

گاهی از اوقات نیاز است کاربر در یک جعبه متنی، فقط متن فارسی وارد کند؛ حتی اگر صفحه کلید او فارسی نباشد و یا بنابر درخواست او، جهت بالا رفتن سرعت ورود اطلاعات یک چنین قابلیتی نیاز می‌شود. چندین سال قبل [farsitype.js](#) اینکار را انجام می‌داد. این اسکریپت با مرورگرهای جدید سازگار نیست و برای نمونه `initKeyEvent` آن در نگارش‌های قدیمی فایرفاکس کار می‌کرد، در کروم هیچ وقت پشتیبانی نشد (به نام `initKeyboardEvent` موجود است؛ اما برای جایگزین کردن حروف عمل نمی‌کند) و مدتی است که فایرفاکس هم به دلایل امنیتی آن را غیرفعال کرده است.

به همین جهت افزونه `farsiInput`، که کدهای آن را در ادامه مشاهده می‌کنید، تهیه گردید. این افزونه تا این تاریخ با IE، فایرفاکس، کروم و اپرا سازگار است و توسط آن کاربر بدون نیاز به داشتن یک صفحه کلید فارسی می‌تواند فارسی تایپ کند. برای سوئیچ به حالت انگلیسی، دکمه `Scroll lock` باید روشن شود و این مورد توسط پارامتر `changeLanguageKey` قابل تغییر است.

```
// 
(function ($) {
    $.fn.farsiInput = function (options) {
        var defaults = {
            changeLanguageKey: 145 /* Scroll lock */
        };
        var options = $.extend(defaults, options);

        var lang = 'fa';

        var keys = new Array(1711, 0, 0, 0, 0, 1608, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1705,
1572, 0, 1548,
                                1567, 0, 1616, 1571, 8250, 0, 1615, 0, 0, 1570, 1577, 0, 0, 0, 1569, 1573,
0, 0, 1614, 1612, 1613, 0, 0,
                                8249, 1611, 171, 0, 187, 1580, 1688, 1670, 0, 1600, 1662, 1588, 1584,
1586, 1740, 1579, 1576, 1604, 1575,
                                1607, 1578, 1606, 1605, 1574, 1583, 1582, 1581, 1590, 1602, 1587, 1601,
1593, 1585, 1589, 1591, 1594, 1592);

        var substituteChar = function (charCode, e) {
            if (navigator.appName == "Microsoft Internet Explorer") {
                window.event.keyCode = charCode;
            }
            else {
                insertAtCaret(String.fromCharCode(charCode), e);
            }
        };

        var insertAtCaret = function (str, e) {
            var obj = e.target;
            var startPos = obj.selectionStart;
            var endPos = obj.selectionEnd;
            var scrollTop = obj.scrollTop;
            obj.value = obj.value.substring(0, startPos) + str + obj.value.substring(endPos,
obj.value.length);
            obj.focus();
            obj.selectionStart = startPos + str.length;
            obj.selectionEnd = startPos + str.length;
            obj.scrollTop = scrollTop;
            e.preventDefault();
        };

        var keyDown = function (e) {
            var evt = e || window.event;
            var key = evt.keyCode ? evt.keyCode : evt.which;
            if (key == options.changeLanguageKey) {
                lang = (lang == 'en') ? 'fa' : 'en';
                return true;
            }
        };

        var fixYeKeHalfSpace = function (key, evt) {
            var originalKey = key;
            var arabicYeCharCode = 1610;
            var persianYeCharCode = 1740;
            var arabicKeCharCode = 1603;
            var persianKeCharCode = 1705;
            var halfSpace = 8204;
</pre>
</div>
<div data-bbox="495 956 522 970" data-label="Page-Footer">۱/۵</div>
```

```

        switch (key) {
            case arabicYeCharCode:
                key = persianYeCharCode;
                break;
            case arabicKeCharCode:
                key = persianKeCharCode;
                break;
        }

        if (evt.shiftKey && key == 32) {
            key = halfSpace;
        }

        if (originalKey != key) {
            substituteChar(key, evt);
        }
    };

    var keyPress = function (e) {
        if (lang != 'fa')
            return;

        var evt = e || window.event;
        var key = evt.keyCode ? evt.keyCode : evt.which;
        fixYeKeHalfSpace(key, evt);
        var isNotArrowKey = (evt.charCode != 0) && (evt.which != 0);
        if (isNotArrowKey && (key > 38) && (key < 123)) {
            var pCode = (keys[key - 39]) ? (keys[key - 39]) : key;
            substituteChar(pCode, evt);
        }
    }

    return this.each(function () {
        var input = $(this);
        input.keypress(function (e) {
            keyPress(e);
        });
        input.keydown(function (e) {
            keyDown(e);
        });
    });
})(jQuery);
// ]]>

```

مثالی از نحوه بکارگیری آن:

```

<html>
<head>
    <title>تکست باکس فارسی</title>
    <script type="text/javascript" src="jquery-1.9.1.min.js"></script>
    <script type="text/javascript" src="jquery.farsiInput.js"></script>
    <style type="text/css">
        input, textarea
        {
            font-family: tahoma;
            font-size: 9pt;
        }
    </style>
</head>
<body>
    <input dir="rtl" id='text1' />
    <br />
    <textarea dir="rtl" id='text2' rows="15" cols="84"></textarea>
    <script type="text/javascript">
        $(function () {
            $("#text1, #text2").farsiInput();
        });
    </script>
</body>
</html>

```


نظرات خوانندگان

نویسنده: امیرحسین جلوداری
تاریخ: ۲۱:۲۰ ۱۳۹۲/۰۲/۲۰

خیلی ممنون (: ... زیاد از حد کاربردی: دی

نویسنده: کاوه احمدی
تاریخ: ۱۸:۴۴ ۱۳۹۲/۰۲/۲۱

البته FarsiType اسکرپت قدیمی‌ای است که به نظرم این روزها کارکرد خود را ازدست داده است. اما اشارات شما در مورد کارکرد آن در مرورگرهای متفاوت صحیح نیست. دست کم بنده در آخرین نسخه‌های مرورگرها می‌توانم به درستی از آن استفاده کنم. احتمالاً شما از آخرین نسخه آن بهره نگرفته‌اید (:

<http://farsitype.ir>

نویسنده: الهه
تاریخ: ۱۹:۵۷ ۱۳۹۲/۰۲/۲۱

چقدر خوب بود ، ممنونم خیلی دنبالش بودم

نویسنده: محسن
تاریخ: ۲۱:۲۷ ۱۳۹۲/۰۲/۲۱

بدلیل سازگاری مناسب با کتابخانه JQuery افزونه FarsiInput را بهتر از FarsiType ارزیابی می‌کنم. همچنین FarsiType قابلیت هایی دارد که پیشنهاد می‌کنم به FarsiInput اضافه شود. برای مثال تغییر Direction که کار آسانی است و همچنین تغییر زبان با ctrl + Space. غیرفعال شدن در صورتی که صفحه کلید فارسی است، پیشنهاد نمی‌شود چرا که نیاز مبرمی به جابجایی حروف عربی مورد نیاز است. در کل FarsiInput سبک‌تر است و آینده بهتری را برای آن می‌توان متصور شد. همانطور که یکی دیگر از دوستان اشاره کردند این افزونه بیش از حد کاربردی است. مخصوصاً در برنامه‌های کاربردی تحت وب! با تشکر از وحید نصیری و همچنین کاوه احمدی برای همه تلاششان.

نویسنده: یزدان
تاریخ: ۱۱:۳۴ ۱۳۹۲/۰۲/۲۴

بسیار خوب . فقط اگه برای اعداد هم فارسی میشد کامل بود . اگر مقدوره قابلیت فارسی سازی اعداد رو هم اضافه کنید .

نویسنده: علیرضا
تاریخ: ۱۶:۲۴ ۱۳۹۲/۰۶/۲۶

برای فارسی کردن اعداد متغیر keys را به این صورت عوض کنید

```
var keys = new Array(1711, 0, 0, 0, 0, 1608, 0, 0, 0, 1776, 1777, 1778, 1779, 1780, 1781, 1782, 1783, 1784, 1785, 0, 1705, 1572, 0, 1548, 1567, 0, 1616, 1571, 8250, 0, 1615, 0, 0, 1570, 1577, 0, 0, 0, 1569, 1573, 0, 0, 1614, 1612, 1613, 0, 0, 8249, 1611, 171, 0, 187, 1580, 1688, 1670, 0, 1600, 1662, 1588, 1584, 1586, 1740, 1579, 1576, 1604, 1575, 1607, 1578, 1606, 1605, 1574, 1583, 1582, 1581, 1590, 1602, 1587, 1601, 1593, 1585, 1589, 1591, 1594, 1592);
```

همچنین برای تغییر جهت صفحه متد each را تغییر دهید

```
return this.each(function () {  
    var input = $(this);  
    input.css("direction","rtl");  
    input.keypress(function (e) {  
        keyPress(e);  
    });  
    input.keydown(function (e) {  
        keyDown(e);  
    });  
});
```

نویسنده: علیرضا

تاریخ: ۱۳۹۲/۱۰/۰۳ ۱۳:۳

چگونه اعداد داخل editor را فارسی کنیم؟ با jquery داخل ادیتور را select کردم ولی کار نکرد!

editor: ckeditor

نویسنده: شفیقی

تاریخ: ۱۳۹۳/۰۳/۰۳ ۱۴:۵۹

باسلام

متأسفانه هم فایل شما و هم فایل FarsiType این مشکل رو داره که اگه در صفحه از Ajax و UpdatePanel استفاده بشه ، فقط در اولین مرتبه که صفحه لود میشه کار میکنه . همین که یکبار صفحه PostBack بشه ، دیگه کار نمیکنه و انگلیسی تایپ میشه . لطفا درصورت امکان یه راهکار بدید تا قابل استفاده باشه .

ممنون

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۳/۰۳ ۱۷:۵

« استفاده‌ی همزمان از آپدیت پنل ASP.Net و پلاگین‌های جی‌کوئری »

نویسنده: ناهید

تاریخ: ۱۳۹۳/۰۷/۲۶ ۱۲:۳۶

سلام

اگر بخوایم فقط حروف فارسی باشه و اعداد رو نزنه چیکار باید بکنیم.
در ضمن مرسی مفید و عالی بود

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۷/۲۶ ۱۵:۲۱

در متدهای keypress و keydown فوق، اگر متد e.preventDefault() فراخوانی شود، دکمه‌ی فشرده شده نمایش داده نخواهد شد. در همینجا بازه‌ی key دریافتی را بررسی کنید. مثلا 97 مساوی a است تا 122 که z است. اگر خارج از آن بود متد e.preventDefault() را فراخوانی کنید.