

Fórum:Common.js (dokumentace)

Základní informace

Skript MediaWiki:Common.js se spouští při každém načtení kterékoli stránky ve WikiSkriptech. Je tudíž vhodný pro funkce, které se mají vykonat na každé stránce, a pro inicializaci vedlejších skriptů.

Javascriptový kód lze vkládat pouze do jmenného prostoru **MediaWiki** a jeho název musí končit příponou **.js**. Pro správné spuštění vedlejšího skriptu je nutné přidat do MediaWiki:Common.js řádek

```
importScript("MediaWiki:Název skriptu.js");
```

Obecné funkce

V tomto oddílu by měly být funkce vhodné k použití kdekoli a kdykoli.

Funkce `getElementsByClass()`

```
// FUNKCE VRACEJÍCÍ POLE VŠECH ELEMENTŮ
// označených danou třídou (searchClass)
// s možností označit oblast hledání (domNode)
// a html značku (tagName)
function getElementsByClass(searchClass, domNode, tagName) {
    if (domNode == null) domNode = document;
    if (tagName == null) tagName = '*';
    var el = new Array();
    var tags = domNode.getElementsByTagName(tagName);
    var tcl = " "+searchClass+" ";
    for(i=0,j=0; i<tags.length; i++) {
        var test = " " + tags[i].className + " ";
        if (test.indexOf(tcl) != -1)
            el[j++] = tags[i];
    }
    return el;
}
```

Tato funkce vrací pole všech elementů na stránce, které patří do zadané CSS třídy (pomocí HTML atributu – vlastnosti **class**).

Funkce má tři parametry, z nichž je povinný pouze první, **searchClass**. Tento parametr udává jméno třídy, jejíž elementy na stránce hledáme. Druhý parametr, **domNode**, charakterizuje oblast hledání (můžeme hledat v celém dokumentu nebo jen jeho části). Třetí parametr **tagName** určuje, které konkrétní HTML značky hledáme.

Funkce `getXMLHttpRequest`

```
// FUNKCE ZÍSKÁVAJÍCÍ ODPOVĚĎ SERVERU
// xmlvar = proměnná určená ke komunikaci se serverem
// url = URL stránky k načtení
// todo = funkce spouštějící se po získání odpovědi
function getXMLHttpRequest(xmlvar, url, todo) {
    eval(xmlvar + ' = null;'); // definice

    if (window.XMLHttpRequest) {
        // pro nové prohlížeče:
        eval(xmlvar + ' = new XMLHttpRequest();');
    } else if (window.ActiveXObject) {
        // pro IE5 a IE6
        eval(xmlvar + ' = new ActiveXObject("Microsoft.XMLHTTP");');
    }

    eval('xmlvarValue = ' + xmlvar + ');');

    if (xmlvarValue!=null) {
        eval(xmlvar+'.onreadystatechange=' + todo + ');');
        eval(xmlvar+'.open("GET",url,true);');
        eval(xmlvar+'.send(null);');
    } else {
        window.alert("Chyba: Váš prohlížeč nepodporuje XMLHTTP. Některé stránky nemusí být zobrazeny správně.");
    }
}
```

Funkce `getXMLHttpRequest()` kontaktuje server na adrese **url** zadané v parametru a získává jeho odpověď. Tu poté zpracuje pomocí funkce zadané ve druhém parametru **todo**. V případě chyby spojení vypíše chybovou hlášku. Výpis odpovědi je pak dostupný v objektu uvedeném v parametru **xmlvar**.

Funkce zpracovávající odpověď serveru by měla obsahovat následující kód (`xmlhttpvar` by mělo být nahrazeno vlastní proměnnou – doporučuji název začínající právě `xmlhttp`, viz příklad níže):

```
function xy() {
    if (xmlhttpvar.readyState==4) { // 4 = "loaded"
```

```

if (xmlhttpvar.status==200) {// 200 = OK

    var htmlRes = xmlhttpvar.responseText; // odpoved serveru

    // (...) zpracovani odpovedi

} else {
    window.alert("Nepodarilo se spojit se serverem."); // chybova hlaska
}
}
}

```

Příklad spuštění této funkce při načtení stránky:

```

var xmlhttpvar; // definována proměnná

$(function(){if (wgNamespaceNumber != 0) return; getXMLHttpRequest("xmlhttpPortal", wgServer + "/api.php", "odkazujiciPortaly");});

```

Funkce trim()

```

// FUNKCE ODSTRAŇUJÍCÍ MEZERY NA ZAČÁTKU A KONCI ŘETĚZCE
function trim(str) {
    str = str.replace(/^\s+|\s+$/g, "");
    return str;
}

```

Tato funkce oseká řetězec znaků vložený do jejího parametru od prázdných znaků na začátku a na konci řetězce.

Příklad

```

retezec = "    Krysomil hrbatohlavý    šel na houby    /    ";

retezec = trim(retezec); // osekání řetězce

document.write(retezec); // nyní: "Krysomil hrbatohlavý    šel na houby    /"

```

Funkce mwGetUserRight(right)

```

// FUNKCE PODÁVÁ INFORMACI O TOM, ZDA JE UŽIVATEL V DANÉ SKUPINĚ (right)
// vrací true, pokud ano
function mwGetUserRight(right) {
    userrights = mw.config.get('wgUserGroups');
    userrights = userrights.join();
    output = userrights.search(right);
    if (output<0) return false;
    else return true;
}

```

Tato funkce testuje, zda aktuálně přihlášený uživatel patří do dané skupiny uživatelů (tj. mezi správce, administrátory, učitele, redaktory, apod.). Požadované oprávnění se vkládá jako parametr funkce.

Příklad

```

if (mwGetUserRight("editor")) {
    // pokud je uživatel redaktorem, spustí se tato část kódu
} else {
    // pokud uživatel nepatří mezi redaktory, spustí se tato část kódu
}

```

Funkce pro obnovení posledních změn

Funkce obnoví stránku s posledními změnami každých 10 minut. Je vhodné pro dlouhodobé pohodlné sledování posledních změn, aniž by rušilo při práci.

```

// OBNOVA POSLEDNÍCH ZMĚN KAŽDÝCH 10 MINUT
if (wgPageName == "Speciální:Poslední změny") {
    window.setInterval("location.reload();", 600000);
}

```

Překlad stránek pomocí Google Translate

```

function googleTranslateElementInit() {
    new google.translate.TranslateElement({
        pageLanguage: 'auto',
        includedLanguages: 'en,cs,sk',
        autoDisplay: false,
        gaTrack: true,
        gaId: 'UA-4765930-7',
    });
}

mw.loader.load("http://translate.google.com/translate_a/element.js?cb=googleTranslateElementInit", "text/javascript");

```

Kód přidá záložku pro překlad stránek pomocí Google Translate (<http://translate.google.com>). Kód je vygenerovaný pomocí Google (http://translate.google.com/translate_tools) a upravený do podoby, která funguje v MediaWiki.

Funkce `mw.loader.load` je určena k načítání externích zdrojů JS nebo CSS. Je podrobně popsána v dokumentaci MediaWiki (https://www.mediawiki.org/wiki/ResourceLoader/Default_modules#mediaWiki.loader).

Vložené skripty

Zde jsou vloženy vedlejší JavaScripty pomocí funkce `importScript()`, která je určena k načítání interních JavaScriptových kódů.

```
/* ===== Vložené skripty ===== */
//skript pro hledání článků, které nemají odkaz na portálech
if (mw.getUserRight("editor")) importScript("MediaWiki:MissingPortalLinks.js");

// skript pro upload souborů
if (wgPageName == "Speciální:Načíst_soubor") importScript("MediaWiki:UploadForm.js");

// skript kontrolující editace nových uživatelů
if (wgAction != "view") importScript("MediaWiki:KontrolaEditace.js");

// skript přidávající editační klávesové zkratky
if (wgAction != "view") importScript("MediaWiki:EditacniKlavesoveZkratky.js");

// úprava osobního menu
importScript("MediaWiki:UpravyMenu.js");

// skript pro zobrazení disclaimeru a smazání příp. cookie při odhlášení
if (wgAction == "view") importScript("MediaWiki:Disclaimer.js");

// sbalování částí sekcí portálů
importScript("MediaWiki:SbalovaniPortalu.js");

// nový vzhled - oranžová lišta, odkazy na editace, vyhledávací pole, překlady...
importScript("MediaWiki:MenuLista.js");

// rolující seznam žádaných článků
// jen pro novější prohlížeče
if (window.XMLHttpRequest) {
    importScript("MediaWiki:ScrollingList.js");
} else {
    // schovat box "žádané články"
    $(function() {
        if (document.getElementById("wanted-pages-mp-box")) {
            // schovat box
            document.getElementById("wanted-pages-mp-box").style.display = "none";
        }
    });
}
```

Skrývání části hlavní stránky

Tato část obsahuje kód, který zaručuje skrývání části oborů na hlavní stránce.

```
// přidání onclick na "rozbal všechny obory"
$( '#rozbalObory' ).click( function() {
    $( '.skryteObory' ).fadeToggle('slow');
    $( '#rozbalObory' ).css('display','none');
    $( '#sbalObory' ).css('display','block');
    $( '.viditelneObory' ).removeClass('viditelneObory').addClass('viditelneOboryNoGrad');
});

// přidání onclick na "sbal obory"
$( '#sbalObory' ).click( function() {
    $( '.skryteObory' ).fadeToggle('slow',function () {
        $( '#rozbalObory' ).css('display','block');
        $( '#sbalObory' ).css('display','none');
        $( '.viditelneOboryNoGrad' ).addClass('viditelneObory').removeClass('viditelneOboryNoGrad');
    });
});
```