

Создание анимированного поля ввода HTML средствами CSS и Sass

Радионов Сергей Владимирович

Приамурский государственный университет им. Шолом-Алейхема

студент

Аннотация

В работе представлена реализация красивого, анимированного текстового поля ввода для веб-сайта, используя таблицы стилей CSS и препроцессор Sass.

Ключевые слова: сайт, ввод, поле, web, html, CSS, Sass.

Creating an animated HTML input field using CSS and Sass

Radionov Sergei Vladimirovich

Sholom-Aleichem Priamursky State University

student

Abstract

This paper presents an implementation of a beautiful, animated text input field for a website using CSS stylesheets and the Sass preprocessor.

Keyword: website, input, field, web, html, CSS, Sass.

В настоящее время очень много людей пользуются различными веб-сервисами и сайтами, которые конкурируют между собой не только в количестве и качестве функционала, но и удобстве и красоте интерфейса. Ведь пользователь выберет более красивый интерфейс, даже если функционал немного уступает. Поэтому нельзя при разработке веб-сервиса отводить качество интерфейса на второй план.

Актуальность создания анимированного поля ввода HTML средствами CSS и Sass обусловлена увеличением качества интерфейса, а именно его визуальной составляющей. Это позволит привлечь больше потенциальных пользователей.

Цель исследования: создать анимированное поле ввода HTML средствами CSS и Sass, которое будет красивым и оставлять положительный отклик у пользователя.

Для исследования способа создания анимированной кнопки веб-сайта были прочитаны различные источники. В статье С.А.Новикова рассмотрен формальный язык описания внешнего вида документа, написанного с использованием языка гипертекстовой разметки HTML, и внедрение его в HTML [1]. В современной действительности существует много способов создания сайтов, состоящих из веб-страниц. Каждый из них имеет преимущества и недостатки. В статье И.В.Смагиной проведен сравнительный

анализ при работе с гипертекстовым языком разметки документов HTML и каскадными таблицами стилей CSS [2]. В работе Г.С.Сейдаметова и Р.Р.Биктимирова рассматриваются возможности, предоставляемые CSS для создания адаптивных сеток, преимущества Grid Layout перед использованием сторонних CSS-фреймворков, модулей и других инструментов, не описанных в W3C спецификации [3]. А.Максутова в своей статье рассмотрела статистический сайт на HTML, CSS и на Bootstrap. Главная задача - показать в сложной среде, современные интерактивные документы, изучить правила их оформления и имеющиеся инструментальные средства, приобрести навыки, необходимые для успешной работы [4]. Статья Р.Р.Биктимирова, А.Б.Джемалетдинова и Р.И.Ибраимова ориентирована, прежде всего, на начинающих разработчиков веб-интерфейсов. В работе описан один из CSS препроцессоров - SASS, в частности подключение его к проекту, сравнение его с обычным CSS. Рассмотрен инструмент Compass который расширяет возможности SASS [5]. Ю.С.Гончарова в своей работе изучила применение препроцессора Sass и фреймворков при верстке адаптивного сайта [6]. Д.А.Варганов рассмотрел существующие инструменты по пост-обработке стилевых файлов с целью упрощения задач верстальщиков и фронтенд-разработчиков. Описаны основные возможности данных обработчиков. Указаны основные недостатки препроцессоров и их использования [7]. Также стоит обратить внимание и на иностранные исследования в этой области [8-9].

Перед началом разработки анимированного поля ввода необходимо представить как оно должно выглядеть. В обычном состоянии поле ввода будет иметь тонкую границу, прямые углы. На наведение курсора реагировать не будет. Название поля в обычном состоянии будет находиться внутри самого поля, а подсказка будет скрыта за полем. В активном состоянии название будет плавно перемещено в положение над полем, а подсказка плавно из-за поля переместится в положение под полем. Также во время перемещения подсказки размер шрифта немного увеличится, что создаст красивый эффект (Рис. 2).

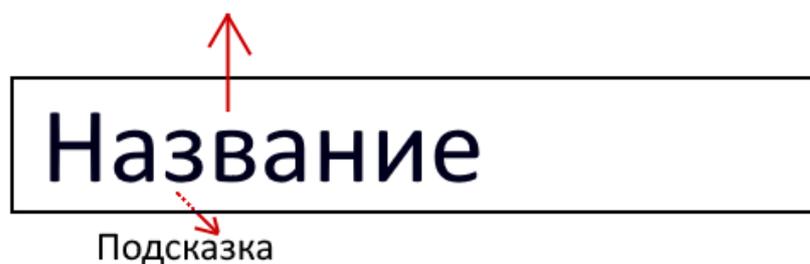


Рис. 2. Схема анимации название и подсказки

После представления конечного вида разрабатываемого поля ввода можно приступать к реализации задуманного. Сначала в html необходимо создать основной блок, где будет располагаться поле ввода, название, подсказка, а также границы, для которых позже будет задана анимация

(Рис.3). Необходимо разместить этот блок в тегах «body» для того, чтобы можно было смотреть на результат при изменении CSS стилей.

```
<body>
  <div class="input-effect">
    <input type="text" placeholder=" ">
    <label>Название</label>
    <span class="helper">Подсказка</span>
  </div>
</body>
```

Рис. 3. Блок поля ввода

Необходимо создать файл «style.scss» и настроить препроцессор Sass. В этом файле для начала нужно описать цвета и шрифт, которые будем использовать (Рис.4).

```
:root{
  --main-brd-color: #5d5d5d;
  --brd-focus-color: #168979;
  --input-bg-color: #fff;
  --error-color: #cd0022;

  font-family: 'Ubuntu Condensed', sans-serif;
}
```

Рис. 4. Описание цветов и шрифта

Далее нужно описать класс «input-effect» (Рис. 5), где:

1. Time – время анимации
2. Позиция адаптивная, выравнивание по левому краю
3. Выставлены нужная ширина и высота
4. Выставлены отступы, достаточные для размещения в них названия и подсказки

```
div.input-effect {

  $time: 0.4s;
  position: relative;
  float: left;
  width: 250px;
  height: 30px;
  margin: 22px 0 20px 0;
```

Рис. 5. Описание класса «input-effect»

Далее идет настройка самого поля ввода «input», которое располагается внутри предыдущего блока (Рис.6), где:

1. Позиция абсолютная
2. Расположение в левом верхнем углу родителя
3. Размер на всю ширину родителя

4. Граница толщиной 1 пиксель
5. Выравнивание текста по центру
6. Выставлены нужные цвета
7. Выставлено время анимации

```
input {
  z-index: 1;
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  border: 1px solid var(--main-brd-color);
  font-size: 20px;
  height: 100%;
  line-height: 100%;
  text-align: center;
  outline: none;
  background-color: var(--input-bg-color);
  color: var(--main-brd-color);
  transition: $time;
}
```

Рис. 6. Описание «input»

К полю ввода можно добавить классы «text-left», который выравнивает вводимый текст по левому краю (Рис. 7), а также класс «invalid» который окрашивает границы, название и подсказку в красный цвет (Рис. 8).

```
&.text-left {
  text-align: left;
  padding-left: 10px;
}
```

Рис. 7. Описание класса «text-left»

```
&.invalid {
  & ~ label, & ~ span.helper {
    color: var(--error-color);
  }
}
```

Рис. 8 Описание класса «invalid»

После этого был описан тег «label» (Рис. 9), где:

1. Расположение выше, чем «input»
2. Позиция абсолютная
3. Выставлено необходимое расположение относительно поля ввода

4. Отключены все взаимодействие с мышкой, объект полностью прозрачен для кликов.
5. Описан класс, располагающий название по центру поля ввода

```
label {
  z-index: 11;
  position: absolute;
  top: 5px;
  left: 10px;
  font-size: 20px;
  transition: $time;
  pointer-events: none;

  &.center {
    transform: translate(-50%, 0);
    left: 50%;
  }
}
```

Рис. 9. Описание «label»

Также были описаны стили подсказки (Рис. 10), которые включают:

1. Расположение под полем ввода
2. Позиционирование абсолютное
3. Выставлено нужное расположение
4. Выставлена прозрачность 10%
5. Описан класс, располагающий подсказку по центру

```
span.helper {
  z-index: 0;
  position: absolute;
  bottom: 0;
  left: 10px;
  font-size: 12px;
  transition: $time;
  color: var(--main-brd-color);
  opacity: .1;

  &.center {
    transform: translate(-50%, 0);
    left: 50%;
  }
}
```

Рис. 11. Описание подсказки

Теперь осталось настроить описание стилей для анимации. Анимация должна переместить название вверх на 25 пикселей, изменить его цвет на зеленый, закрасить границы ввода зеленым цветом, переместить подсказку вниз на 25 пикселей, увеличить её шрифт до 16 пикселей, сделать

непрозрачной. Все это было описано (Рис. 12). Анимация активируется псевдоклассом «focus» поля ввода. Также положение всех элементов кроме подсказки остается в состоянии «focus» пока не отображается placeholder поля, т.е. пока поле содержит какое-то значение.

```
&:focus, &:not(:placeholder-shown) {  
  border: 2px solid var(--brd-focus-color);  
  
  & ~ label {  
    top: -25px;  
  }  
  
  &:not(.invalid) ~ label {  
    color: var(--brd-focus-color);  
  }  
}  
  
&:focus {  
  & ~ span.helper {  
    bottom: -25px;  
    font-size: 16px;  
    opacity: 1;  
  }  
}
```

Рис. 12. Описание состояния «focus»

Разработанный input по умолчанию отображено на рисунке 13.

Название

Рис. 13. Поле ввода

Поле в состоянии «focus» данными выглядит как на рисунке 14.

Название

Подсказка

Рис. 14. Поле в состоянии «focus»

На рисунке 15 показано, как выглядит поле с введенными данными.

Название

данные

Рис. 15. Поле с введенными данными

При добавлении полю класса «invalid» поле выглядит как на рисунке 16.

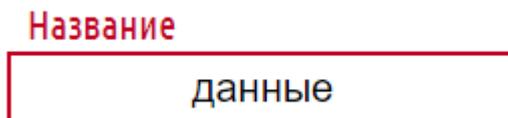


Рис. 16. Поле ввода с классом «invalid»

Таким образом, было разработано красивое, анимированное поле ввода для использования в веб-сайтах.

Библиографический список

1. Новиков С.А. CSS-каскадные таблицы стилей. Внедрение CSS в html // В сборнике: Информационные технологии и прикладная математика Всероссийский межвузовский сборник аспирантских и студенческих научных работ. 2017. С. 121-124.
2. Смагина И.В., Романова О.А. Сравнительная характеристика приемов работы с html и css при создании веб-документов // Научные Записки Орел ГИЭТ. 2012. № 1 (5). С. 444-447.
3. Сейдаметов Г.С., Биктимиров Р.Р. Использование css grid layout при построении сеток сайта // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2018. № 3 (21). С. 51-58.
4. МаксUTOва А. Статистический сайт на html, css и на bootstrap // Вестник КГУСТА. 2017. № 3 (57). С. 50-53.
5. Биктимиров Р.Р., Джемалетдинов А.Б., Ибраимов Р.И. Sass - простое написание стилей // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2017. № 1 (15). С. 137-143.
6. Гончарова Ю.С. Применение препроцессора Sass и фреймворков при верстке адаптивного сайта // В сборнике: Молодость. Интеллект. Инициатива Материалы V Международной научно-практической конференции студентов и магистрантов. 2017. С. 17.
7. Варганов Д.А. Обзор постпроцессоров css и их задач // В сборнике: Альманах научных работ молодых ученых университета ИТМО XLVII научная и учебно-методическая конференция Университета ИТМО по тематикам: экономика; менеджмент, инноватика. 2018. С. 82-83.
8. Duckett J. HTML & CSS: design and build websites. Indianapolis, IN : Wiley, 2011. Т. 15.
9. Catlin H., Catlin M. L. Pragmatic guide to Sass. Pragmatic Bookshelf, 2011.