<script>

const previous = document.querySelector('.previous');

const next = document.querySelector('.next');

const images = document.querySelector('.slider-carousel').children;

const totalImages = images.length;

let currentIndex = 0;

// Event Listeners to previous and next buttons

previous.addEventListener('click', () => {

previousImage()

})

next.addEventListener('click', () => {

nextImage();

})

// Function to go to next Image

function nextImage(){

images[currentIndex].classList.remove('main');

if(currentIndex == totalImages-1){

currentIndex = 0;

}

else{

currentIndex++;

}

images[currentIndex].classList.add('main');

}

// Function to go to previous Image

function previousImage(){

images[currentIndex].classList.remove('main');

if(currentIndex == 0){

currentIndex = totalImages-1;

}

else{

currentIndex--;

}

images[currentIndex].classList.add('main');

}

</script>

</body>

</html>

Output

example-activate-two-buttons

Congratulations, you have created CSS Slider using HTML, CSS, and Javascript.

Slide Show Functionality

Our Slider is currently using manual control to work. This means, that until the user clicks on the previous or next button, the image won't change. However, most of the sliders present online do not work like this. They have a slide show effect which allows them to change frames automatically after a set time has passed.

We can do that too in our slider. Using javascript, we can use the setInterval function that will automatically call the next function after a set time. This will help the slider to work automatically and change the slider after a set time has passed.

Code

<!DOCTYPE html>

<html>

<head>

<title>Slider in CSS</title>

<style>

body{

background-color: rgb(58, 58, 58);

margin-top: 100px;

}

.carousel-container {

width: 600px;

height: 400px;

position: relative;

margin: 0 auto;

}

.navigation-buttons .previous {

position: absolute;

z-index: 10;

font-size: 25px;

top: 40%;

left: 10px;

font-weight: 700;

}

.navigation-buttons .next {

right: 10px;

position: absolute;

font-size: 25px;

z-index: 10;

top: 40%;

}

.navigation-buttons .nav-btn {

background: rgba(255, 255, 255, 0.55);

cursor: pointer;

border-radius: 50%;

width: 30px;

height: 30px;

display: flex;

justify-content: center;

align-items: center;

padding: 5px;

box-shadow: 2px 2px 10px rgba(0, 0, 0, 0.4);

}

.navigation .nav-btn:hover {

background: white;

}

.slider-carousel {

margin-top: 20px;

transition: all 0.3s ease;

}

.slider-carousel img {

width: 100%;

transition: all 0.3s ease;

border:8px solid white;

}

.images {

position: absolute;

display: none;

}

.main {

display: block;

}

.image-text {

position: absolute;

bottom: 0;

width: 103%;

display: flex;

font-size: 20px;

justify-content: center;

align-items: center;

color: rgb(255, 255, 255);

background: rgba(0, 0, 0, 0.3);

height: 35px;

}

</style>

</head>

<body>

<div id="parent-container">

<div class="navigation-buttons">

<div class="previous nav-btn"><</div>

<div class="next nav-btn">></div>

</div>

<div class="slider-carousel">

<div class="images main">

<img src="https://cdn.pixabay.com/photo/2015/04/19/08/32/marguerite-729510\_\_480.jpg" alt="flower 1" />

<div class="image-text">Image 1</div>

</div>

<div class="images">

<img src="https://cdn.pixabay.com/photo/2014/02/27/16/10/tree-276014\_960\_720.jpg" alt="flower 2" />

<div class="image-text">Image 2</div>

</div>

<div class="images">

<img src="https://cdn.pixabay.com/photo/2015/04/23/22/00/tree-736885\_960\_720.jpg" alt="flower 3" />

<div class="image-text">Image 3</div>

</div>

</div>

</div>

<script>

const previous = document.querySelector('.previous');

const next = document.querySelector('.next');

const images = document.querySelector('.slider-carousel').children;

const totalImages = images.length;

let currentIndex = 0;

// Event Listeners to previous and next buttons

previous.addEventListener('click', () => {

previousImage()

})

next.addEventListener('click', () => {

nextImage();

})

setInterval(()=>{

nextImage();

},1000);

// Function to go to next Image

function nextImage(){

images[currentIndex].classList.remove('main');

if(currentIndex == totalImages-1){

currentIndex = 0;

}

else{

currentIndex++;

}

images[currentIndex].classList.add('main');

}

// Function to go to previous Image

function previousImage(){

images[currentIndex].classList.remove('main');

if(currentIndex == 0){

currentIndex = totalImages-1;

}

else{

currentIndex--;

}

images[currentIndex].classList.add('main');

}

</script>

</body>

</html>

Output

output-slide-show-functionality

The Navigation Dots

Apart from using caption or Image text in your slider, you can also use navigation dots. Navigation dots works similarly to text or caption. They tell the user which frame they are currently accessing. Navigation dots are used in many modern sliders as they are convenient and easy to build. Navigation dots are mostly radio buttons.

A benefit of using navigation dots is that it allows the user to access any frame of the slider without any issue. Unlike the common slider where the user has to access each image one by one, navigation dots allows the user to visit any frame or image by clicking on the navigation dot corresponding to that image.

The basic principle behind the navigation dots is similar to that of slider images. The current image will have a corresponding navigation button that will have an active class. Once the image changes, the active class will be transferred or added to the current image and current navigation dot.

Code

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

\* {box-sizing: border-box}

body {font-family: Verdana, sans-serif; margin:0}

.images {display: none}

img {

vertical-align: middle;

width:100%;

}

/\* Slideshow container \*/

.carousel-container {

max-width: 1000px;

position: relative;

margin: auto;

}

/\* Next & previous buttons \*/

.previous, .next {

cursor: pointer;

position: absolute;

top: 50%;

width: auto;

padding: 16px;

margin-top: -22px;

color: white;

font-weight: bold;

font-size: 18px;

transition: 0.6s ease;

border-radius: 0 3px 3px 0;

user-select: none;

}

/\* Position the "next button" to the right \*/

.next {

right: 0;

border-radius: 3px 0 0 3px;

}

/\* On hover, add a black background color with a little bit seethrough \*/

.prev:hover, .next:hover {

background-color: rgba(0,0,0,0.8);

}

/\* The dots/bullets/indicators \*/

.navigation-dot {

cursor: pointer;

height: 15px;

width: 15px;

margin: 0 2px;

background-color: #bbb;

border-radius: 50%;

display: inline-block;

transition: background-color 0.6s ease;

}

.active, .navigation-dot:hover {

background-color: #717171;

}

</style>

</head>

<body>

<div class="carousel-container">

<div class="images fade">

<img src="https://cdn.pixabay.com/photo/2015/04/19/08/32/marguerite-729510\_\_480.jpg">

</div>

<div class="images fade">

<img src="https://cdn.pixabay.com/photo/2014/02/27/16/10/tree-276014\_960\_720.jpg">

</div>

<div class="images fade">

<img src="https://cdn.pixabay.com/photo/2015/04/23/22/00/tree-736885\_960\_720.jpg">

</div>

<!-- Previous and Next Buttons -->

<a class="previous" onclick="plusSlides(-1)">❮</a>

<a class="next" onclick="plusSlides(1)">❯</a>

</div>

<br>

<!-- Navigation Dots-->

<div style="text-align:center">

<span class="navigation-dot" onclick="currentSlide(1)"></span>

<span class="navigation-dot" onclick="currentSlide(2)"></span>

<span class="navigation-dot" onclick="currentSlide(3)"></span>

</div>

<script>

var currentIndex = 1;

//Show current image

showSlides(currentIndex);

//Function to move Next

function plusSlides(n) {

showSlides(currentIndex += n);

}

//Function to move back

function currentSlide(n) {

showSlides(currentIndex = n);

}

//Initiate moving of slides

function showSlides(n) {

var i;

var slides = document.getElementsByClassName("images");

var dots = document.getElementsByClassName("navigation-dot");

if (n > slides.length) {currentIndex = 1}

if (n < 1) {currentIndex = slides.length}

for (i = 0; i < slides.length; i++) {

slides[i].style.display = "none";

}

for (i = 0; i < dots.length; i++) {

dots[i].className = dots[i].className.replace(" active", "");

}

slides[currentIndex-1].style.display = "block";

dots[currentIndex-1].className += " active";

}

</script>

</body>

</html>

Output

example-of-navigation-dots-in-slider

Alternate Example

Sliders created using Javascript are good as long as javascript is enabled in the browser. As soon as the user turns the javascript off, the slider will no longer work since it uses javascript to achieve slider functionality.

It is not necessary to use javascript for making sliders. You can even make a slider using HTML and CSS only. All you have to figure out is how you will work when the user will click on the previous or next button. A slider created using HTML and CSS will use less browser memory and will work even if javascript is disabled in the browser.

You can use attributes and their properties to make an active class. For the navigation button, you can create an active class that works when the button or dot is clicked. This works similarly to that in Javascript, however, instead of using DOM manipulation, we will be using CSS functionality such as target, etc to achieve this.

We have created a sample slider for you using HTML and CSS. We have used pseudo-active classes such as check to determine which navigation dot the user has clicked most recently. Then corresponding to that we show the image to the user. For every image, we have set margin and padding, since we are not hiding the images.

Code

<!DOCTYPE html>

<html>

<head>

<title> Slider using HTML and CSS </title>

<style>

img{

height:100%;

width:100%;

}

#frame {

margin: 0 auto;

height:400px;

width: 600px;

max-width: 100%;

}

#frame input[type=radio] {

display: none;

}

#frame label {

cursor: pointer;

text-decoration: none;

}

#slides {

padding: 10px;

background: #00F;

position: relative;

z-index: 1;

}

#overflow {

width: 100%;

overflow: hidden;

}

#frame1:checked~#slides .inner {

margin-left: 0;

}

#frame2:checked~#slides .inner {

margin-left: -100%;

}

#frame3:checked~#slides .inner {

margin-left: -200%;

}

#slides .inner {

transition: margin-left 800ms

cubic-bezier(0.770, 0.000, 0.175, 1.000);

width: 400%;

line-height: 0;

height: 300px;

}

#slides .frame {

width: 25%;

float: left;

display: flex;

justify-content: center;

align-items: center;

height: 100%;

color: black;

}

#controls {

margin: -180px 0 0 0;

width: 100%;

height: 50px;

z-index: 3;

position: relative;

}

#controls label {

transition: opacity 0.2s ease-out;

display: none;

width: 50px;

height: 50px;

opacity: .4;

}

#controls label:hover {

opacity: 1;

}

/\* active class for images\*/

#frame1:checked~#controls label:nth-child(2),

#frame2:checked~#controls label:nth-child(3),

#frame3:checked~#controls label:nth-child(1){

float: right;

margin: 0 -50px 0 0;

display: block;

}

/\* active class for navigation button\*/

#frame1:checked~#controls label:nth-last-child(2),

#frame2:checked~#controls label:nth-last-child(3),

#frame3:checked~#controls label:nth-last-child(1){

float: left;

margin: 0 0 0 -50px;

display: block;

}

#bullets {

margin: 150px 0 0;

text-align: center;

}

#bullets label {

display: inline-block;

width: 10px;

height: 10px;

border-radius: 100%;

background: #ccc;

margin: 0 10px;

}

#frame1:checked~#bullets label:nth-child(1),

#frame2:checked~#bullets label:nth-child(2),

#frame3:checked~#bullets label:nth-child(3){

background: #444;

}

</style>

</head>

<body>

<div id="frame">

<input type="radio" name="frame" id="frame1" checked />

<input type="radio" name="frame" id="frame2" />

<input type="radio" name="frame" id="frame3" />

<div id="slides">

<div id="overflow">

<div class="inner">

<div class="frame">

<img src="https://cdn.pixabay.com/photo/2015/04/19/08/32/marguerite-729510\_\_480.jpg">

</div>

<div class="frame">

<img src="https://cdn.pixabay.com/photo/2014/02/27/16/10/tree-276014\_960\_720.jpg">

</div>

<div class="frame">

<img src="https://cdn.pixabay.com/photo/2015/04/23/22/00/tree-736885\_960\_720.jpg">

</div>

</div>

</div>

</div>

<div id="controls">

<label for="frame1"></label>

<label for="frame2"></label>

<label for="frame3"></label>

</div>

<div id="bullets">

<label for="frame1"></label>

<label for="frame2"></label>

<label for="frame3"></label>

</div>

</div>

</body>

</html>

Output

example-sliders-using-html-csss

Conclusion

Sliders are a set of frames, wherein each frame can be traversed respectively.

They are used in many modern-day websites to show reviews or testimonials.

To make a slider the user should be aware of concepts such as HTML tags, CSS concepts such as stacking, positioning, etc, and basic javascript such as variables and DOM manipulation.

Sliders work by having an active class that has its display on. The rest of the frames have their display turned off. When a user navigates via a button or dots, we change the element that will have the active class using DOM manipulation.

If the user does not have javascript enabled in the browser, then the slider will not work.

You can use just HTML and CSS to create Slider. This will save browser memory and computational time. It will also work in every browser that supports HTML and CSS.

This article is written by

R

Rahul Negi

Updated - 18 Dec 2022

15 mins read

Published : 19 Dec 2022

Scroll to top!

CSS

How would you rate this article?

Very BadBadAverageGoodLoved It!

Free Courses by top Scaler instructors

View All

Java for Beginners

Java for Beginners

A Course by Tarun Luthra

5

31446

C++ for Beginners

C++ for Beginners

A Course by Prateek Narang

5

17635

Python for Beginners

Python for Beginners

A Course by Rahul Janghu

4.90

16577

JavaScript for Beginners

JavaScript for Beginners

A Course by Mrinal Bhattacharya

4.8

10812

DBMS

DBMS

A Course by Srikanth Varma

5

10576

Python and SQL for Data Science

Python and SQL for Data Science

A Course by Srikanth Varma

5

8676

DSA Problem Solving for Interviews using Java

DSA Problem Solving for Interviews using Java

A Course by Jitender Punia

4.9

7543

DSA for Beginners using Java

DSA for Beginners using Java

A Course by Subhesh Kumar

4.95

5862

Instagram System Design

Instagram System Design

A Course by Anshuman Singh

5

4718

SQL v/s NoSQL

SQL v/s NoSQL

A Course by Mohit Yadav

4.8

4666

Video Courses

Contests

calenderEvents

Reading Tracks

topics

Got suggestions? We would love to hear your feedback.

topics logo

A Free learning platform

made with heart icon by

scaler logo

Download the App!

Playstore Icon

Explore Scaler

Academy

Data Science & ML

Neovarsity

Explore Topics

Courses

Challenges

Contest

Topics

Articles

Events

Resources

About Us

Blog

Careers

Review

Instagram

Youtube

Twitter

Facebook

Linkedin

Discord

Privacy Policy

Terms of Use

Contact Us

Copyright 2022 InterviewBit Technologies Pvt. Ltd.

All Rights Reserved.

Google Translate

Original