

Как создать слайдер изображений с помощью JavaScript, чтобы улучшить ваш сайт

05.05.2023

Приходилось ли вам сталкиваться с ситуацией, когда вы хотите отобразить несколько изображений/видео в одном разделе веб-страницы, но не хотите занимать много места? Хотите добавить интерактивности своим веб-страницам или сделать их более привлекательными? Слайдер изображений может избавить вас от этой проблемы. В этой статье мы дадим определение слайдера изображений, рассмотрим необходимые условия для создания слайдера изображений и способы его создания с помощью HTML, JavaScript и CSS.

Что такое слайдер изображений?



Слайдеры – это карусели или слайд-шоу, в которых отображаются тексты, изображения или видео. В этой статье речь пойдет о слайдерах изображений. Большинство веб-разработчиков используют слайдеры на главной странице для демонстрации рекламных предложений или наиболее важной информации.

Вот некоторые причины, по которым вы должны использовать слайдеры на своих веб-страницах:

- **Улучшение пользовательского опыта:** Идеальная веб-страница должна быть компактной, чтобы конечным пользователям не приходилось прокручивать и листать ее, чтобы получить важные данные. Если у вас есть несколько изображений, вы можете избавить пользователей от необходимости прокрутки, отобразив их в слайдере.
- **Визуальная привлекательность:** Большинство пользователей веб-сайта не станут тратить много времени на простую веб-страницу. Вы можете улучшить визуальную привлекательность с помощью слайдеров и анимации.
- **Экономия места:** Если у вас есть 20 изображений, которые вы хотите разместить на веб-странице, они могут занять много места. Создание слайдера позволит вам сэкономить место и при этом дать пользователям доступ ко всем изображениям.
- **Отображение динамического контента:** Вы можете использовать слайдеры для отображения динамического контента, такого как вставки социальных сетей, блоги и новости.

Необходимые условия для создания слайдера изображений

- **Базовое понимание HTML:** Здесь мы опишем структуру слайдера. Вам должно быть удобно работать с различными HTML-тегами, классами и div'ами. Базовое понимание CSS:

Мы будем использовать CSS для стилизации слайдеров изображений и кнопок.

- **Базовое понимание JavaScript:** Мы будем использовать JavaScript, чтобы сделать слайдеры изображений отзывчивыми.
- **Редактор кода:** Вы можете использовать редактор кода по своему усмотрению. Я буду использовать Visual Studio Code.
- Коллекция изображений.

Настройка папки проекта

Нам понадобится папка проекта, папка с изображениями в ней, а также файлы HTML, CSS и JavaScript. Я назову свой проект “Image-Slider”. Вы можете создать проект вручную или воспользоваться этими командами для начала работы:

```
mkdir Image-Slider
```

```
cd Image-Slider
```

```
mkdir Images && touch index.html styles.css script.js
```

Добавьте все изображения в папку “Images”, которую мы создали внутри папки проекта, и переходите к следующему шагу. Это папка моего проекта, в которую я добавил шесть изображений, которые я буду использовать для создания слайдера. Весь наш HTML-код будет находиться в файле index.html.

Типы слайдеров изображений

У нас может быть два типа слайдеров изображений: автоматический слайдер и автоматический слайдер изображений с кнопками.

Автоматический слайдер изображений

Автоматический слайдер автоматически прокручивается к следующему изображению через заданное время, например, через 3 секунды.

HTML-код

Вот мой HTML-код;

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="styles.css">
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15
.3/css/all.min.css" />
</head>
<body>
  <div id="slider">
    <div class="slide">
      <h1>African</h1>
      
    </div>
    <div class="slide">
      <h1>American</h1>
      
    </div>
    <div class="slide">
      <h1>Asian</h1>
      
    </div>
    <div class="slide">
      <h1>Arabic</h1>
    </div>
  </div>
</body>
</html>
```

```
        
</div>
<div class="slide">
    <h1>Modern </h1>
    
</div>
<div class="slide">
    <h1> European </h1>
    
</div>
</div>

<script src="script.js"></script>

</body>

</html>
```

Из этого кода мы можем отметить следующее:

- Я импортировал свой файл CSS в файл index.html в раздел <head>. Он будет использоваться в последующих шагах. <link rel="stylesheet" href="styles.css">
- Я добавил JavaScript в свой HTML-код непосредственно перед закрывающим тегом <body>. Я буду использовать JavaScript для добавления функциональности слайдерам.
- Каждый слайд имеет класс слайда.
- Я использовал тег для импорта изображений из папки Images.

Стиль автоматического слайдера изображений с помощью CSS

Теперь мы можем стилизовать наши изображения, поскольку мы уже связали CSS и HTML файлы.

Добавьте этот код в ваш файл CSS:

```
#slider {
```

```
    width: 80%;
}
.slide {
    width: 80%;
    display: none;
    position: relative;
}
.slide img {
    width: 80%;
    height: 80%;
}
.slide.active {
    display: block;
}
```

Из этого кода мы можем отметить следующее:

- Мы установили ширину и высоту нашего слайдера на 80%.
- Мы установили для активного слайда значение `block`, что означает, что только активный слайд будет отображаться, а остальные будут скрыты.

Как добавить JavaScript, чтобы сделать слайдер изображений отзывчивым

Добавьте этот код в файл `script.js`:

```
var slides = document.querySelectorAll('.slide');
var currentSlide = 0;
var slideInterval = setInterval(nextSlide,2000);

function nextSlide() {
    slides[currentSlide].className = 'slide';
    currentSlide = (currentSlide+1)%slides.length;
    slides[currentSlide].className = 'slide active';
}
```

Этот JavaScript делает следующее;

- Мы используем селектор `document.querySelectorAll`, чтобы

- выбрать все элементы с классом `slide`.
- Мы задаем начальное значение `currentSlide` 0.
 - Мы устанавливаем `slideInterval` равным 2000 (2 секунды), то есть изображения в слайде меняются через каждые 2 секунды.
 - Этот код `slides[currentSlide].className = 'slide';` удаляет активный класс из текущего слайда
 - Этот код `currentSlide = (currentSlide + 1) % slides.length;` увеличивает индекс текущего слайда.
 - Этот код `slides[currentSlide].className = 'slide active';` добавляет активный класс к текущему слайду.

Автоматический слайдер с кнопками

Созданный нами слайдер изображений автоматически прокручивается через каждые 2 секунды. Теперь мы можем создать изображение, в котором пользователи могут переходить к следующему слайду, нажав на кнопку, или автопрокрутку через каждые 3 секунды, если пользователь не нажимает на кнопки.

HTML-код

Вы можете изменить содержимое вашего файла `index.html` следующим образом:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <title>Image Slider</title>
    <link
      rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15
.3/css/all.min.css"
    />
    <link rel="stylesheet" href="styles.css" />
```

```

</head>
<body>
  <div class="container">

    <div class="mySlides">
      
    </div>
    <div class="mySlides">
      
    </div>
    <div class="mySlides">
      
    </div>
    <div class="mySlides">
      
    </div>
    <div class="mySlides">
      
    </div>
    <a class="prev" onclick="plusSlides(-1)"><</a>
    <a class="next" onclick="plusSlides(1)">>>/a>
    <div class="caption-container">
      <p id="caption"></p>
    </div>
    <div class="row">
      <div class="column">
        
      </div>
      <div class="column">
        
      </div>
      <div class="column">
        

```



```
        </div>
        <div class="column">
            
        </div>
        <div class="column">
            
        </div>
    </div>
</div>

    <script src="script.js"></script>
</body>
</html>
```

Этот HTML-код работает следующим образом;

- У нас есть класс `mySlides`, который содержит пять изображений.
- У нас есть две кнопки, `“prev”` и `“next”` с `onClick`, позволяющие пользователям прокручивать слайды.
- У нас есть миниатюра, которая показывает изображения в нижней части веб-страницы.

Код CSS

Добавьте это в ваш код:

```
* {
    box-sizing: border-box;
}

.container {
    position: relative;
}

.mySlides {
    justify-content: center;
}
```

```
.cursor {
  cursor: pointer;
}

.prev,
.next {
  cursor: pointer;
  position: absolute;
  top: 40%;
  width: auto;
  padding: 16px;
  margin-top: -50px;
  color: rgb(34, 143, 85);
  font-weight: bold;
  font-size: 20px;
  border-radius: 0 3px 3px 0;
  user-select: none;
  -webkit-user-select: none;
}

.next {
  right: 0;
  border-radius: 3px 0 0 3px;
}

.prev:hover,
.next:hover {
  background-color: rgba(0, 0, 0, 0.8);
}

.caption-container {
  text-align: center;
  background-color: orangered;
  padding: 2px 16px;
  color: white;
}

.row:after {
  content: "";
  display: table;
  clear: both;
}
```

```

}

.column {
  float: left;
  width: 16.66%;
}

.demo {
  opacity: 0.6;
}

.active,
.demo:hover {
  opacity: 1;
}

```

Код CSS работает следующим образом:

- Мы установили свойство `display` класса `.mySlides` как `none`, что означает, что все слайды скрыты по умолчанию.
- Мы установили непрозрачность наведенных миниатюр равной 1 с помощью правила `active, .demo:hover {opacity: 1;}`.

Код JavaScript

```

let slideIndex = 1;
showSlides(slideIndex);

function plusSlides(n) {
  showSlides(slideIndex += n);
}

function currentSlide(n) {
  showSlides(slideIndex = n);
}

let slideInterval = setInterval(() => {
  plusSlides(1);
}, 3000);

```

```

let          slideshowContainer          =
document.getElementsByClassName('slideshow-container')[0];
slideshowContainer.addEventListener('mouseenter', () => {
  clearInterval(slideInterval);
});

slideshowContainer.addEventListener('mouseleave', () => {
  slideInterval = setInterval(() => {
    plusSlides(1);
  }, 3000);
});

function showSlides(n) {
  let i;
  let slides = document.getElementsByClassName("mySlides");
  let dots = document.getElementsByClassName("demo");
  let captionText = document.getElementById("caption");
  if (n > slides.length) {slideIndex = 1}
  if (n < 1) {slideIndex = 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[slideIndex-1].style.display = "block";
  dots[slideIndex-1].className += " active";
  captionText.innerHTML = dots[slideIndex-1].alt;
}

```

Вы можете получить окончательный исходный код [здесь](#).

Наш код JavaScript делает следующее:

- У нас есть функция `currentSlide` со значением `display`, установленным как `none`. Это правило позволяет нашей веб-странице отображать только текущий слайд.
- У нас есть функция `plusSlides`, которая добавляет/вычитает заданное значение из `slideIndex`, чтобы определить, какой слайд отображать.

- Если пользователь не нажимает на кнопки, слайды будут автоматически прокручиваться через каждые 3000 миллисекунд.

Тестирование и отладка слайдера изображений

Ошибки и погрешности являются частью пути разработчика, и наличие их в вашем коде не означает, что вы плохой разработчик. Если ваш код не работает даже после выполнения описанных выше шагов, вы можете попробовать эти варианты тестирования и отладки для устранения ошибок:

- **Отлаживайте каждый файл отдельно:** Наш код состоит из трех файлов – HTML, CSS и JavaScript. Эти три языка имеют разные правила. Вы можете проверить правильность HTML, CSS и JavaScript кодов с помощью таких инструментов, как W3C Markup Validation Service для HTML, CSS Validator для CSS кода и Chrome DevTools для проверки JavaScript кода.
- **Выполняйте различные типы тестов:** Вы можете провести визуальные тесты, чтобы убедиться, что изображения отображаются правильно, тесты производительности, чтобы проверить, являются ли изображения отзывчивыми, и функциональные тесты, чтобы убедиться, что изображения являются навигационными.
- **Тестируйте код с различными форматами и размерами изображений:** Хороший слайдер изображений должен работать с различными форматами и размерами изображений.
- **Автоматизируйте свои тесты:** Автоматизация повсюду, и вы тоже можете извлечь из нее пользу при тестировании. Вы можете использовать такие инструменты, как Selenium или LoadRunner, для написания и выполнения сценариев автоматизации тестирования. Эти инструменты также позволяют повторно использовать некоторые из ваших

тестов.

- **Документируйте свои тесты:** Тестирование – это непрерывный процесс. Возможно, вам придется постоянно улучшать тесты, пока ваш код не будет работать так, как ожидается. Документируйте этот процесс, чтобы сделать ваш код читабельным и удобным для использования.

Слайдеры изображений: Лучшие практики

- **Будьте проще:** Слайдеры привлекательны. Однако количество изображений в слайдере должно быть небольшим. Идеальным вариантом будет 4-7 изображений на слайд.
- **Тестируйте свои слайдеры:** Проверьте их функциональность, прежде чем опубликовать их в Интернете.
- **Создавайте отзывчивые слайдеры:** Убедитесь, что созданные вами слайдеры реагируют на различные размеры экрана.
- **Используйте высококачественные изображения:** Возьмите/скачайте высококачественные изображения для своих слайдеров. Сохраняйте изображения в формате SVG – это отличный подход, так как они не теряют своего качества при изменении размера.
- **Измените размер изображений:** У вас могут быть разные размеры и форматы изображений. Всегда следите за тем, чтобы изображения в слайдере были одинакового размера. Этого можно добиться с помощью CSS.

Подведение итогов

Мы надеемся, что теперь вы можете создать полнофункциональный слайдер изображений для отображения важных данных на вашем сайте без использования UI-фреймворков. Вы можете использовать тот же подход для создания видеослайдеров на веб-страницах.