Filtrer une liste en pur javascript et sans jQuery - via Ecyseo

jeudi 5 juin 2014, par placido

```
/*
# ------------------ BEGIN LICENSE BLOCK ------------------
#
# This file is part of SIGesTH
#
# Copyright (c) 2009 - 2014 Cyril MAGUIRE, (!Pragmagiciels)
# Licensed under the CeCILL v2.1 license.
# See http://www.cecill.info/licences.fr.html
#
# -------------------- END LICENSE BLOCK -------------------
*/
/*
HTML Structure
Filtrer
```

2. First element

```
*/
;(function(window,undefined){

    'use_strict';

    var keyCode = 0;
    var isIE = isBrowserIE();
    function isBrowserIE() {
        var nav = navigator.userAgent.toLowerCase();
        return (nav.indexOf('msie') != -1) ? parseInt(nav.split('msie')[1]) : false;
    }

    // from http://www.javascripter.net/faq/onkeydown.htm
    function fromKeyCode (n) {
        if( 47<=n && n<=90 ) return unescape('%'+(n).toString(16))
        if( 96<=n && n<=105) return 'NUM '+(n-96)
        if(112<=n && n<=135) return 'F'+(n-111)
        switch (n) {
            case (3): return 'Cancel' //DOM_VK_CANCEL
            case (6): return 'Help'   //DOM_VK_HELP
            case (8): return 'Backspace'
            case (9): return 'Tab'
            case (12): return 'NUM 5'  //DOM_VK_CLEAR
            case (13): return 'Enter'
            case (16): return 'Shift'
```

```
        case (17): return 'Ctrl'
        case (18): return 'Alt'
        case (19): return 'Pause|Break'
        case (20): return 'CapsLock'
        case (27): return 'Esc'
        case (32): return 'Space'
        case (33): return 'PageUp'
        case (34): return 'PageDown'
        case (35): return 'End'
        case (36): return 'Home'
        case (37): return 'Left Arrow'
        case (38): return 'Up Arrow'
        case (39): return 'Right Arrow'
        case (40): return 'Down Arrow'
        case (42): return '*' //Opera
        case (43): return '+' //Opera
        case (44): return 'PrntScrn'
        case (45): return 'Insert'
        case (46): return 'Delete'

        case (91): return 'WIN Start'
        case (92): return 'WIN Start Right'
        case (93): return 'WIN Menu'
        case (106): return '*'
        case (107): return '+'
        case (108): return 'Separator' //DOM_VK_SEPARATOR
        case (109): return '-'
        case (110): return '.'
        case (111): return '/'
        case (144): return 'NumLock'
        case (145): return 'ScrollLock'

        //Media buttons (Inspiron laptops)
        case (173): return 'Media Mute On|Off'
        case (174): return 'Media Volume Down'
        case (175): return 'Media Volume Up'
        case (176): return 'Media >>'
        case (177): return 'Media <'
        case (191): return '/ ?'
        case (192): return '\` ~'
        case (219): return '[ {'
        case (220): return '\\ |'
        case (221): return '] }'
        case (222): return '\' "'
        case (224): return 'META|Command'
        case (229): return 'WIN IME'

        case (255): return 'Device-specific' //Dell Home button (Inspiron laptops)

    }
    return null
```

```
    }

    function arrayCompare(a1, a2) {
        if (a1.length != a2.length) return false;
        var length = a2.length;
        for (var i = 0; i < length; i++) {
            if (a1[i] !== a2[i]) return false;
        }
        return true;
    }
    function inArray(needle, haystack) {
        var length = haystack.length;
        for(var i = 0; i < length; i++) {
            if(typeof haystack[i] == 'object') {
                if(arrayCompare(haystack[i], needle)) return true;
            } else {
                if(haystack[i] == needle) return true;
            }
        }
        return false;
    }

    var object = {
        init: function() {
            var input = document.getElementsByTagName('input');
            for (var i = 0, c = input.length; i < c; i++) {
                this.addEventListener(input[i],"keypress", this);
                this.addEventListener(input[i],"keydown", this);
            };
        },
        field:'',
        idPressed: '',
        fieldVal: '',
        addEventListener: function(el, eventName, handler) {
            if (el.addEventListener) {
                el.addEventListener(eventName, handler, false);
            } else {
                el.attachEvent('on' + eventName, object.handleEvent);
            }
        },
        handleEvent: function(e) {
            var obj = Object;
            var evt = e ? e:event;
            var chrTyped, chrCode = 0;

            if (evt.charCode!=null)     chrCode = evt.charCode;
            else if (evt.which!=null)   chrCode = evt.which;
            else if (evt.keyCode!=null) chrCode = evt.keyCode;

            if (chrCode==0) chrTyped = 'SPECIAL KEY';
            else chrTyped = String.fromCharCode(chrCode);
```

```
        if (evt.keyCode==8) chrTyped = 'Backspace';

        if (isIE) {
            obj = object;
        } else {
            obj = this;
        }
        obj.field = evt.target || evt.srcElement;
        obj.idPressed = obj.field.getAttribute('id');
        obj.fieldVal = obj.field.value;

        if (chrTyped != 'SPECIAL KEY') {
            obj.action(chrTyped);
        }
    },
    action: function(a) {
        var obj = Object;
        if (isIE) {
            obj = object;
        } else {
            obj = this;
        }

        var id = obj.idPressed;
        if (id == null) return true;

        var val = obj.fieldVal;
        var idList = id.substring(14,id.length); //we remove "inputFilterFor" from input id

        var list = document.getElementById('filterFor'+idList); //to find list associated
with input
        if (list == null) return true;

        var listItem = list.getElementsByTagName('li');
        if (listItem == null) return true;

        var span = null;

        //text filled in input form
        if (a != 'Backspace') {
            //we add key because input value is filled after key is up
            val += a;
        } else {
            val = val.substring(0, val.length-1);//we remove last char because input value
is updated after key is up
        }
        // if empty, all the list is display
        if(val == ''){
            for(var i =0, count = listItem.length; i < count; i++) {
                listItem[i].style.display = 'list-item';
            }
```

```
            return true;
        }
        // regex build from what is filled in form : (.*)e(.*)x(.*)e(.*)m(.*)p(.*)l(.*)e(.*)
        var regexp = '\\b(.*)';
        for(var i =0, count = val.length; i < count; i++){
            if (inArray(val[i],['.','?','*','+','/'])) {regexp += '(\\'+val[i]+')(.*)';}
            else if (val[i] != '') {regexp += '('+val[i]+')(.*)';}
        }
        regexp += '\\b';
        // for each item of the list
        for(var i =0, count = listItem.length; i < count; i++) {
            listItem[i].style.display = 'list-item';

            // we looking for span tag wrapped by link tag
            var links = listItem[i].getElementsByTagName('a');
            for (var k = 0, c= links.length; k < c; k++) {
                var span = links[k].getElementsByTagName('span');
            }
            // we take the first one
            if (span[0] != undefined) {
                span = span[0];

                if(span.innerHTML.match(new RegExp(regexp,'i'))) {
                    listItem[i].style.display = 'list-item';
                } else {
                    listItem[i].style.display = 'none';
                }
            }
        }
        return true;
    }
};

// Init
object.init();

})(window);
```

Télécharger

Voir en ligne : Filtrer une liste en pur javascript et sans jQuery - Ecyseo