

УДК 004.4

РЕАЛИЗАЦИЯ ЭЛЕМЕНТА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА SLIDER С ПОМОЩЬЮ JAVASCRIPT

Белов В.Н., Ковалёв А.И., Новиков С.А.*Арзамасский филиал ННГУ им. Н.И. Лобачевского, Арзамас, e-mail: bwn.arz@list.ru*

В данной статье будет рассмотрен один из самых популярных и часто используемых Java скриптов – Slider. Говоря другими словами – это обычный ползунок, используемый для выставления нужных значений или общего уровня каких-либо значений. Данный плагин получил огромное распространение в сети, благодаря своей относительной простоте и большому функционалу. Плагин Slider позволяет превращать блочные элементы, такие как div`ы, в элементы управления, которые называются ползунками. Установить нужный диапазон значений, изменение громкости, ценовые рамки, все это чаще всего реализуется с помощью Slider`а. Передвигать кнопки ползунков можно с помощью мыши, или, если ползунок находится в фокусе, с помощью стрелок на клавиатуре. Будут рассмотрены все особенности внедрения и инициализации плагина, основные положительные стороны и существующие на данный момент минусы. Для использования данного материала требуется минимальное знание JavaScript и HTML.

Ключевые слова: ползунок, плагин, скрипт

IMPLEMENTATION UL ELEMENT SLIDER THROUGH JAVASCRIPT

Belov V.N., Kovalev A.I., Novikov S.A.*Arzamas branch of Lobachevsky State University of Nizhni Novgorod, Arzamas, e-mail: bwn.arz@list.ru*

In this article one of the most popular and often used Java of scripts – Slider will be considered. Speaking in other words - it is the normal slider used for exposure of the necessary values or the overall level of any values. This plug-in gained huge distribution to networks, thanks to the relative simplicity and a big functionality. The plug-in of Slider allows to turn block elements, such as div`y, into controls which are called a romper suit. To set the necessary value range, volume change, price frames, all this most often is implemented by means of Slider`a. It is possible to move buttons of sliders using a mouse, or if the slider to be in focus, by means of arrows on the keypad. All features of implementation and initialization of a plug-in, the main positive sides and minuses existing at the moment will be considered. Use of this material requires the minimum knowledge of JavaScript and HTML.

Keywords: slider, plugin, script

Рассмотрен пример реализации элемента пользовательского интерфейса «Slider». Реализация будет проходить на прототипно-ориентированном языке программирования JavaScript. Перед нами стояла задача разместить на странице произвольное количество slider`ов и контролировать их состояние (получать или задавать определенные значения). Также нужна была возможность изменять внешний вид элементов ползунка(самой шкалы и рукояти). Отличительной важностью обладало соблюдение кроссбраузерности.

Кроссбраузерность – свойство сайта отображаться и работать во всех современных браузерах идентично. Под идентичностью понимается отсутствие развалов верстки и возможность отображать материал с одинаковой степенью читабельности. Понятие «кроссбраузерность» часто путают с пиксельным соответствием, что на самом деле является разными понятиями. Так как веб-технологии все время развиваются, приемлемую кроссбраузерность можно обеспечить только для последних версий современных браузеров [7].

От себя хочется добавить, что предпочтительнее всего писать плагин изначально

но для Internet Explorera, в связи с тем, что большая часть пользователей пользуется именно данным браузером. И лишь после, когда плагин корректно отображается в IE, подгонять его под остальные современные популярные браузеры, например Opera, Mozilla Firefox, Google Chrome, US Browser, Safari и другие. Это мера предпринимается для уменьшения количества случаев невозможности отображения и использования скрипта пользователям [3].

Так как стандартный ползунок, реализуемый на html, отличается относительной простотой и не отвечает нужным требованиям, поставленным перед нами, было решено создавать slider на JavaScript [4].

Материалы и методы исследования

Перед тем, как приступить к реализации самого скрипта, нужно подготовить css классы ползунка:

```
.slider {  
    background-image: url(gifka.gif);  
    background-repeat: repeat-x;  
}  
.knob {  
    position: relative;  
    background-image: url(gifka2.gif);  
}
```

Этим самым мы определяем фоны и позицию кнопки на самом ползунке. Установив нужные значения, мы определили фоновую картинку и повторение ее только по горизонтали. Располагаться кнопка на slider'е будет относительно исходного места. Далее в теле нашего html документа нужно определить блочные элементы для размещения slider'ов:

```
<div id=»sl»></div>
<div id=»sl2»></div>
```

Результаты исследования и их обсуждения

Чтобы не описывать каждый раз наш стиль внутри тега, выделяем стиль во внешнюю таблицу стилей, а к тегу при этом добавляем атрибут id с именем селектора. И теперь к самому интересному – наш JavaScript класс slider'a:

```
function slider(elemId, sliderWidth, range1, range2, step) {
    var knobWidth = 17;           // ширина и высота бегунка
    var knobHeight = 21;          // изменяются в зависимости от используе-
    мых изображений
    var sliderHeight = 21;        // высота slider'a
    var offsX,tmp;                // вспомогательные переменные
    var d = document;
    var isIE = d.all || window.opera; // определяем модель DOM
    var point = (sliderWidth-knobWidth-3)/(range2-range1);
    // point - количество пикселей на единицу значения

    var slider = d.createElement('DIV'); // создаем slider
    slider.id = elemId + '_slider';
    slider.className = 'slider';
    d.getElementById(elemId).appendChild(slider);

    var knob = d.createElement('DIV'); // создаем ползунок
    knob.id = elemId + '_knob';
    knob.className = 'knob';
    slider.appendChild(knob);          // добавляем его в документ

    knob.style.left = 0;               // бегунок в нулевое значение
    knob.style.width = knobWidth+'px';
    knob.style.height = knobHeight+'px';
    slider.style.width = sliderWidth+'px';
    slider.style.height = sliderHeight+'px';

    var sliderOffset = slider.offsetLeft; // sliderOffset – абсолютное сме-
    щение slider'a
    tmp = slider.offsetParent;          // от левого края в пикселях (в IE
    не работает)
    while(tmp.tagName != 'BODY') {
        sliderOffset += tmp.offsetLeft; // тут его и находим
        tmp = tmp.offsetParent;
    }

    if(isIE)                            // в зависимости от модели DOM
    {                                    // / /
        knob.onmousedown = startCoord;
        slider.onclick = sliderClick;
        knob.onmouseup = endCoord;
        slider.onmouseup = endCoord;
    }
    else {
        knob.addEventListener(«mousedown», startCoord, true);
        slider.addEventListener(«click», sliderClick, true);
        knob.addEventListener(«mouseup», endCoord, true);
        slider.addEventListener(«mouseup», endCoord, true);
    }
}
```

```

// функции установки/получения значения
function setValue(x) // установка по пикселям
{
    if(x < 0) knob.style.left = 0;
    else if(x > sliderWidth-knobWidth-3) knob.style.left = (sliderWidth-3-
knobWidth)+'px';
    else {
        if(step == 0) knob.style.left = x+'px';
        else knob.style.left = Math.round(x/(step*point))*step*point+'px';
    }
    d.getElementById('info').value = getValue(); // это вывод значения
}
для примера
function setValue2(x) // установка по значению
{
    if(x < range1 || x > range2) alert('Value is not included into a slider range!');
    else setValue((x-range1)*point);

    d.getElementById('info').value = getValue();
}
function getValue()
{return Math.round(parseInt(knob.style.left)/point)+range1;}

//слушатели событий
function sliderClick(e) {
    var x;
    if(isIE) {
        if(event.srcElement != slider) return; //IE onclick bug
        x = event.offsetX - Math.round(knobWidth/2);
    }
    else x = e.pageX-sliderOffset-knobWidth/2;
    setValue(x);
}

function startCoord(e) {
    if(isIE) {
        offsX = event.clientX - parseInt(knob.style.left);
        slider.onmousemove = mov;
    }
    else {
        slider.addEventListener(«mousemove», mov, true);
    }
}

function mov(e) {
    var x;
    if(isIE) x = event.clientX-offsX;
    else x = e.pageX-sliderOffset-knobWidth/2;
    setValue(x);
}

function endCoord() {
    if(isIE) slider.onmousemove = null;
    else slider.removeEventListener(«mousemove», mov, true);
}

// объявляем функции setValue2 и getValue как методы класса
this.setValue = setValue2;
this.getValue = getValue;
}

// конец класса

var mysl1 = new slider('sl', 300, 0, 200, 20);
var mysl2 = new slider('sl2', 400, 100, 200, 0);

```

Параметры данного класса:

- `elemId` – ID элемент, внутри которого размещен ползунок.
- `sliderWidth` – длина ползунка, измеряемая в пикселях.
- `max` – максимальное значение элемента, соответствующее крайнему правому положению ползунка.

• `min` – минимальное значение элемента, соответствующее крайнему левому положению ползунка.

• `range1`, `range2` – нижняя и верхняя границы значений ползунка.

• `step` – шаг изменения значения ползунка (0 – единичный шаг).

В конечном результате должны получиться вот такие `slider` ы:

`Slider` длиной 300px, диапазон значений от 0 до 200, шаг 20:



`Slider` длиной 400px, диапазон значений от 100 до 200, шаг 0 (1):

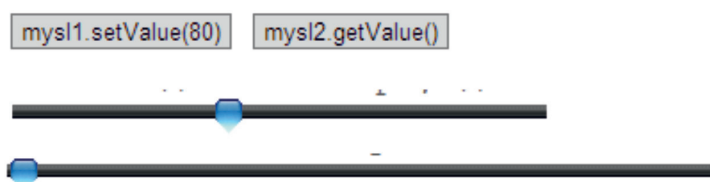


В ползунке может быть любое количество рукояток. Если их больше одной, то вместо поля `value` нужно использовать `values`, который будет хранить массив значений (по одному на каждую рукоятку) [6].

Для того чтобы устанавливать и получать значения, нужно вписать:

```
mysl1.setValue(80); // устанавливаемое
                       значение должно попадать в диапазон slider'a
mysl2.getValue();
```

После добавления этих строк появятся 2 кнопки, при нажатии на которые на нужном нам слайдере будет выставляться значение, равное тому, что было указано:



Также стоит отметить, что существует небольшая проблема. Обязательно нужно соблюдать пропорции между длиной, диапазоном и шагом `slider`'а, дабы избежать неадекватности нашего плагина [5].

Заключение

В итоге с помощью вышеописанного общедоступного кода у нас получается функционирующий ползунок, значения, размеры и диапазоны которого можно подстраивать под конкретную задачу. Реализация занимает минимум времени и усилий, поэтому это отличный вариант для начинающих веб-разработчиков. Код можно не только подстроить под себя, но и улучшить, добавив дополнительные значения, которые будут подходить под конкретные задачи и выполнять определенные функции, которые не были указаны в данной статье [1, 2].

Список литературы

1. Белов В.Н., Ковалев А.И. Математический вестник педвузов и университетов Волго-Вятского региона // Некоторые аспекты использования электронных ключей в подходе защиты информации. – Киров, 2015. – С. 318–325.
2. Напалков С.В., Сазанов А.А., Широков Л.В. Web-комплексы и их приложения // Web-технологии в образовательном пространстве: проблемы, подходы, перспективы, сборник статей участников Международной научно-практической конференции. Под общей редакцией С.В. Арюткиной, С.В. Напалкова. – 2015. – С. 125–130.
3. Трухманов В.Б. Математические модели в экономике и их анализ с помощью компьютерных средств // В сборнике: Педагогические технологии математического творчества сборник статей участников международной научно-практической конференции. Редакционная коллегия: под общей редакцией М.И. Зайкина, С.В. Арюткина (ответственный редактор), С.В. Напалков, Т.В. Романова. – 2011. – С. 338–342.
4. Хабрахабр – Начинаящим Java программистам [Электронный ресурс]. – Основные Java пакеты // URL <https://habrahabr.ru/post/43293/> (дата обращения: 28.06.2016).
5. Широков Л.В., Яблонский Д.В., Глухова А.Ф. Дифференциальные уравнения и уравнения в частных производных // Учебное пособие для студентов вузов. Государственное образовательное учреждение высшего

профессионального образования «Арзамасский государственный педагогический институт им. А.П. Гайдара». – Арзамас, 2006.

6. BestJavaScriptsCodes – JavaScripts Codes [Электронный ресурс]. – Stylish Button // URL http://www.javascriptbestcodes.com/Stylish_Button.htm (дата обращения: 29.06.2016).

7. CSS для Чайников – Руководство по CSS [Электронный ресурс]. – Селекторы и свойства. // URL <http://technologyweb.org/> (дата обращения: 29.06.2016).

8. CSS Tutotial – W3Schools [Электронный ресурс]. – Examples in Each Chapter. //URL <http://www.w3schools.com/css/> (дата обращения: 26.06.2016).

9. EasyWebSkrpts.net – Веб-программирование [Электронный ресурс]. – JavaScript Slider //URL: <http://easywebscripts.net/javascript/slider.php> (дата обращения: 27.06.2016).

10. Htmlbook – Самоучитель по HTML [Электронный ресурс]. – Ползунок //URL: <http://htmlbook.ru/samhtml5/formy/polzunok> (дата обращения: 27.06.2016).

11. JavaScript – Центральный JavaScript ресурс. [Электронный ресурс] – Редактирование и отладка скриптов // URL. <http://javascript.ru/tools/development> (дата обращения: 27.06.2016).

12. JQUERY – JavaScript Библиотека [Электронный ресурс]. – Ползунок UI //URL: http://jquery.page2page.ru/index.php5/Ползунок_UI (дата обращения: 27.06.2016).

13. Tutotial – Уроки php, JS, css [Электронный ресурс]. – Работа с ползунком range в JS // URL: <http://tutorial.promo.net.ua/java-script/range-service-for-border/input-type-range/> (дата обращения: 28.06.2016).

14. Webstudio2u – Веб-программирование.[Электронный ресурс]. – Кроссбраузерность. // URL <http://webstudio2u.net/ru/programming/126-cross-browser.html> (дата обращения: 29.06.2016).

15. Wikipedia – свободная энциклопедия [Электронный ресурс]. – Кроссбраузерность // URL: <https://ru.wikipedia.org/wiki/Кроссбраузерность> (дата обращения: 29.06.2016).