

Base de connaissances JavaScript

Rafraîchir un formulaire à la sortie d'un champ

Permet de simuler le clique sur le bouton "Rafraîchir" lors de la sortie de l'édition d'un champ.

Code JS

```
<a name="Refresh1"></a>

<script type="text/javascript">
$('Field_NOMDUCHAMP').onblur = function (){
    document.Form0.action=document.Form0.action+'#Refresh1';
    document.getElementById('buttonRefresh').click();
}

</script>

<a name="Refresh1"></a>

<script type="text/javascript">
$('Field_NOMDUCHAMP').onblur = function (){
    document.Form0.action=document.Form0.action+'#Refresh1';
    document.getElementById('buttonRefresh').click();
}

</script>
```

Remplacez **NOMDUCHAMP** par le nom interne du champ concerné.

Masquer une section

Permet de masquer définitivement une section en javascript.**Code JS**

[En V5](#)

```
<script
type="text/javascript">$('Section_XXXXX').parentNode.parentNode.style.display
= "none"</script>
```

[En V6](#)

```
<script type="text/javascript">
WKYJS.hideSection('XXXXX');
</script>
```

Contrôle de la visibilité des sections par les valeurs renseignées pour un champ

En cliquant sur un bouton radio représentant une valeur d'un champ, rendre possible de montrer ou de masquer une ou plusieurs sections d'un formulaire.

Code JS

En V6

Une nouvelle librairie a été développée pour permettre de faciliter le masquage des champs et des sections.

Merci de vous référencer à [WKYJS](#).

En V5

```
<script type="text/javascript">
  var sectionElements = [];

  function make_sectionswitch(i, show) {

    return function () {
      for (var j = 0; j > sectionElements.length; j++) {

        if (sectionElements[j] != null) sectionElements[j].style.display =
getDisplay((i == j), show);

      }

    }

  }

  function getDisplay(condition, show) {

    if (condition) return show ? "" : "none";
    else return show ? "none" : "";

  }

  function setSectionsDisplay(fieldName, sectionNames, fieldValues, show) {
    var radios = document.getElementById("Field_" +
fieldName).getElementsByTagName("input");

    var index = sectionElements.length;
    for (i = 0; i > radios.length; i++) {
      var v = radios[i].value;

      for (j = 0; j > fieldValues.length; j++) {
        if (fieldValues[j] == v) {
```

```

    sectionElements[index] = document.getElementById("Section_" +
sectionNames[j]);
    if (sectionElements[index] != null) {
        radios[i].onclick = make_sectionswitch([index], show);
        sectionElements[index].style.display = getDisplay(radios[i].checked, show);
    }
    break;
}
}
index++;
}

}
</script>

```

Exemple

```

setSectionsDisplay ("ChoixDuTypeDeContrat", ["Assurance Vie", "Assurance
Habitation"], ["1", "2"], true);

```

Pour un click sur le bouton 1 du champ ChoixDuTypeDeContrat, la section "Assurance Vie" seule est montrée. Pour un click sur le bouton 2 du champ ChoixDuTypeDeContrat, la section "Assurance Habitation" seule est montrée.

[Fichier:hideSectionsByRadio.zip](#)

Masquer l'ajout de ligne d'un tableau dynamique

En ajoutant le code dans le formulaire, il n'est plus possible d'ajouter ou supprimer des lignes.

Remplacer dans le code le "tab_dyn" par le nom de votre table dynamique

Code CSS

```

<style type="text/css">
    table.dynamicTable td.actionColumn { display: none; }
    #FormFields tfoot { display: none; }
</style>

```

OU BIEN

```

<style type="text/css">
#Table_tab_dyn tfoot, #Table_tab_dyn td.actionColumn { display: none; }
</style>

```

Colorer le tour des champs qui ont le focus

Il est possible de colorer la bordure des champs qui ont le focus dans les formulaires en manipulant la feuille de style principale de workey. Pour se faire, il vous faut modifier le fichier **workey-style.css**.

Voici la déclaration courante que nous avons choisi pour nos champs de texte du support, pour vous donner un ordre d'idée sur le type de modification à effectuer.

```
textarea:focus, select:focus, input:focus {  
border-color: rgba(82, 168, 236, 0.8);  
box-shadow: 0 1px 3px rgba(0, 0, 0, 0.1) inset, 0 0 8px rgba(82, 168, 236,  
0.6);  
outline: 0 none;  
}
```

Masquer/Afficher des champs en fonction du changement de valeur d'autre champs

Ce script permet d'afficher des champs en fonction du changement de valeur d'un autre champ.

Il est possible d'ajouter des conditions pour obtenir l'affichage en fonction d'une valeur spécifique.

showLists.js

```
/**  
 * Affiche un champ lorsque la valeur d'un autre champ change  
 * Copyleft C-Log International - 2012  
 */  
  
document.observe('dom:loaded', function()           // Attend que les objets de la  
page soient chargés pour exécuter le script  
{  
    function hideSecunda() { $('Field_Secunda').hide(); }           // Cache  
le champ  
    function hideTertia() { $('Field_Tertia').hide(); }           // Cache  
le champ  
    function hideQuarta() { $('Field_Quarta').hide(); }           // Cache  
le champ  
    function showSecunda() { $('Field_Secunda').show(); }           //  
Affiche le champ  
    function showTertia() { $('Field_Tertia').show(); }           // Affiche  
le champ  
    function showQuarta() { $('Field_Quarta').show(); }           // Affiche  
le champ  
    hideQuarta(); hideTertia(); hideSecunda();           // Cache les champs  
    var selectPrima=document.getElementById("Field_Prima"),  
    selectSecunda=document.getElementById("Field_Secunda"),  
    selectTertia=document.getElementById("Field_Tertia"),  
    selectQuarta=document.getElementById("Field_Quarta");  
  
    selectPrima.onchange=function showSecunda() {  
    $('Field_Secunda').show(); }           // Affiche 2 quand 1 change  
    selectSecunda.onchange=function showTertia() {  
    $('Field_Tertia').show(); }           // Affiche 3 quand 2 change
```

```
        selectTertia.onchange=function showQuarta() {  
$( 'Field_Quarta' ).show(); }           // Affiche 4 quand 3 change  
});
```

Usage

Voici un fichier .gti (version 4.2) illustrant le fonctionnement du script.

[Fichier:showlists.zip](#)

Définir la taille d'une colonne dans un tableau dynamique

Il est possible de définir une taille fixe pour une colonne d'un tableau dynamique en ajoutant un champ texte non éditable avec le code suivant.

Code JS

```
<style type="text/css">  
#Field_Activite { display: block; min-width: 600px; }  
</style>
```

Field_Activite : Activite est le nom du champ de la colonne.

Envoyer un document en cliquant sur un état

En ajoutant ce code dans un formulaire, il est possible d'envoyer directement un document en cliquant sur l'état.

Modifier le fichier JS "clog-workey.war\scripts\hideableblock.js" et ajouter la ligne suivante :

```
openTableLoginPassword (idxClicked);  
document.getElementById("buttonSubmit").click(); // *** Ligne à ajouter ***  
return true;
```

Présélectionner le premier état et cacher la sélection des états suivants

Ajouter le code suivant dans un champ JS du formulaire à simplifier :

```
<script type="text/javascript"> document.observe('dom:loaded', function(){  
$( 'NextStateSelection' ).hide();  
document.getElementsByName('nextStateRadio')[0].checked = true;});</script>
```

Définir la taille d'une colonne dans un tableau dynamique

Pour ajouter dans un formulaire un bouton <rafraichie>

Ajouter un champ de type texte non-éditable avec comme valeur par défaut :

- Avant la 3.5 :

```
<a name="Bouton1"></a>
<a
href="javascript:document.Form0.action=document.Form0.action+'#Bouton1';submi
tLink(document.Form0,'$LinkSubmit$1');">
  
</a>
```

- A partir de la 3.5 :

```
<a name="Bouton1"></a>
<a
href="javascript:document.Form0.action=document.Form0.action+'#Bouton1';docum
ent.getElementById('buttonRefresh').click();">
  
</a>
```

Rendre un champ obligatoire par JS (dans un tableau dynamique) :

Voilà le code JS à ajouter pour vérifier que le champ Composante_5 est toujours rempli :

```
<script type="text/javascript">
(function () {
  var submit = document.getElementsByName("$LinkSubmit$1");
  var old_onclick = submit.onclick;
  submit.onclick = function () {
    var v = document.getElementById("Field_Composante_5").value;
    if (v === null || v === "") {
      alert("Le champ 'Composante' doit être rempli.");
      return;
    }
    old_onclick();
  };
})();
</script>
```

Auto-complétion à partir d'un Ldap :

Deux objectifs :

- Rendre un champ en auto-complétion à partir d'un Ldap
- Renseigner d'autres champs à partir de ce champ.

Voici un GTI illustratif et un WAR à placer sous /deploy.

[Fichier:TestLdap.zip](#)

Sélection du premier état par défaut

Sélection d'un état par défaut, Le premier. Code expression par défaut à mettre dans un champ texte non éditable.

```
<script type="text/javascript">
  (function(){var radio = document.getElementsByName("nextStateRadio")[0];
  radio.checked=true; radioStateClick(radio);})();
</script>
```

Voilà le nouveau script – il s'exécute au moment du window.onload() lorsque tout les éléments de la page son présent.

```
<script type="text/javascript">
(function(){
    var old_onload=window.onload;window.onload=function(){
        old_onload();
        var
radio=document.getElementsByName("nextStateRadio")[0];if(radio===null)return;
radio.checked=true;
        radioStateClick(radio);
    }
})();
</script>
```

Afficher des infobulles sur un ou des objets

Affichages de bulles d'aide lorsque le curseur de la souris passe sur des liens, paragraphes ou autres objets.

infobulles.js

Voici le code source à sauvegarder dans clog-workey.war.

```
document.observe('dom:loaded', function() {
    // Installation de l'infobulle sur le champ Field_FormIM_datecons
    infoBulle($('Field_FormIM_datecons'),
        "<b>Quel temps fait il ?</b><br/>Il fait beau et
chaud, les nuages vont vers l'ouest."
    );
    // Installation de l'infobulle sur le champ Field_Form_IM_datefinval
    infoBulle($('Field_Form_IM_datefinval'),
        "<b>Que dois-je renseigner dans le champ Date de fin
?</b><br/>Indiquer la date de fin de la prestation."
    );
});
```

```
function infoBulle(elt, text){
    var bulle = new Element('div', {
        id:    elt.id + '_infobulle',
```

```

        style: 'padding:3px; border: solid 1px black;
background-color:#EEE; color:black; display: none; position:absolute; top:0;
left:0;'
    }
    );

    bulle.innerHTML = text;

    var epos = elt.positionedOffset();
    bulle.style.top  = (epos.top  + 20) + "px";
    bulle.style.left = (epos.left + 10) + "px";

    elt.insert({'after':bulle});
    Event.observe(elt, "mouseover", function(){ bulle.style.display = ''
});
    Event.observe(elt, "mouseout", function(){ bulle.style.display =
'none' });
}

```

Appel des fonctions sur les objets

Dans un formulaire Workey:

- Créer un champ de type chaîne de caractères en lecture seule non éditable
- Valeur par défaut du champ: `<script type="text/javascript" src="/js/infobulles.js"></script>`

Afficher les sections de formulaires dans des onglets séparés.

Ce script permet d'afficher les sections d'un formulaire à l'horizontal dans des onglets presque indépendants.

sections_to_tabs.js

```

/* Conversion des sections de formulaire en onglets placés en haut du
formulaire .
*
* (c) 2009 - C-Log International
*/
(function(){

    // Recherche des sections du formulaires (éléments <table> dont l'Id
commence par "Section_"

    // XXX Actuellement, on recherche toutes les sections du formulaire. On
pourrait choisir de
    // ne parcourir les sections directement à la racine du formulaire ou
bien ne sélectionner
    // que les section commençant par "Section_TAB_" (par exemple)

```



```

    var tables =
document.getElementById("FormFields").getElementsByTagName("table");
    var sections = [];
    var firstSet = false;

    for (var i = 0; i < tables.length; i++)
    {
        if (tables[i].id && tables[i].id.match("^Section_.*$"))
        {
            var cap = tables[i].getElementsByTagName("caption")[0];
            // IE ne supporte pas textContent
            var txt = cap.innerText || cap.textContent;
            var name = txt.replace(/^\s+/, "").replace(/\s+$/, "");
            sections[name] = tables[i];
            if (firstSet === true)
            {
                tables[i].style.display = "none";
            }
            firstSet = true;
            tables[i].className = "collapsible tabs";
        }
    }

    var formFieldBody =
document.getElementById("FormFields").getElementsByTagName("tbody")[0];

    // Ajout d'une ligne dans le formulaire contenant les onglets
    var tabsRow = document.createElement("tr");
    var tabsCell = document.createElement("td");
    tabsCell.colSpan = 2;
    tabsCell.id = "FormTabs";

    //var tabsLink = document.createElement("a");
    //tabsLink.setAttribute("name", "FormTabsLink");
    //tabsCell.appendChild(tabsLink);

    var tabsUl = document.createElement("ul");
    tabsUl.setAttribute("class", "workyTabs");
    firstSet = false;

    var selectedTab = null;

    var hash = window.location.hash;
    //alert("hash: [" + hash + "] / fl: [" + window.location + "]");

    for (var x in sections)
    {
        if(typeof sections == 'function') { continue; }

        // Création d'un onglet par section

```

```

var tabsLi = document.createElement("li");
tabsLi.id = "TAB_" + x;

var hashPart = "#" + sections.id;

if (firstSet === false)
{
    if (hash)
    {
        if (hash == hashPart)
        {
            tabsLi.className = "tabSelected";
            selectedTab = sections;
            firstSet = true;
        }
    }
    else
    {
        tabsLi.className = "tabSelected";
        selectedTab = sections;
        firstSet = true;
    }
}

var tabsA = document.createElement("a");
tabsA.setAttribute("href", hashPart);

//AA    tabsA.observe("click", function(ev){
tabsA.onclick = function(){

    var form = document.getElementById('workeyForm');
    var cleanAction = form.action.replace(/#.*\/, '');
    form.action = cleanAction + this.hash;

    // Action lors de la sélection d'un onglet, affichage de la
section et coloration de l'onglet sélectionné
    for (var y in sections)
    {
        if(typeof sections[y] == 'function') { continue }

        if (y == this.innerText || y == this.textContent)
        {
            sections[y].style.display = "";
            document.getElementById("TAB_" + y).className =
"tabSelected";
        }
        else
        {
            sections[y].style.display = "none";
            document.getElementById("TAB_" + y).className = "";
        }
    }
}

```

```

        }

        return false;
    }
//AA    );

    var link = document.createTextNode(x);
    tabsA.appendChild(link);
    tabsLi.appendChild(tabsA);
    tabsUl.appendChild(tabsLi);
}

tabsCell.appendChild(tabsUl);

tabsRow.appendChild(tabsCell);

var firstRow = formFieldBody.getElementsByTagName("tr")[0];
formFieldBody.insertBefore(tabsRow, firstRow);

if (selectedTab)
{
    for (var y in sections)
    {
        if (sections[y].style) { sections[y].style.display = selectedTab ==
sections[y] ? "" : "none"; }
    }
}

})();

```

sections_to_tabs.css

```

/* http://htmldog.com/articles/tabs/ */
#FormTabs {
    width: 100%;
    border-bottom: Solid 1px Black;
    padding: 0.5em 0.5em 0 1em;
}

#FormTabs ul {
    list-style: none;
    padding: 0;
    margin: 0;
}

#FormTabs li {
    display: inline;
    border: solid;
    white-space: nowrap;
    border-width: 1px 1px 0 1px;
    margin: 0 0.5em 0 0;
}

```

```

}

#FormTabs li a {
    padding: 0 .5em;
    text-decoration: none;
}

#FormTabs li.tabSelected {
    background-color: #DDD;
}

table.tabs caption {
    display: none;
}

```

Appel de la fonction dans un formulaire

Créer un champ de type text non éditable et ayant comme valeur par défaut:

```

<script type="text/javascript" src="/js/sections_to_tabs.js"></script>
<link type="text/css" rel="stylesheet" href="/js/sections_to_tabs.css"/>

```

Ce champ doit être placé en bas de formulaire afin que le script s'exécute après que les autres objets de la page soient chargés.

Convertir les retours chariots pour un transfert RTF

Ce script permet de convertir les caractères de retour chariots venant d'une zone de texte simple en retour chariots compatibles avec un format RTF.

CR.js

Champs: Champ1 (zone de texte avec retour à la ligne automatique), Champ2 (insertion du script JavaScript), Champ3 (zone de texte avec retour à la ligne automatique).

```

function updatechamp3()
{
    var reg=new RegExp("\n", "gi");
    var source = $(Field_Champ1).value;
    $(Field_Champ3).value=source.replace(reg,"\par");
}

```

Exemple : [Fichier:CR.zip](#)

[WorkeyFormCustomizer](#)

[This document](#) shows how to customize forms generated by Workey. It is aimed at developers with Javascript and DOM skills.