sans-serif;
  padding: 2%;
  text-align: center;
}
*{
  box-sizing: border-box;
}
```

Now that we have the **body** set, we can create the slide container with the id name as **slideshow1**

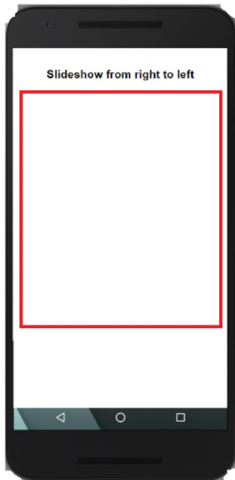
HTML

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="index.css" rel="stylesheet" type="text/css"/>
    <title>Slideshow by Huixin Wu</title>
  </head>
  <body>
    <div id="slideshow1">

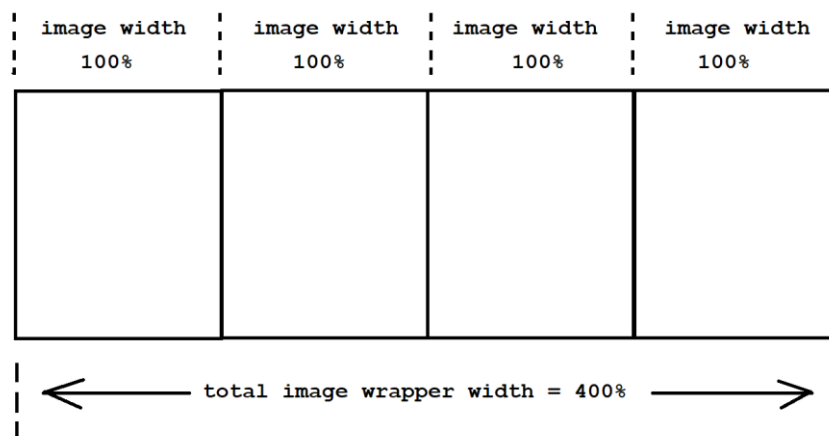
    </div> <!--end of slideshow1 -->
  </body>
</html>
```

Once we have the html structure for slideshow1, in css, we are going to set the height to 400px, the width to fit 100% of the app screen, and all the overflow information in slideshow1 will be hidden. Also, we are going to add border to slideshow1 for reference, but remember that the border will be deleted once the slideshow frame is set.

```
#slideshow1{  
  overflow: hidden;  
  height: 400px;  
  width: 100%;  
  border: solid red;  
}
```

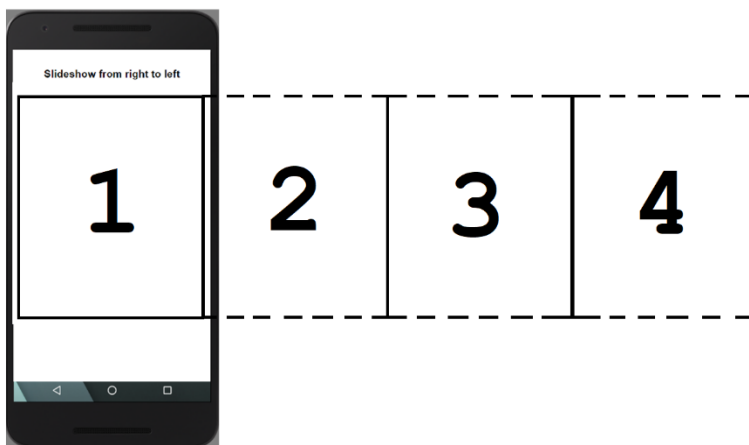


After it, we need to create a container to wrap all four numbers. We are going to name the wrapper **slide-wrapper**. The total width of the **slide-wrapper** is the sum of the width of each image in the slideshow. For example, if we are planning to display four images in a slideshow and each image will display 100% to the width of the app screen, then the width of the **slide-wrapper** will be 400%.

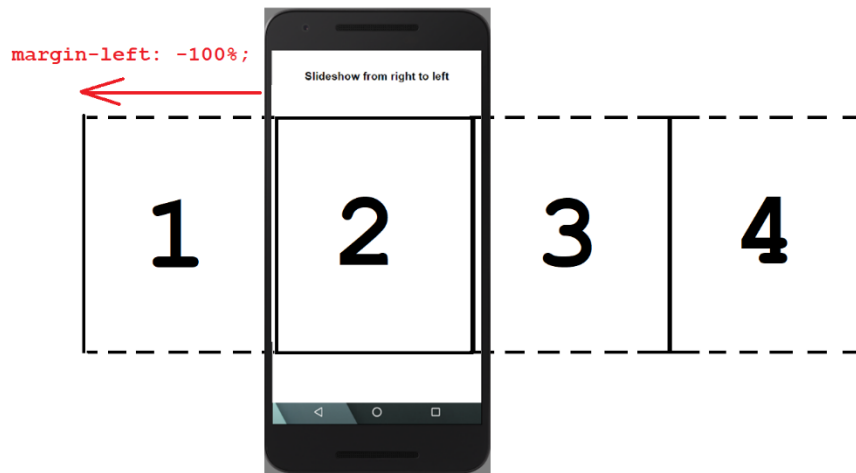




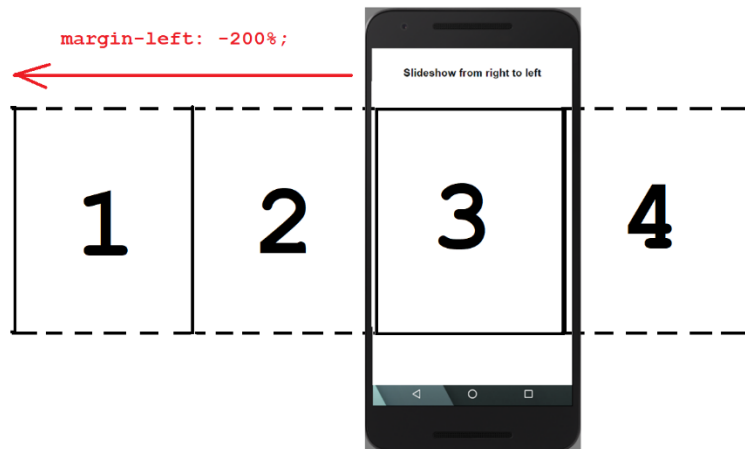
The other feature that we need to add to the **slide-wrapper** is the animation. The animation will slide each image in the slideshow from right to left one at the time. To do so, at each time frame the **slide-wrapper** will add margin to the left by 100% of the image width. Since the **slide-wrapper** is shifted to the left, the **margin-left** should be set to **-100%**. For example, at the first time frame, the slideshow will show the first image:



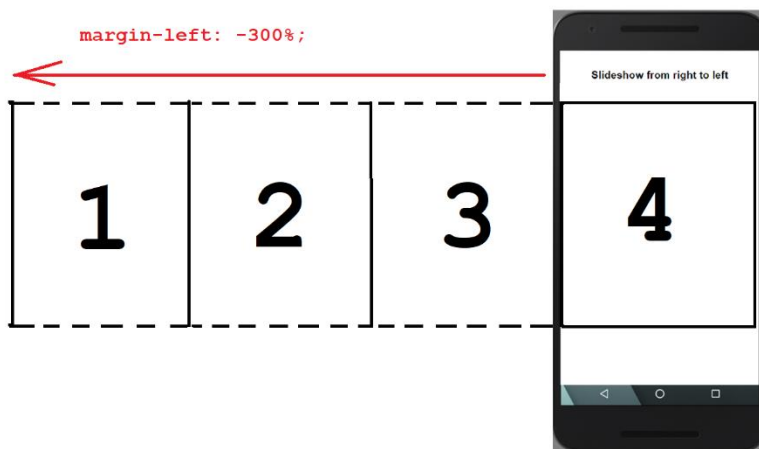
At the second time frame, the **margin-left** set to **-100%** to show the second image:



At the third time frame, the **margin-left** set to **-200%** to show the third image:



At the fourth time frame, the **margin-left** set to **-300%** to show the fourth image:



Add animation to `.slide-wrapper`

We are naming the animation as `slideshow`, the total animation duration is 12 seconds, and it is an infinite slideshow

```
.slide-wrapper {  
  width: 400%;  
  height: 100%;  
  animation: slideshow 12s infinite;  
}  
  
@keyframes slideshow {  
  0% {margin-left: 0px;}  
  35% {margin-left: -100%;}  
  70% {margin-left: -200%;}  
  100% {margin-left: -300%;}  
}
```

CSS

If we run the slideshow, nothing is showing because we have not set each image into the **slide-wrapper** yet. Since we are planning to show four numbers, then we need to create four slide divisions:

```
<!DOCTYPE html>  
<html lang="en" dir="ltr">  
  <head>  
    <meta charset="utf-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <link href="index.css" rel="stylesheet" type="text/css"/>  
    <title>Slideshow by Huixin Wu</title>  
  </head>  
  <body>  
    <div id="slideshow1">  
      <div class="slide-wrapper">  
        <div class="slide"><h1>1</h1></div>  
        <div class="slide"><h1>2</h1></div>  
        <div class="slide"><h1>3</h1></div>  
        <div class="slide"><h1>4</h1></div>  
      </div> <!--end of slide-wrapper -->  
    </div> <!--end of slideshow1 -->  
  </body>  
</html>
```

HTML

Since all four **slide** divisions is set inside the **slide-wrapper**, then each **slide** division width must be set to 25%, $\frac{100\%}{4} = 25\%$

Also, all four divisions will be in-line from left to right, for this, we can set all four divisions to float left

```
.slide {  
  float: left;  
  height: 100%;  
  width: 25%;  
}
```

CSS

We can also add some CSS to all **h1** within **slide** class:

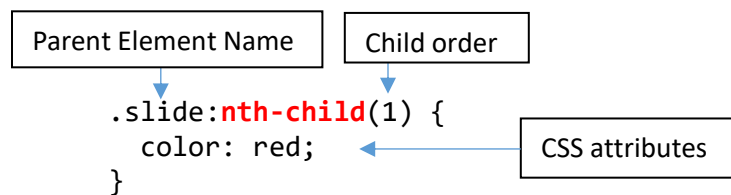
```
.slide h1{
  text-align: center;
  font-size: 10em;
}
```

CSS

Now, we can run the slideshow and see the number displaying one at the time.

The next features to add to each slide is the background color. There is different way we can set different background color to each slide, for example, we can use different class name to each slide division, or add the CSS code in the HTML code using **style="background-color: yellow;"**

For this activity, we are going to use the **nth-child** selector in CSS to set different background color to each **slide** division. The **nth-child** selector is a structural pseudo-class, meaning it is used to style content based on its relationship with the parent element. For our program, the parent element will be **slide** because **slide** has four divisions of the same class name. Then, each **slide** division, according to line order, will be the **nth-child** of the **slide** element. The syntax of the **nth-child** is:



For example, if we are setting the background color for the first slide to lightblue, then we can write the code as:

```
.slide:nth-child(1) {
  background-color: lightblue;
}
```

Then, to set different background color to each four slide elements:

```
.slide:nth-child(1) {
  background-color: lightblue;
}

.slide:nth-child(2) {
  background-color: pink;
}

.slide:nth-child(3) {
  background-color: lightgreen;
}

.slide:nth-child(4) {
  background-color: orange;
}
```

CSS

Now, we can run the slideshow and see the number of each slide changes with different background color.

If we want each number of the slideshow to rest before it shifts to the next number, we can adjust the previous animation's time frame, `@keyframes slideshow`, to rest by 10% of the total time frame at each slide, and 20% of the total time frame as the transition time between slide:

```
@keyframes slideshow {
  0% {margin-left: 0px;}
  10% {margin-left: 0px;}
  30% {margin-left: -100%;}
  40% {margin-left: -100%;}
  60% {margin-left: -200%;}
  70% {margin-left: -200%;}
  90% {margin-left: -300%;}
  100% {margin-left: -300%;}
}
```

CSS

Run the slideshow and see how the number rest for 1.2 seconds, 10% of the total time frame 12 seconds, before it moves to the next number.

The other features that we can add to the animation is to see slideshow images to backward to the beginning to the slideshow when it reaches the last slide. For this, we can adjust the time frame to up 85% for the last slide. This will allow the slideshow to have the rest 86% to 100% to backward to the beginning to the slideshow. Therefore, the rest time at each slide will be 10% of the total time frame, the transition time between slides will be 15% of the total time frame, and the backward time will be 15% of the total time frame, which is from 86% to 100%:

```
@keyframes slideshow {
  0% {margin-left: 0px;}
  10% {margin-left: 0px;}
  25% {margin-left: -100%;}
  35% {margin-left: -100%;}
  50% {margin-left: -200%;}
  60% {margin-left: -200%;}
  75% {margin-left: -300%;}
  85% {margin-left: -300%;}
}
```

CSS