

Новые возможности веб-форм

3

Все разработчики, занимавшиеся созданием сложных пользовательских интерфейсов, отлично знают, что возможности элементов форм HTML сильно ограничены. Текстовые поля, меню, переключатели, флажки — да еще неуклюжие списки с множественным выделением, работу с которыми приходится дополнительно разъяснять пользователям («Удерживая клавишу Ctrl, щелкайте на нужных строках, а если вы работаете на Mac, то используйте клавишу Cmd»).

И тогда вы делаете то, что делают все нормальные люди, — используете jQuery UI¹ или реализуете собственные элементы управления или функции в виде комбинации HTML, CSS и JavaScript. А потом, взглянув на форму с множеством ползунков, календарей, счетчиков, полей с заполняющим текстом и визуальных редакторов, вы быстро понимаете, что создали сущий кошмар для себя. Вам придется следить за тем, чтобы элементы управления, добавленные на страницу, не конфликтовали с другими элементами управления или библиотеками JavaScript на странице. Вы можете потратить несколько часов на собственную реализацию календаря, а потом обнаружить, что она не работает, потому что плагин был написан недостаточно аккуратно и несовместим с последней версией jQuery, используемой в проекте.

Улыбнулись? Наверное, потому, что описанная ситуация вам хорошо знакома. Нахмурились? Видимо, по той же причине. Однако не стоит отчаиваться. В этой главе мы построим пару веб-форм с полями новых типов, а также реализуем автофокус и заполняющий текст. Также будут рассмотрены простые средства проверки данных на стороне клиента. В за-

¹ <http://jqueryui.com/>

вершение мы рассмотрим, как с помощью нового атрибута `contenteditable` преобразовать любое поле HTML в редактируемый элемент.

Конкретно в этой главе рассматриваются следующие возможности:

`<input type="email">`

Поле для ввода адреса электронной почты. [O10.1, IOS, A3]

`<input type="url">`

Поле для ввода URL-адреса. [O10.1, IOS5, A3]

`<input type="range">`

Ползунок. [C5, S4, F23, IE10, O10.1]

`<input type="number">`

Поле для ввода числовых данных, часто в виде элемента-счетчика. [C5, S5, O10.1, IOS5, A3]

`<input type="color">`

Поле для выбора цвета. [C5, O11]

`<input type="date">`

Поле для ввода даты. Поддерживаются типы `date`, `month` и `week`. [C5, S5, O10.1]

`<input type="datetime">`

Поле для ввода даты со временем. Поддерживаются типы `datetime`, `datetime-local` и `time`. [S5, O10.1]

`<input type="search">`

Поле для ввода ключевых слов поиска. [C5, S4, O10.1, IOS]

`<input type="text" autofocus>`

Поддержка передачи фокуса конкретному элементу формы. [C5, S4]

`<input type="email" placeholder="me@example.com">`

Поддержка автоматического включения текста в поле формы. [C5, F4, S4]

`<input type="email" required>`

Блокировка отправки страницы, если поле осталось незаполненным. [C23, F16, IE10, O12]

`<input pattern=" /^(\\s*|\\d+)$ /">`

Блокировка отправки страницы, если содержимое поля не соответствует заданному образцу. [C23, F16, IE10, O12]

`<p contenteditable>lorem ipsum</p>`

Поддержка редактирования контента «на месте» в браузере. [C4, F3.5, S3.2, IE6, O10.1, iOS5, A3]

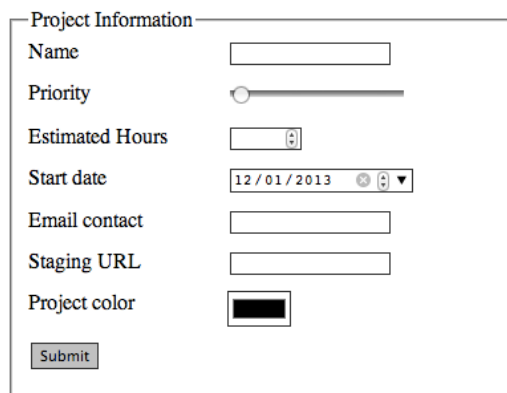
Итак, для начала познакомимся поближе с некоторыми из этих исключительно полезных типов полей.

Рецепт 5. Описание данных при помощи новых полей

В HTML5 появились новые типы полей форм, позволяющие более точно описать вводимые пользователем данные. В дополнение к стандартным элементам (текстовые поля, переключатели и флажки), формы также могут содержать такие элементы, как поля адресов электронной почты, календари, палитры для выбора цвета, счетчики (spinboxes) и ползунки. Браузеры сами отображают расширенные элементы, а разработчик избавляется от необходимости использовать JavaScript. На мобильных устройствах, а также виртуальных клавиатурах планшетных и сенсорных устройств тип поля может использоваться для выбора раскладки клавиатуры. Например, браузер Safari для iOS при вводе данных в полях URL и email отображает альтернативную раскладку с удобным вводом специальных символов @, ., : и /.

Компания AwesomeCo работает над новым веб-приложением, предназначенным для управления проектами. В этом приложении разработчики и руководители смогут отслеживать ход работы по разным проектам, которыми они занимаются. Для каждого проекта указано название, контактный адрес электронной почты и URL-адрес, по которому руководители могут просмотреть предварительную версию создаваемого приложения. Также на форме должны отображаться поля для начальной даты, приоритета и приблизительного количества рабочих часов до завершения проекта. Наконец, начальник отдела разработки хочет назначить каждому проекту определенный цвет, чтобы ему было проще опознавать проекты при просмотре отчетов.

Давайте построим макет страницы описания проекта с использованием новых полей форм HTML5. В конечном итоге форма должна выглядеть примерно так.



The image shows a form titled "Project Information" with the following fields and controls:

- Name:** A standard text input field.
- Priority:** A range slider control.
- Estimated Hours:** A spinbox control with up and down arrows.
- Start date:** A date picker control showing "12/01/2013" with a calendar icon and a dropdown arrow.
- Email contact:** A text input field.
- Staging URL:** A text input field.
- Project color:** A color picker control showing a black color swatch.
- Submit:** A button at the bottom left of the form.

Описание формы

Давайте создадим новую страницу HTML5 с базовой формой HTML, которая осуществляет отправку данных методом POST. Предполагается, что где-то существует страница, которая эту форму обрабатывает. Поле названия проекта ничем особенным не отличается, поэтому мы воспользуемся старым и проверенным типом `text`.

html5forms/index.html

```
<form method="post" action="/projects/1">

  <fieldset id="personal_information">
    <legend>Project Information</legend>
    <ol>
      <li>
        <label for="name">Name</label>
        <input type="text" name="name" id="name">
      </li>
      <li>
        <input type="submit" value="Submit">
      </li>
    </ol>

  </fieldset>

</form>
```

Обратите внимание: при создании разметки формы в упорядоченный список включаются метки (элементы `<label>`). Они играют важную роль в обеспечении доступности форм. Атрибут `for` ссылается на идентификатор ассоциированного элемента формы; это помогает экранным дикторам идентифицировать поля на странице. Упорядоченный список является хорошим средством перечисления полей без использования сложных таблиц или структур `div`. Кроме того, он позволяет задать порядок, в котором, по вашему мнению, должны заполняться поля. После того как форма будет построена, мы воспользуемся правилами CSS для упорядочения ее элементов.

Создание ползунка

Ползунки (sliders) используются для увеличения или уменьшения числовых значений. В нашем приложении этот элемент хорошо подойдет для

визуального представления и изменения приоритета проекта. Ползунок реализуется полем типа `range`.

html5_forms/index.html

```
<label for="priority">Priority</label>
<input type="range" min="0" max="10"
  name="priority" value="0" id="priority">
```

Добавьте его на форму в новом элементе `li` (по аналогии с предыдущим полем). В Chrome, Safari и Opera реализован виджет Slider, который выглядит так.



Также обратите внимание на определение нижней и верхней границ диапазона `min` и `max`. Они ограничивают значения, которые могут храниться в поле формы.

Счетчики и работа с числовыми данными

Мы часто работаем с числовыми данными. И хотя ввести число с клавиатуры несложно, счетчики (spinboxes) удобны для незначительного изменения числовых данных. Счетчик представляет собой поле, справа от которого находятся две кнопки со стрелками — для увеличения и уменьшения хранимого значения.

На нашей форме в поле счетчика будет храниться приблизительный объем работы в часах. Помимо упрощения ввода данных, использование этого поля четко описывает тип содержащихся в нем данных.

html5_forms/index.html

```
<label for="estimated_hours">Estimated Hours</label>
<input type="number" name="estimated_hours"
  min="0" max="1000"
  id="estimated_hours">
```

В Chrome, Safari и Opera поддерживается элемент-счетчик, который выглядит так.



В поле счетчика по умолчанию разрешен прямой ввод с клавиатуры. Как и в случае с ползунками, для поля можно задать минимальное и максимальное значение. Однако эти значения не распространяются на значения, непосредственно введенные в поле, так что ввод придется ограничить при помощи сценариев или средствами проверки данных HTML5, которые будут рассматриваться в рецепте 8.

Также обратите внимание на возможность управления приращением, которое определяется параметром `step`. По умолчанию приращение равно 1, но ему также можно присвоить любое числовое значение.

Дата

Дата начала работы над проектом весьма важна, поэтому операции с ней должны быть по возможности удобными. Для хранения даты идеально подходит поле типа `date`.

html5_forms/index.html

```
<label for="start_date">Start date</label>
<input type="date" name="start_date" id="start_date"
  value="2013-12-01">
```

Поле `date` четко описывает тип содержащихся в нем данных, а при правильной реализации пользователь увидит календарный виджет наподобие того, который отображается в Chrome.

The screenshot shows a web form with the following fields: 'Start date', 'Email contact', 'Staging URL', and 'Project color'. A 'Submit' button is located below the 'Project color' field. The 'Start date' field is active, showing a date picker widget. The widget has a text input at the top containing '12/01/2013' and a dropdown menu below it showing 'December 2013'. Below the dropdown is a calendar grid with the following structure:

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

На момент написания книги среди браузеров для настольных систем полноценный календарный элемент для выбора даты поддерживался только в Chrome и Opera, но некоторые мобильные браузеры используют это поле для упрощенного ввода месяца, даты и года. Все остальные браузеры просто отображают текстовое поле.

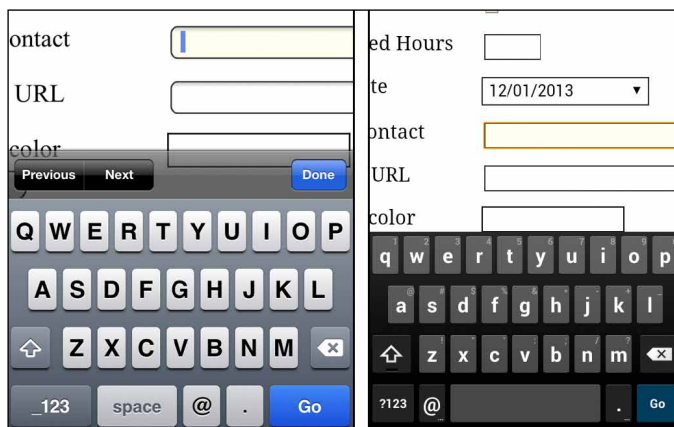
Адрес электронной почты

Поле типа `email` предназначено для хранения либо отдельного адреса, либо списка адресов электронной почты. Таким образом, это поле идеально подходит для нашего приложения.

```
html5_forms/index.html
```

```
<label for="email">Email contact</label>
<input type="email" name="email" id="email">
```

Поля этого типа приносят особенно заметную пользу на мобильных устройствах, на которых раскладка виртуальной клавиатуры изменяется для удобства ввода адресов электронной почты. Обратите внимание на присутствие символа `@` в клавиатурах iPhone и Android.



URL-адрес

На форме из нашего примера должен размещаться URL-адрес для просмотра предварительной версии проекта. Этот тип поля также адаптирован для ввода URL-адресов. Чтобы включить в форму поле для URL-адреса, достаточно добавить в нее следующий фрагмент:

```
html5forms/index.html
```

```
<label for="url">Staging URL</label>
<input type="url" name="url" id="url">
```

Этот тип также особенно удобен, если ваши посетители просматривают сайт с устройств iPhone или Android — устройство отображает совершенно

иную раскладку клавиатуры с кнопками для ускоренного ввода веб-адресов (по аналогии с клавиатурой, отображаемой при вводе URL-адреса в адресной строке мобильного браузера).

Цвет

Наконец, пользователю необходимо дать возможность выбрать цветовой код проекта. Для этой цели будет использовано поле типа `color`.

html5forms/index.html

```
<label for="project_color">Project color</label>
<input type="color" name="project_color" id="project_color">
```

На момент написания книги лишь немногие браузеры отображали элемент выбора цвета, но это не препятствовало использованию полей в страницах. Использование правильной разметки для описания контента пригодится в будущем, особенно если потребуется предоставить обходную поддержку.

Применение стилей к форме

Мы применим к форме простейшее оформление CSS. Создайте новый файл с именем `stylesheets/style.css` и включите ссылку на него в секцию `<head>` страницы формы.

html5_forms/index.html

```
<link rel="stylesheet" href="stylesheets/style.css">
```

Для начала уберем нумерацию, поля и отступы из списка:

html5_forms/stylesheets/style.css

```
ol{
  list-style: none;
  margin: 0;
  padding :0;
}
ol li{
  clear: both;
  margin: 0 0 10px 0;
  padding: 0;
}
```

Затем выровняем метки и поля ввода, а также слегка изменим оформление полей ввода.

html5_forms/stylesheets/style.css

```
label{
  float: left;
  width: 150px;
}

input{ border: 1px solid #333; }

input:focus{ background-color: #ffe; }
```

Псевдокласс `:focus` позволяет применить стилевое оформление только к тому полю, которое в настоящий момент имеет фокус ввода.

Мы построили вполне презентабельную и семантически правильную веб-форму. Можно пойти еще дальше и идентифицировать каждый из типов полей средствами CSS.

Chrome и Opera поддерживают большинство новых элементов, но при открытии страницы в Firefox, Safari или Internet Explorer многие поля отображаются как обычные текстовые поля.

Обходное решение

Браузеры, в которых новые типы полей не реализованы, просто заменяют их текстовыми полями, чтобы с формой по крайней мере можно было работать. На этой стадии можно принять решение об использовании другого виджета или библиотеки. Например, вы проверяете, поддерживает ли браузер элемент календаря, и если не поддерживает — добавляете его средствами jQuery UI. Со временем полноценные элементы будут поддерживаться всеми основными браузерами, и тогда «заплатки» можно будет удалить. Процесс выглядит практически одинаково для всех элементов управления, для которых необходимо создать обходное решение.

Замена элемента выбора цвета

Элемент выбора цвета легко идентифицируется и заменяется при помощи jQuery и плагина `jQuery-simple-color`¹. Мы ищем любое поле `<input>` с типом `color` и применяем плагин jQuery:

```
$('.input[type=color]').simpleColor();
```

Так как в разметке используются новые типы форм, добавлять имя класса или другую разметку для идентификации элементов выбора цвета не нужно. Селекторы атрибутов хорошо сочетаются с HTML5.

¹ <http://recursive-design.com/projects/jquery-simple-color/>

Обычно при использовании таких плагинов, как SimpleColor, мы загружаем плагин, присоединяем его к странице в теге `<script>`, а затем присоединяем его к элементу при помощи отдельного фрагмента JavaScript. Но если браузер имеет встроенную поддержку элемента выбора цвета, то использовать плагин не нужно, поэтому мы включим фрагмент кода JavaScript для проверки того, поддерживает ли браузер поле `input` с типом `color`. Если поле не поддерживается, плагин загружается динамически.

Для начала, поскольку нам понадобится библиотека jQuery, мы загружаем ее непосредственно перед закрывающим тегом `<body>`:

```
html5_forms/index.html
```

```
<script
  src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1
      /jquery.min.js">
</script>
```

Затем создается новый файл с именем `javascripts/fallbacks.js`, который включается в конец страницы HTML тегом `<script>`;

```
html5_forms/index.html
```

```
<script src="javascripts/fallbacks.js"></script>
```

В файле `javascripts/fallbacks.js` создается функция, проверяющая поддержку `color` в браузере:

```
html5_forms/javascripts/fallbacks.js
```

```
1 function hasColorSupport(){
  - element = document.createElement("input");
  - element.setAttribute("type", "color");
  - var hasColorType = (input.type === "color");
5  // Проверка частичной реализации
  - if(hasColorType){
  -   var testString = "foo";
  -   element.value = testString;
  -   hasColorType = (input.value !== testString);
10 }
  - return(hasColorType);
  - }
```

Сначала мы создаем элемент и присваиваем его атрибуту `type` значение `color`. Затем чтение атрибута `type` проверяет, разрешил ли браузер задать значение атрибута. Если чтение возвращает значение `color`, значит, этот тип элемента поддерживается.