

# Rules for writing javascript NET Genium Framework

# Content

<b>1</b>	<b>Spaces, tabs and newlines .....</b>	<b>3</b>
<b>2</b>	<b>Simple apostrophes.....</b>	<b>4</b>
<b>3</b>	<b>Intermediates.....</b>	<b>5</b>
<b>4</b>	<b>Buttons and postbacks .....</b>	<b>7</b>

# 1 Spaces, tabs and newlines

- ❗ We use spaces between "if" and the left parenthesis, to the left and right of the "=" sign, after the comma as a separator for variables or function parameters etc.
- ❗ We use tabs to indent the code from the left margin of a new line. The tab represents two consecutive spaces, and is a shortcut key for inserting them into the code in the javascript editor. Each code dive requires one tab – two spaces. We do not indent the source code at the first level.
- ❗ Enters – transitions to a new line – we use before the left parenthesis "{" and after it, as well as before the right parenthesis "}" and behind her.

```
function hotovo()  
{  
    if (!vykazy()) return false;  
  
    var stav = control_GetValue(#ng_stav#);  
    if (stav == 'Schválený#en:Approved')  
    {  
        control_SetValue(#ng_stav#, stav = stavHistory);  
        alert(translate('Úkol je možné schválit pouze tlačítkem Schválit, které je umístěno pod pol  
approved only by approver and only by button Approve!'));  
    }  
  
    if (stav == 'Hotový#en:Completed')  
    {  
        if (#ng_dokoncenokym#.value == '') #ng_dokoncenokym#.value = '#loginname#';  
        if (#ng_datumdokonceni#.value == '') control_SetValue(#ng_datumdokonceni#, new Date());  
    }  
    else if (nID == 0)  
    {  
        #ng_dokoncenokym#.value = '';  
        control_SetValue(#ng_datumdokonceni#, '', '');  
    }  
    else  
    {  
        #ng_dokoncenokym#.value = '';  
        control_SetValue(#ng_datumdokonceni#, '', '');  
    }  
  
    control_SetValue(#ng_schvaleno#, '', '');  
    barva();  
}
```

## ⚠ Bad example

```
function hotovo(){if(!vykazy())return false;var stav=control_GetValue(#ng_stav#);if(stav=='Schválený#en:Approved'){
```

## 2 Simple apostrophes

- ❶ *We always use single apostrophes as the default delimiter. Because it is possible to combine javascript code with server functions, we always leave double quotes behind server function calls.*
- ❷ *Server function parameters must often be enclosed in double quotes because the parameter value may contain a comma, which is considered a parameter separator by default. Texts inside double quotes are treated as string constants and do not replace server functions, only variables. The only exception is the parameters of the "SQL" and "SQLARRAY" functions, inside which it is possible to call server functions, even though the parameters of these functions are wrapped in double quotes.*

```
function showSchedule()
{
    var resources = 'SQL("SELECT ng_uzivatel FROM ng_planukolu WHERE pid = #id#", ";")';
    window.open('Default.aspx?newtab-viewpage-408-width-' + innerWidth2() + '-resources-' + resources);
}

var array = SQLARRAY("SELECT ng_cislozakazky, ng_nazevzakazky, SUM(ng_hodiny) FROM ng_pvcastecny LEFT JOIN ng_crmobchvztah J1 ON J1.id =
ng_pvcastecny.ng_idodberatele LEFT JOIN ng_zakazka J2 ON J2.id = ng_pvcastecny.ng_idzakazky LEFT JOIN ng_pvtyp J3 ON J3.id = ng_pvcastecny.ng_typ WHERE
ng_pvcastecny.ng_datum = EQUALS(#ng_datum:value#,,FORMATDATESQL(#today#),FORMATDATESQL(#ng_datum:value#)) AND ng_pvcastecny.ng_uzivatel =
EQUALS(#ng_uzivatel:value#,,FORMATSTRINGSQL(#loginname#),FORMATSTRINGSQL(#ng_uzivatel:value#)) GROUP BY ng_cislozakazky, ng_nazevzakazky");
for (var i = 0; i < array.length; i++)
{
    sb.append('<tr><td align="center">');
    sb.append(array[i][0]);
    sb.append('</td><td>');
    sb.append(array[i][1]);
    sb.append('</td><td align="right">');
    sb.append(jsFN(jsR(jsN(array[i][2]), 1)));
    sb.append(' h</td></tr>');
}
}
```

## 3 Intermediates

**i** Javascript code defined in control events must contain a trailing semicolon.

✕
Radio

---

Obecné	Hodnoty	<b>Události</b>	Tlačítko	Nápověda	Filtr	Práva	Administrace
--------	---------	-----------------	----------	----------	-------	-------	--------------

**OnClick**

?

Uložit
Smazat

```

= t0 ></div></td><td><div class= tx style= width:158px ></div></td><td style= display:none; ><div class= tx ></div><
"100%"><label id="C9_lbl"* Status</label></td><td class="fmt"><table id="C9_Table2" cellspacing="0" cellpadding="0">
"0" onclick="if (event == null || srcElement2(event).tagName == &#39;INPUT&#39;) { form_Change(); hotovo();}" style='
e="Zadany#en:Assigned" checked="checked" tabindex="3"><label for="C9_rbl_0">Assigned</label></td></tr><tr><td><input :
  
```

**i** The only exception is the javascript code used in the "ActionButtons" control for the "Save", "Delete" and "Back" buttons. This javascript, on the other hand, must not contain a semicolon at the end, because it is used in the condition as a result of the call in the source code of the resulting web page.

✕
ActionButtons

---

Obecné	<b>Uložit</b>	Smazat	Zpět
--------	---------------	--------	------

<b>JavaScript</b>	<input type="text" value="beforeSave()"/>	<span>?</span> <i>Název funkce bez středníku na konci, která se spustí po stisknutí tlačítka „Uložit“</i>
<b>OnBeforeSave</b>	<b>Upravit skript (20)</b> <input type="checkbox"/> Zakázat skript	?
<b>OnAfterSave</b>	<b>Upravit skript (31)</b> <input type="checkbox"/> Zakázat skript	?
<b>OnBeforeOpen</b>	<b>Upravit skript (1)</b> <input type="checkbox"/> Zakázat skript	?

Uložit

```
width: 80px; height: 20px; border: 1px solid #ccc; margin-bottom: 5px; border-radius: 5px; text-align: center; font-size: 12px; color: #333; background-color: #fff; cursor: pointer; display: inline-block; padding: 2px 10px; text-decoration: none; font-family: sans-serif; font-weight: normal;"><div class="fx"></div></td><td class="f1"><div class="f0"></div></td><td><div class="f1"></div></td></tr><tr><td class="fx"><input type="submit" name="btSave" value="Save" onclick="if (!beforeSave()) return false;" id="btSave" style="width:85px;"><input type="submit" name="btDelete" value="Delete" onclick="if (this.disabled) return false;" id="btDelete" style="width:85px;"></td></tr></table>
```

## 4 Buttons and postbacks

- i** Buttons that are set as “Script”, “Print to template”, “Open view page” or “Open edit form” first run the javascript code defined in the “Client” field (may or may not be defined), trigger a postback or submit the form with the “POST” method (sending all the data filled in the form including sending the information about the button that triggered the postback), and on the server side it will run the defined server script. After the script completes, the user is returned to the form, until then the form content is hidden, and the text “Please wait...” is displayed at the top of the form.

Obecné
**Událost**
Nápověda
Práva
Administrace

**Klient**

**Server**

(Nic)

Otevřít editační formulář

Otevřít nahlížeč stránku

Tisk do šablony

Skript

**Možnosti**

1. Uložit záznam

2. Spustit skript

**Upravit skript (0)**

3. Uložit záznam

? Seznam javascriptových příkazů, které se vykonají po stisknutí tlačítka

?

Uložit
Kopírovat
Smazat

- i** Clicking the button can be called programmatically using the javascript function “bt\_Click(string id)”. A detailed description of javascript functions is given in the separate manual “JavaScript Functions”. Only one postback can be called from javascript, i.e. clicking only one button at a time. Calling the “bt\_Click” function (and thus the postback) multiple times in a row is evaluated as an error, and NET Genium will only call the first postback in the sequence.
- i** The javascript function “form\_Update()” is used to update the form, which also uses postback and the hidden “btUpdate” button to update all control elements in the form.