

# Pravidla pro psaní javascriptu Framework NET Genium

# Obsah

1	Mezery, tabulátory a přechody na nové řádky.....	3
2	Jednoduché apostrofy.....	4
3	Středníky .....	5
4	Tlačítka a postbacky .....	7

# 1 Mezery, tabulátory a přechody na nové řádky

- 1 Mezery používáme mezi „if“ a levou závorkou, nalevo a napravo od znaménka „=“, za čárkou jakožto oddělovačem proměnných nebo parametrů funkcí, apod.
- 1 Tabulátory používáme pro odsazení kódu od levého kraje nové řádky. Tabulátor zastupuje dvě mezery po sobě, a je zkratkovou klávesou pro jejich vložení do kódu v editoru javascriptu. Každé zanoření kódu vyžaduje jeden tabulátor – dvě mezery. Zdrojový kód na první úrovni neodsazujeme.
- 1 Entery – přechody na nový řádek – používáme před levou závorkou „{“ i za ní, stejně tak před pravou závorkou „}“ i za ní.

```
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();
}
```

## ⚠ Špatný příklad

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

## 2 Jednoduché apostrofy

- ❶ Jako výchozí oddělovač vždy používáme jednoduché apostrofy. Protože je možné kombinovat javascriptový kód se serverovými funkcemi, vždy si necháváme dvojité uvozovky v záloze pro volání serverových funkcí.
- ❶ Parametry serverových funkcí musí být často umístěny ve dvojitých uvozovkách, protože mohou v hodnotě parametru obsahovat čárku, která se standardně považuje za oddělovač jednotlivých parametrů. Texty uvnitř dvojitých uvozek se považují za stringové konstanty, a nedochází v nich k nahrazování serverových funkcí, pouze proměnných. Jedinou výjimkou jsou parametry funkcí „SQL“ a „SQLARRAY“, uvnitř kterých je možné volat serverové funkce, přestože jsou parametry těchto funkcí obaleny dvojitými uvozovkami.

```
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 Středníky

**i** Javascriptový kód definovaný v událostech ovládacích prvků musí obsahovat středník na konci.

✕
Radio

Obecné

Hodnoty

Události

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** Jedinou výjimkou je javascriptový kód používaný v ovládacím prvku „ActionButton“ u tlačítek „Uložit“, „Smazat“ a „Zpět“. Tento javascript naopak středník na konci obsahovat nesmí, protože je ve zdrojovém kódu webové výsledné stránky používán v podmínce jako výsledek volání.

✕
ActionButtons

Obecné

Uložit

Smazat

Zpět

**JavaScript**

**OnBeforeSave**

**OnAfterSave**

**OnBeforeOpen**

?

*Název funkce bez středníku na konci, která se spustí po stisknutí tlačítka „Uložit“*

?

**Upravit skript (20)**

Zakázat skript

?

**Upravit skript (31)**

Zakázat skript

?

**Upravit skript (1)**

Zakázat skript

Uložit



## 4 Tlačítka a postbacky

- i** Tlačítka, která jsou nastavená jako „Skript“, „Tisk do šablony“, „Otevřít nahlížecí stránku“ nebo „Otevřít editační formulář“, nejdříve spustí javascriptový kód definovaný v poli „Klient“ (může i nemusí být definovaný), vyvolají postback resp. submit formuláře metodou „POST“ (odeslání všech dat vyplněných ve formuláři včetně odeslání informace o tlačítku, které postback vyvolalo), a na straně serveru spustí definovaný serverový skript. Po dokončení skriptu je uživatel vrácen zpět do formuláře, do té doby je obsah formuláře skrytý, a v horní části formuláře se zobrazuje text „Prosím čekejte...“.

Obecné
Událost
Nápověda
Práva
Administrace

**Klient**

**Server**

(Nic)

Otevřít editační formulář

Otevřít nahlížecí 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** Kliknutí na tlačítko je možné vyvolat programově pomocí javascriptové funkce „bt\_Click(string id)“. Detailní popis javascriptových funkcí je uveden v samostatné příručce „JavaScriptové funkce“. Z javascriptu je možné zavolat pouze jeden postback, tedy kliknutí pouze na jedno tlačítko v jednu chvíli. Zavolání funkce „bt\_Click“ (a tedy i postbacku) vícekrát po sobě je vyhodnoceno jako chyba, a NET Genium vyvolá pouze první postback v pořadí.
- i** Pro aktualizaci formuláře slouží javascriptová funkce „form\_Update()“, která používá též postback a skryté tlačítko „btUpdate“ pro aktualizaci všech ovládacích prvků ve formuláři.