


Serverové funkce

Framework NET Genium

double ABS(double number)


 **Použití:** HTML, Skript

 **Popis:** Funkce vrací absolutní hodnotu z hodnoty zadané parametrem „number“.

 **Příklad:**

```
ABS(-1.23)
// Vrací „1,23“
ABS(#A#)
```

DateTime ADDDAYS(DateTime date, double days)

 **Použití:** HTML, JavaScript


 **Popis:** Funkce přičte k datu zadanému parametrem „date“ počet dnů definovaných parametrem „days“.

 **Příklad:**

```
ADDDAYS(1.1.2000, -1.2)
// Vrací „30.12.1999 19:12“
ADDDAYS(1.1.2000, 0)
// Vrací „01.01.2000“
ADDDAYS(1.1.2000, 2)
// Vrací „03.01.2000“
ADDDAYS(#today#, 0)
```

DateTime ADDDAYS(DateTime date, double days, int skipholidays)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce přičte k datu zadanému parametrem „date“ počet dnů definovaných parametrem „days“. Parametr „skipholidays“, který může být jakékoliv celé číslo, značí, že budou přeskočeny svátky. Je-li tedy výsledné datum statní svátek, sobota nebo neděle, přičte dalších „skipholidays“ dnů.

 **Příklad:**

```
ADDDAYS(1.1.2000, -9.2, -2)
// Vrací „22.12.1999 19:12“
ADDDAYS(1.1.2000, 0, 1)
// Vrací „03.01.2000“
ADDDAYS(1.1.2000, 12, 0)
// Vrací „13.01.2000“
ADDDAYS(#today#, 0, 3)
```

DateTime ADDDAYS(DateTime date, double days, int skipolidays, string satsun)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce přičte k datu zadanému parametrem „date“ počet dnů definovaných parametrem „days“. Parametr „skipolidays“, který může být jakékoliv celé číslo, značí, že budou přeskočeny svátky. Je-li tedy výsledné datum statní svátek, sobota nebo neděle, přičte dalších „skipolidays“ dnů. Defaultně jsou přeskočeny soboty a neděle, pro zamezení přeskočení soboty nebo neděle je třeba jako parametr „satsun“ uvést text „sat“ nebo „sun“.

 **Příklad:**

```
ADDDAYS(1.1.2000, -9.2, -2, sat)
// Vrací „22.12.1999 19:12“
ADDDAYS(1.1.2000, 0, 1, sun)
// Vrací „02.01.2000“
ADDDAYS(1.1.2000, 12, 0, sat)
// Vrací „13.01.2000“
ADDDAYS(#today#, 0, 3, sun)
```

DateTime ADDDAYS(DateTime date, double days, int skipolidays, string sat, string sun)

 **Použití:** HTML, JavaScript, Skript

 **Popis:** Funkce přičte k datu zadanému parametrem „date“ počet dnů definovaných parametrem „days“. Parametr „skipolidays“, který může být jakékoliv celé číslo, značí, že budou přeskočeny svátky. Je-li tedy výsledné datum statní svátek, sobota nebo neděle, přičte dalších „skipolidays“ dnů. Defaultně jsou přeskočeny soboty a neděle, pro zamezení přeskočení obou těchto dní je třeba jako parametry „sat“ a „sun“ uvést text „sat“ a „sun“.

 **Příklad:**

```
ADDDAYS(1.1.2000, -9.2, -2, sat, sun)
// Vrací „22.12.1999 19:12“
ADDDAYS(1.1.2000, 0, 1, sat, sun)
// Vrací „02.01.2000“
ADDDAYS(1.1.2000, 12, 0, sat, sun)
// Vrací „13.01.2000“
ADDDAYS(#today#, 0, 3, sat, sun)
```

DateTime ADDHOURS(DateTime date, double hours)

 **Použití:** HTML, JavaScript, Skript

 **Popis:** Funkce přičte k datu zadanému parametrem „date“ počet hodin definovaných parametrem „hours“.

 **Příklad:**

```
ADDHOURS(1.1.2000, -1.5)
// Vrací „31.12.1999 22:30“
ADDHOURS(1.1.2000, 0)
// Vrací „01.01.2000“
ADDHOURS(1.1.2000, 5)
// Vrací „01.01.2000 05:00“
ADDHOURS(#now#, 10)
```

DateTime ADDMINUTES(DateTime date, double minutes)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce přičte k datu zadanému parametrem „date“ počet minut definovaných parametrem „minutes“.

 **Příklad:**

```
ADDMINUTES(1.1.2000, -2.5)
// Vrací „31.12.1999 23:57“
ADDMINUTES(1.1.2000, 0)
// Vrací „01.01.2000“
ADDMINUTES(1.1.2000, 2)
// Vrací „01.01.2000 00:02“
ADDMINUTES(#now#, 0)
```

DateTime ADDMONTHS(DateTime date, int months)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce přičte k datu zadanému parametrem „date“ počet měsíců definovaných parametrem „months“. Zadáání desetinného čísla v parametru „months“ způsobí převod na datový typ „int“, nedochází k zaokrouhlování, ale k oříznutí čísel za desetinnou čárkou.

 **Příklad:**

```
ADDMONTHS(1.1.2000, -2.5)
// Vrací „01.11.1999“
ADDMONTHS(1.1.2000, 0)
// Vrací „01.01.2000“
ADDMONTHS(1.1.2000, 2)
// Vrací „01.03.2000“
ADDMONTHS(#now#, -2)
```

DateTime ADDTIMESPANINWORDS(DateTime date, string expression1, string expression2, ...)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce přičte k datu zadanému parametrem „date“ časový údaj vyjádřený slovním popisem definovaným parametrem „expression1“, „expression2“, atd. Funkce může obsahovat libovolný počet parametrů, který je větší nebo roven 2. Slovní popis časového údaje je možné definovat libovolným jazykem, který je součástí NET Genia.

 **Příklad:**

```
ADDTIMESPANINWORDS (1.1.2000, 2 months)
// Vrací „01.03.2000“
ADDTIMESPANINWORDS (1.1.2000, 1 year)
// Vrací „01.01.2001“
ADDTIMESPANINWORDS (1.1.2000, 1 year, 2 months, 1 day)
// Vrací „02.03.2001“
ADDTIMESPANINWORDS (1.1.2000, 1 den)
// Vrací „02.01.2000“
ADDTIMESPANINWORDS(1.1.2000 12:05, 1 rok, 2 měsíce, 1 den, 5 minut, 28 vteřin)
// Vrací 02.03. 2001 12:10
```

DateTime ADDWORKDAYS(DateTime date, int days)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce přičte k datu zadanému parametrem „date“ počet pracovních dnů definovaných parametrem „days“. Zadáání desetinného čísla v parametru „days“ způsobí převod na datový typ „int“, nedochází k zaokrouhlování, ale k oříznutí čísel za desetinnou čárkou.

 **Příklad:**

```
ADDWORKDAYS(1.1.2000, -2.5)
// Vrací „30.12.1999“
ADDWORKDAYS(1.1.2000, 0)
// Vrací „01.01.2000“
ADDWORKDAYS(1.1.2000, 2)
// Vrací „04.01.2000“
ADDWORKDAYS(#now#, -2)
```

DateTime ADDYEARS(DateTime date, int years)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce přičte k datu zadanému parametrem „date“ počet roků definovaných parametrem „years“. Zadáání desetinného čísla v parametru „years“ způsobí převod na datový typ „int“, nedochází k zaokrouhlování, ale k oříznutí čísel za desetinnou čárkou.

 **Příklad:**

```
ADDYEARS(1.1.2000, -2.5)
// Vrací „01.01.1998“
ADDYEARS(1.1.2000, 0)
// Vrací „01.01.2000“
ADDYEARS(1.1.2000, 2)
// Vrací „01.01.2002“
ADDYEARS(#today#, 2)
```

double ARCCOS(double number)

 **Použití:** HTML, Skript


 **Popis:** Funkce vrací arcus cosinus z hodnoty zadané parametrem „number“.

 **Příklad:**

```
ARCCOS(0)
// Vrací „1,5707963267949“
ARCCOS(2)
// Vrací „Není číslo“
ARCCOS(#A#)
```

double ARCSIN(double number)


 **Použití:** HTML, Skript


 **Popis:** Funkce vrací arcus sinus z hodnoty zadané parametrem „number“.

 **Příklad:**

```
ARCSIN(0)
// Vrací „1“
ARCSIN(-1)
// Vrací „-1,5707963267949“
ARCSIN(#A#)
```

double ARCTAN(double number)

 **Použití:** HTML, Skript

 **Popis:** Funkce vrací arcus tangens z hodnoty zadané parametrem „number“.

 **Příklad:**

```
ARCTAN(-0.5)
// Vrací „-0,463647609000806“
ARCTAN(3)
// Vrací „1,24904577239825“
ARCTAN(#A#)
```

string ATTACHMAILSIGNATURE(string body, string signature)

Použití: HTML, JavaScript, Skript

Popis: Funkce vloží do odesílané e-mailové zprávy zadané parametrem „body“ e-mailový podpis definovaný parametrem „signature“. Pokud funkce vyhodnotí, že jsou v odesílané zprávě nějaké e-mailové odpovědi, vloží e-mailový podpis před tyto odpovědi. V opačném případě je emailový podpis vložen na konec e-mailové zprávy.

Příklad:

```
ATTACHMAILSIGNATURE(abc, Best regards)
```

int ATTACHMENT2PDF(int file)

Použití: Skript

Popis: Funkce zkonvertuje souborovou přílohu zadanou parametrem „file“ do formátu „pdf“. Funkce přijímá ID souborové přílohy ve formátu „doc“, „docx“, „xls“ nebo „xlsx“, a vrátí ID nově vytvořené souborové přílohy ve formátu „pdf“.

Příklad:

```
ATTACHMENT2PDF(#ng_file#)
```

int ATTACHMENT2PDF(string file)

Použití: Skript


Popis: Funkce zkonvertuje souborovou přílohu zadanou parametrem „file“ do formátu „pdf“. Funkce přijímá úplnou cestu souborové přílohy ve formátu „doc“, „docx“, „xls“ nebo „xlsx“, a vrátí ID nově vytvořené souborové přílohy ve formátu „pdf“.

Příklad:

```
ATTACHMENT2PDF(#rootpath#Temp\file.doc)  
ATTACHMENT2PDF(#rootpath#Temp\file.xls)
```


string ATTACHMENTCONTENT(int file)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací obsah souborové přílohy zadané parametrem „file“ ve formě textového řetězce.

 **Příklad:**

```
ATTACHMENTCONTENT(#ng_file#)
```

string ATTACHMENTDIMENSIONS(int image)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací rozměry souborové přílohy – obrázku – zadané parametrem „image“ ve formátu „šířka;výška“.

 **Příklad:**

```
ATTACHMENTDIMENSIONS(#ng_image#)
```

string ATTACHMENTGALLERY(string attachments, int maxwidth, int maxheight)

 **Použití:** HTML


 **Popis:** Funkce zobrazí v seznamu pod sebou zmenšené obrázky nebo odkazy na souborové přílohy. Souborové přílohy jsou zadané parametrem „attachments“ jako seznam ID souborových příloh oddělených středníkem. Parametr „maxwidth“ a „maxheight“ určuje maximální šířku a výšku zmenšeného obrázku v pixelech.

 **Příklad:**

```
ATTACHMENTGALLERY(1;1;1, 120, 120)
```

string ATTACHMENTIMG(int file)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací HTML kód obrázku s přílohou zadanou parametrem „file“.

 **Příklad:**

```
ATTACHMENTIMG(#ng_soubor#)
// Vrací „“
```

string ATTACHMENTNAME(int file, bool extension)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací název souboru včetně přípony souborové přílohy zadané parametrem „file“. Volitelný parametr „extension“ udává, zda má funkce vracet název souboru včetně přípony souborové přílohy, nebo jen název souboru bez přípony.

 **Příklad:**

```
ATTACHMENTNAME(#ng_soubor#)
ATTACHMENTNAME(#ng_soubor#, true)
```

string ATTACHMENTURL(int file)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací privátní internetovou adresu souborové přílohy zadané parametrem „file“, funkční pouze pro aktuálně přihlášeného uživatele, s časovou platností omezenou na 24 hodin.

 **Příklad:**

```
ATTACHMENTURL(1)
// Vrací „Download.aspx?hWiPNy0VInyQJbufLAlpTA=="
ATTACHMENTURL(#ng_soubor#)
```

string ATTACHMENTURL(int file, string url)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací veřejnou internetovou adresu souborové přílohy zadané parametrem „file“ s neomezenou časovou platností.

 **Příklad:**

```
ATTACHMENTURL(1, https://www.netgenium.com)
// Vrací „https://www.netgenium.com/Download.aspx?hWiPNy0VInyQJbufLA1pTA==“
ATTACHMENTURL(#ng_soubor#, https://www.netgenium.com)
```

string ATTACHMENTVIDEO(int file, int maxwidth, int maxheight)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací HTML kód videa s přílohou zadanou parametrem „file“. Parametr „maxwidth“ a „maxheight“ určuje maximální šířku a výšku videa v pixelech.

 **Příklad:**

```
ATTACHMENTVIDEO(#ng_soubor:value#, 1000, 500)
// Vrací „<video><source src="Download.aspx?hWiPNy0VInyQJbufLA1pTA=="></video>“
```

double AVERAGE(object[] array)

 **Použití:** Skript


 **Popis:** Funkce vrací průměr hodnot skriptové proměnné zadané parametrem „array“, je-li proměnná polem hodnot. Není-li proměnná polem hodnot, vrací funkce hodnotu proměnné.

 **Příklad:**

```
AVERAGE (#A#)
```

string BASE64DECODE(string base64)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací dekódovaný řetězec z textového řetězce zadaného parametrem „base64“.

 **Příklad:**

```
BASE64DECODE (YWJj)
// Vrací „abc“
```

string BASE64ENCODE(string value)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací zakódovaný řetězec ve formátu „base64“ z textového řetězce zadaného parametrem „value“.

 **Příklad:**

```
BASE64ENCODE (abc)
// Vrací „YWJj“
```

string BUTTON(string title, string question, string button, string key)

 **Použití:** Nahlížecí tabulka

 **Popis:** Funkce zobrazí uživatelsky definované tlačítko v nahlížecí tabulce. Nadpis tlačítka je definován parametrem „title“. Parametr „question“ určuje otázku, která se zobrazí v potvrzovacím okně typu „confirm“ po stisknutí tlačítka. Parametr „button“ představuje název skrytého obslužného tlačítka, které se musí nacházet na stejné nahlížecí stránce nebo editačním formuláři jako nahlížecí tabulka se zobrazovaným tlačítkem. Po stisknutí tlačítka v nahlížecí tabulce a následném potvrzení otázky je programově vyvoláno stisknutí obslužného tlačítka, které spouští skript na straně serveru. Parametr „key“ definuje označení záznamu, u kterého bylo tlačítko v nahlížecí tabulce stisknuto. Parametr „key“ slouží zejména k možnosti získat ID daného záznamu v rámci skriptu pomocí další serverové funkce „REQUESTFORM(abc)“. Uvedenou funkci lze dále ve skriptu využít například v přiřazení do proměnné („A = Integer(REQUESTFORM(abc))“ nebo v podmínce SQL dotazu. Pro zobrazení tlačítka je nutné, aby volání funkce „BUTTON“ bylo uloženo v hodnotě ovládacího prvku „TextBox“, nastaveného „Jen ke čtení“ a jako „Skryté pole“.

 **Příklad:**

```
BUTTON(Vymazat, Opravdu chcete záznam smazat?, DeleteItem, abc) // V hodnotě ovládacího prvku
„TextBox“
REQUESTFORM(abc) // Ve skriptu obslužného tlačítka
```

string CAPTCHA()

 **Použití:** HTML v editačním formuláři

 **Popis:** Funkce vloží do HTML kódu editačního formuláře ověřovací kód captcha, který slouží pro odlišení skutečných uživatelů od robotů. Funkce je svázána s funkcí „EVALCAPTCHA“, bez níž by funkce „CAPTCHA“ neměla smysl.

 **Příklad:**

```
CAPTCHA ()
```

double CEILING(double number)

 **Použití:** HTML, Skript


 **Popis:** Funkce vrací nejmenší celé číslo, které je větší nebo rovno hodnotě zadané parametrem „number“.

 **Příklad:**

```
CEILING (-1.23)
// Vrací „-1“
CEILING (1.23)
// Vrací „2“
```

string COLORLEGEND()

 **Použití:** HTML


 **Popis:** Funkce vrací překlad slova „LEGENDA“ na základě jazykového nastavení aktuálně přihlášeného uživatele.

 **Příklad:**

```
COLORLEGEND ()
```

string COLORLEGEND(string text)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací překlad slova „LEGENDA (text)“ na základě jazykového nastavení aktuálně přihlášeného uživatele, společně s textem, který definuje parametr „text“. Tento text může být vícejazyčný.

 **Příklad:**

```
COLORLEGEND(Neuhrazené faktury#en:Unpaid)
```

string COLORLEGEND(string color, string text)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací obdélníček vyplněný barvou zadanou parametrem „color“ v HEX formátu „#ffffff“ s textovým popisem zadaným parametrem „text“. Tento text může být vícejazyčný.

 **Příklad:**

```
COLORLEGEND(#44FF0A, Neuhrazené faktury#en:Unpaid)
```

string COLORLEGEND(int R, int G, int B, string text)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací obdélníček vyplněný barvou zadanou parametry RGB v desítkové soustavě s textovým popisem zadaným parametrem „text“. Tento text může být vícejazyčný.

 **Příklad:**

```
COLORLEGEND(255, 100, 100, Neuhrazené faktury#en:Unpaid)
```

string COLORLINK(string text, string url)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací barevnou šipku a text zadaný parametrem „text“ s odkazem zadaným parametrem „url“. Pokud je na odkaz kliknuto, cílová webová stránka se otevře ve stávajícím rámu stránky.

 **Příklad:**

```
COLORLINK(abc, ViewPage.aspx?viewpage=878)
// Otevře nahlížečí stránku s ID 878 ve stávajícím rámu stránky
COLORLINK(abc, Form.aspx?form=396)
//Otevře editační formulář s ID 396 ve stávajícím rámu stránky
```

string COLORLINK(string text, string url, bool center)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací barevnou šipku a text zadaný parametrem „text“ s odkazem zadaným parametrem „url“. Parametr „center“ říká, že odkaz umístěný v portletech bude zarovnaný na střed. Pokud je na odkaz kliknuto, cílová webová stránka se otevře ve stávajícím rámu stránky.

 **Příklad:**

```
COLORLINK(abc, ViewPage.aspx?viewpage=878, true)
// Otevře nahlížečí stránku s ID 878 ve stávajícím rámu stránky
COLORLINK(abc, Form.aspx?form=396, true)
//Otevře editační formulář s ID 396 ve stávajícím rámu stránky
```

string COLORLINK2(string text, string url)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací barevnou šipku a text zadaný parametrem „text“ s odkazem zadaným parametrem „url“. Pokud je na odkaz kliknuto, cílová webová stránka se otevře v novém okně.

 **Příklad:**

```
COLORLINK2(abc, ViewPage.aspx?viewpage=878)
// Otevře nahlížečí stránku s ID 878 v novém okně
COLORLINK2(abc, Form.aspx?form=396)
//Otevře editační formulář s ID 396 v novém okně
```

string COLORLINK2(string text, string url, bool center)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací barevnou šipku a text zadaný parametrem „text“ s odkazem zadaným parametrem „url“. Parametr „center“ říká, že odkaz umístěný v portletech bude zarovnaný na střed. Pokud je na odkaz kliknuto, cílová webová stránka se otevře v novém okně.

 **Příklad:**

```
COLORLINK2(abc, ViewPage.aspx?viewpage=878, true)
// Otevře nahlížečící stránku s ID 878 v novém okně
COLORLINK2(abc, Form.aspx?form=396, true)
//Otevře editační formulář s ID 396 v novém okně
```

void COLUMN(int columnWidth, int labelWidth, int defaultControlWidth)

 **Použití:** Název ovládacího prvku HTML – tělo ovládacího prvku musí zůstat prázdné

 **Popis:** Funkce zahájí blok ovládacích prvků, které jsou umístěny ve sloupcích a používají reverzní design – ovládací prvky nemají rámeček, mají nastavené barevné pozadí, a při focusu zobrazují barevnou lištu pod ovládacím prvkem. Celkovou šířku sloupce definuje parametr „columnWidth“ šířku sloupce s názvem ovládacího prvku definuje parametr „labelWidth“. Při nastavení hodnoty parametru „labelWidth“ se sloupec s názvem ovládacího prvku nezobrazuje, a ovládací prvky začínají úplně odleva. Výchozí velikost ovládacích prvků definuje parametr „defaultControlWidth“, a má význam u ovládacích prvků CheckBox a Radio, které jsou v reverzním designu umístěny stejně jako ostatní ovládací prvky v barevném boxu.

 **Příklad:**

```
COLUMN(500, 100, 150)
```

void /COLUMN

 **Použití:** Název ovládacího prvku HTML – tělo ovládacího prvku musí zůstat prázdné


 **Popis:** Funkce zakončí blok ovládacích prvků, které jsou umístěny ve sloupcích a používají reverzní design.

 **Příklad:**

```
/COLUMN
```


string COMPANYID(string company)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací název společnosti zadaný parametrem „company“ bez typu společnosti (např. „s.r.o.“, „a.s.“ atd.), mezer, pomlček, čárek a teček a dále převedený na velká písmena.

 **Příklad:**

```
COMPANYID(NetGenium s.r.o.)  
// Vrací „NETGENIUM“
```

string COMPAREDATE(DateTime value1, string operator, DateTime value2, string yes, string no)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací hodnotu parametru „yes“, je-li splněno porovnání hodnoty parametru „value1“ s hodnotou parametru „value2“ za použití operátoru zadaného parametrem „operator“. V opačném případě vrací hodnotu parametru „no“.

 **Příklad:**

```
COMPAREDATE(#today#, =, #now#, Ano, Ne) // Vrací Ne  
COMPAREDATE(#today#, <, #now#, Ano, Ne) // Vrací Ano  
COMPAREDATE(#today#, <=, #now#, Ano, Ne) // Vrací Ano  
COMPAREDATE(#today#, >, #now#, Ano, Ne) // Vrací Ne  
COMPAREDATE(#today#, >=, #now#, Ano, Ne) // Vrací Ne
```

string COMPAREDOUBLE(double value1, string operator, double value2, string yes, string no)

 **Použití:** HTML, JavaScript, Skript

 **Popis:** Funkce vrací hodnotu parametru „yes“, je-li splněno porovnání hodnoty parametru „value1“ s hodnotou parametru „value2“ za použití operátoru zadaného parametrem „operator“. V opačném případě vrací hodnotu parametru „no“.

 **Příklad:**

```
COMPAREDOUBLE(1, =, 2, Ano, Ne) // Vrací Ne
COMPAREDOUBLE(1, <, 2, Ano, Ne) // Vrací Ano
COMPAREDOUBLE(1, <=, 2, Ano, Ne) // Vrací Ano
COMPAREDOUBLE(1, >, 2, Ano, Ne) // Vrací Ne
COMPAREDOUBLE(1, >=, 2, Ano, Ne) // Vrací Ne
```

string CONTACT(int id)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací celé jméno a e-mailovou adresu uživatele zadaného parametrem „id“.

 **Příklad:**

```
CONTACT(1)
// Vrací „NetGenium s.r.o.“ <info@netgenium.com>“
CONTACT(#userid#)
```

string CONTACT(string loginname)

 **Použití:** HTML, JavaScript, Skript

 **Popis:** Funkce vrací celé jméno a e-mailovou adresu uživatele zadaného parametrem „loginname“.

 **Příklad:**

```
CONTACT(NetGenium)
// Vrací „NetGenium s.r.o.“ <info@netgenium.com>“
CONTACT(#loginname#)
```

string CONTACTEMAIL(string contact)

Použití: HTML, JavaScript, Skript

Popis: Funkce vrací e-mailovou adresu kontaktu zadaného parametrem „contact“ typu „e-mailová adresa“ podle specifikace RFC2822.

Příklad:

```
CONTACTEMAIL("NetGenium s.r.o." <info@netgenium.com>)  
// Vrací „info@netgenium.com“
```

string CONTACTNAME(string contact)

Použití: HTML, JavaScript, Skript

Popis: Funkce vrací jméno kontaktu zadaného parametrem „contact“ typu „e-mailová adresa“ podle specifikace RFC2822.

Příklad:

```
CONTACTNAME("NetGenium s.r.o." <info@netgenium.com>)  
// Vrací „NetGenium s.r.o.“
```

string CONTROL(string name, string html, string help)


Použití: HTML v editačním formuláři


Popis: Funkce generuje HTML kód vlastního ovládacího prvku v editačním formuláři.

Příklad:

```
CONTROL(Jméno, <INPUT type=text class=tb>, Zadejte vaše jméno)
```

void COMMIT()

 **Použití:** Skript (na místě, kde se očekává zadání komentáře)

 **Popis:** Funkce potvrdí aktuálně prováděnou transakci serverového skriptu, a provede databázové operace následujících příkazů skriptu, které jsou standardně prováděny vždy až po úspěšném provedení celého skriptu:


- Uložit proměnnou zpět do databáze
- Smazat záznam
- Nový záznam – ihned po spuštění funkce „COMMIT“ dojde k naplnění obsahu proměnné „#nid#“
- Uložit hodnotu do databáze
- Uložit více hodnot do databáze

 **Příklad:**

```
COMMIT ( )
```

int COPYATTACHMENT(int file)

 **Použití:** Skript


 **Popis:** Funkce vytvoří kopii souborové přílohy zadané parametrem „file“, a vrátí její ID.

 **Příklad:**

```
COPYATTACHMENT (#ng_soubor#)
```

int COPYATTACHMENT(int file, string directory)

 **Použití:** Skript


 **Popis:** Funkce vytvoří kopii souborové přílohy zadané parametrem „file“, a uloží ji do adresáře zadaného parametrem „directory“. Funkce vrátí hodnotu „0“.

 **Příklad:**

```
COPYATTACHMENT (#ng_soubor#, C:\Adresář)
```

double COS(double number)

 **Použití:** HTML, Skript


 **Popis:** Funkce vrací cosinus z hodnoty zadané parametrem „number“.

 **Příklad:**

```
COS(-1.23)
// Vrací „0,334237727124503“
COS(1)
// Vrací „0,540302305868139“
COS(#A#)
```

int COUNT(object[] array)

 **Použití:** Skript


 **Popis:** Funkce vrací počet hodnot skriptové proměnné zadané parametrem „array“, je-li proměnná polem hodnot. Není-li proměnná polem hodnot, vrací funkce hodnotu „1“.

 **Příklad:**

```
COUNT (#A#)
```

string CRC16(string value1 [, string value2])

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací „CRC16“ součet všech hodnot uvedených v parametrech funkce. Při stanovení součtu se hodnoty nejdříve oddělí pomlčkou.

 **Příklad:**

```
CRC16(a, b, c)
// Vrací „27651“
```

int CREATEATTACHMENT(string base64)

 **Použití:** Skript


 **Popis:** Funkce vytvoří souborovou přílohu z obrázku zadaného parametrem „base64“. Funkce přijímá zakódovaný řetězec zadaný parametrem „base64“, a vrací číslo souborové přílohy vytvořeného obrázku.

 **Příklad:**

```
CREATEATTACHMENT (#ng_kod#)
CREATEATTACHMENT (data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAFQAAAEsCAYAAAAA1u0HIAAAQAE1EQVR
4nOzd+1tSWRs38PePW7oVJVEUUQwhD5hKkhTlidIOFmlROqRFqWVZmuWkWZmWWRalmWZlmVDWc4YZY8TM...)
```

int CREATEATTACHMENT(string code, string type, int width, int height, bool label)

 **Použití:** Skript

 **Popis:** Funkce vytvoří souborovou přílohu z čárového kódu zadaného parametrem „code“. Typ čárového kódu zadaný parametrem „type“ může nabývat jedné z následujících hodnot:

- CODABAR
- CODE128
- CODE128A
- CODE128B
- CODE128C
- CODE39
- CODE93
- EAN13
- EAN8
- UPCA
- UPCE
- ITF14
- QRCODE


Šířka obrázku je definována parametrem „width“, výška parametrem „height“. Parametr „label“ udává, zda má být ve spodní části obrázku vypsán čárový kód pomocí latinky. Během generování obrázku může dojít k chybě způsobené nesouladem typu čárového kódu s jeho obsahem. Jedná se zejména o nepovolené znaky nebo délku čárového kódu. V takovém případě nedojde k přerušení, ale chyba je vypsána do samotného finálního obrázku červeným písmem na bílém pozadí. Funkce vrací číslo souborové přílohy vytvořeného obrázku.

 **Příklad:**

```
CREATEATTACHMENT(abc, CODE128A, 200, 100, false)
```

void CREATEATTACHMENTTEST(string base64)

 **Použití:** Skript


 **Popis:** Funkce ověří zdrojová data obrázku zadaného parametrem „base64“. Funkce přijímá zakódovaný řetězec zadaný parametrem „base64“, a vrací hodnotu „OK“, pokud má zakódovaný řetězec správný tvar. V opačném případě funkce vyvolá výjimku.

 **Příklad:**

```
CREATEATTACHMENTTEST (#ng_kod#)
CREATEATTACHMENTTEST (data:image/png;base64,iVBORw0KGgoAAAANSUUhEUgAAAfQAAAEsCAYAAAAA1u0HIAAAgAElEQVR4nOzd+1tSWRs38PePW7oVJVEUUQwhD5hKkhTlidIOFmlROqRFqWVZmuWkWZmWWRa1mWZlmVDWc4YZY8TM...)
```

void CREATEATTACHMENTTEST(string code, string type, int width, int height, bool label)

 **Použití:** Skript

 **Popis:** Funkce ověří vstupní parametry čárového kódu zadaného parametrem „code“. Typ čárového kódu zadaný parametrem „type“ může nabývat jedné z následujících hodnot:


- CODABAR
- CODE128
- CODE128A
- CODE128B
- CODE128C
- CODE39
- CODE93
- EAN13
- EAN8
- UPCA
- UPCE
- ITF14
- QRCODE


Šířka obrázku je definována parametrem „width“, výška parametrem „height“. Parametr „label“ udává, zda má být ve spodní části obrázku vypsaný čárový kód pomocí latinky. Funkce vrací hodnotu „OK“, pokud mají vstupní parametry validní hodnoty. V opačném případě funkce vyvolá výjimku.

 **Příklad:**

```
CREATEATTACHMENTTEST(abc, CODE128A, 200, 100, false)
```

void CREATETEXTFILE(string text, string path)

 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce vytvoří na disku textový soubor zadaný parametrem „path“, a zapíše do něj text definovaný parametrem „text“ ve výchozím kódování serveru.

 **Příklad:**

```
CREATETEXTFILE(text, C:\Adresář\soubor.txt)
```

int CREATEZIPATTACHMENT(string attachments, string filename)

 **Použití:** Skript


 **Popis:** Funkce vytvoří ZIP archiv ze souborových příloh s názvem archivu zadaným parametrem „filename“. Souborové přílohy jsou zadané parametrem „attachments“ jako seznam ID souborových příloh oddělených středníkem. Funkce vrací číslo souborové přílohy vytvořeného ZIP archivu.

 **Příklad:**

```
CREATEZIPATTACHMENT(1;2;3, abc.zip)
```

DateTime DATE(DateTime date)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací datum (bez hodin, minut atd.) z data zadaného parametrem „date“.

 **Příklad:**

```
DATE(15.2.2013 12:34)  
// Vrací „15.2.2013“  
DATE(#now#)
```


DateTime[] DateArray(DateTime date1; DateTime date2; ...)

 **Použití:** Skript


 **Popis:** Funkce deklaruje skriptovou proměnnou typu pole hodnot „DateTime“ z jednotlivých hodnot zadaných parametrem „date1“, „date2“, atd. Parametry funkce se oddělují pomocí středníku nebo tabulátoru. Volání funkce „DateArray“ musí dodržovat velká a malá písmena v názvu funkce.

 **Příklad:**

```
DateArray(#now#;#today#)
```

string DATECONDITION(string ng_date, DateTime date1, DateTime date2)

 **Použití:** HTML, Skript

 **Popis:** Funkce vrací SQL kód podmínky, která je určena na porovnávání, zda sloupec s názvem „ng_date“ (obsahem sloupce je datum) leží mezi daty zadanými „date1“ a „date2“.


 **Příklad:**


Použití v podmínce dotazu návrháře skriptu:

```
[(ID)] [rovná se] [OK#crlf#DATECONDITION(ng_od, ng_do, #ng_termin#, #ng_do#)]
[(ID)] [rovná se] [OK#crlf#DATECONDITION(ng_od, ng_do, #ng_termin:history#, #ng_do:history#)]
Před OK#crlf# nesmí být mezera
```

```
DATECONDITION(ng_date, 10.6.2013, 10.6.2013)
// Vrací „(ng_date >= '2013-06-10' AND ng_date < '2013-06-11')“
DATECONDITION(ng_date, 10.6.2013, 10.6.2013 18:30)
// Vrací „(ng_date >= '2013-06-10' AND ng_date < '2013-06-10 18:30')“
DATECONDITION(ng_date, 10.6.2013 15:00, 10.6.2013)
// Vrací „(ng_date = '2013-06-10' OR (ng_date >= '2013-06-10 15:00' AND ng_date < '2013-06-11'))“
DATECONDITION(ng_date, 10.6.2013 15:00, 10.6.2013 18:30)
// Vrací „(ng_date = '2013-06-10' OR (ng_date >= '2013-06-10 15:00' AND ng_date < '2013-06-10 18:30'))“
```

string DATECONDITION(string ng_from, string ng_to, DateTime date1, DateTime date2)

 **Použití:** HTML, Skript


 **Popis:** Funkce vrací SQL kód podmínky, která je určena na porovnávání, zda se datumový interval zadaný parametry (databázovými sloupci) „ng_from“ a „ng_to“ překrývá s intervalem zadaným parametry „date1“ a „date2“.

 **Příklad:**

```
DATECONDITION(ng_from, ng_to, 10.6.2013, 10.6.2013)
// Vrací „((ng_from >= '2013-06-10' AND ng_from < '2013-06-11' AND ng_to IS NULL) OR (ng_from < '2013-06-11' AND ng_to >= '2013-06-10'))“
DATECONDITION(ng_from, ng_to, 10.6.2013, 10.6.2013 18:30)
// Vrací „((ng_from >= '2013-06-10' AND ng_from < '2013-06-10 18:30' AND ng_to IS NULL) OR (ng_from < '2013-06-10 18:30' AND ng_to >= '2013-06-10'))“
DATECONDITION(ng_from, ng_to, 10.6.2013 15:00, 10.6.2013)
// Vrací „(((ng_from = '2013-06-10' OR (ng_from >= '2013-06-10 15:00' AND ng_from < '2013-06-11')) AND ng_to IS NULL) OR (ng_from < '2013-06-11' AND (ng_to = '2013-06-10' OR ng_to >= '2013-06-10 15:00'))))“
DATECONDITION(ng_from, ng_to, 10.6.2013 15:00, 10.6.2013 18:30)
// Vrací „(((ng_from = '2013-06-10' OR (ng_from >= '2013-06-10 15:00' AND ng_from < '2013-06-10 18:30')) AND ng_to IS NULL) OR (ng_from < '2013-06-10 18:30' AND (ng_to = '2013-06-10' OR ng_to >= '2013-06-10 15:00'))))“
```

string DATECONDITION2(string ng_from, string ng_to, DateTime date1, DateTime date2)

 **Použití:** HTML, Skript


 **Popis:** Funkce vrací SQL kód podmínky, která je určena na porovnávání, zda se datumový interval zadaný parametry (databázovými sloupci) „ng_from“ a „ng_to“ překrývá s intervalem zadaným parametry „date1“ a „date2“ včetně intervalu před datem „ng_from“. Funkce nalezne využití při porovnávání například platnosti cen – cena je platná od a do určitého data s tím, že platnost může být také vymezena pouze datem od („ng_to“ je NULL), tudíž s neohraničenou dobou platnosti (cena platí od určitého data až do současnosti).


 **Příklad:**

```
DATECONDITION2(ng_from, ng_to, 10.6.2013, 10.6.2013)
// Vrací „((ng_from < '2013-06-12' AND ng_to IS NULL) OR (ng_from < '2013-06-12' AND ng_to >= '2013-06-10'))“
DATECONDITION2(ng_from, ng_to, 10.6.2013 15:00, 10.6.2013)
// Vrací „((ng_from < '2013-06-12' AND ng_to IS NULL) OR (ng_from < '2013-06-12' AND (ng_to = '2013-06-11' OR ng_to >= '2013-06-10 15:00'))))“
DATECONDITION2(ng_from, ng_to, 10.6.2013, 10.6.2013 18:30)
// Vrací „((ng_from < '2013-06-11 18:30' AND ng_to IS NULL) OR (ng_from < '2013-06-11 18:30' AND ng_to >= '2013-06-10'))“
DATECONDITION2(ng_from, ng_to, 10.6.2013 15:00, 10.6.2013 18:30)
```

```
// Vrací „((ng_from < '2013-06-11 18:30' AND ng_to IS NULL) OR (ng_from < '2013-06-11 18:30'  
AND (ng_to = '2013-06-11' OR ng_to >= '2013-06-10 15:00'))“
```

void DELETEDATA(string dbname, string condition)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce nenávratně smaže data z databázové tabulky zadané parametrem „dbname“ za podmínky zadané parametrem „condition“. Tato funkce nepoužívá zápis do historie, a je optimalizována pro hromadné mazání dat, které by jinak selhalo na uplynutí timeoutu, pokud by bylo spuštěno napřímo v databázi například pomocí programu „Microsoft SQL Server Management Studio“.

 **Příklad:**

```
DELETEDATA(ng_form, id < 123)
```

void DELETEHISTORY()

 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce smaže z databáze historii všech editačních formulářů

- starší než 6 měsíců,
- starší než X měsíců, pokud je X zadáno v nastavení editačního formuláře.

 **Příklad:**

```
DELETEHISTORY()
```

void DELETEHISTORY(string dbname)

 **Použití:** Skript (na místě, kde se očekává zadání komentáře)

 **Popis:** Funkce smaže z databáze historii editačního formuláře zadaného parametrem „dbname“


- starší než 6 měsíců,
- starší než X měsíců, pokud je X zadáno v nastavení editačního formuláře.

 **Příklad:**

```
DELETEHISTORY(susers)
```

void DISABLEAUTOFILL()

 **Použití:** Skript


 **Popis:** Funkce zakáže webovému prohlížeči funkci automatického vyplňování formulářů u ovládacích prvků typu TextBox.

 **Příklad:**

```
DISABLEAUTOFILL ()
```

void DISABLEFORM()

 **Použití:** Skript


 **Popis:** Funkce disabluje všechny ovládací prvky umístěné v editačním formuláři včetně tlačítek „Uložit“ a „Smazat“. Použití funkce „DISABLEFORM“ má z pohledu uživatelů totožný efekt jako použití javascriptové funkce „controls_Disable“, je však bezpečnější u ovládacích prvků typu „File“ a „Image“. Tyto ovládací prvky standardně obsahují tlačítka pro upload souborové přílohy z klientské stanice na server – serverová funkce „DISABLEFORM“ tyto tlačítka bezpečně odebere, kdežto javascriptová funkce „controls_Disable“ je pouze skryje.

 **Příklad:**

```
DISABLEFORM ()
```

double DISTANCE(double lat1, double lon1, double lat2, double lon2)




 **Použití:** Skript

 **Popis:** Funkce vrací vzdálenost v kilometrech mezi dvěma zeměpisnými místy zadanými parametry „lat1“, „lon1“, „lat2“ a „lon2“. Jednotlivé parametry představují zeměpisné souřadnice daných míst.

 **Příklad:**




```
DISTANCE(14.108998537, 50.0723726721, 14.4476009458, 50.0794492033)  
// Vrací „2.859“
```

void DIV(string id)

-  **Použití:** Název ovládacího prvku HTML – tělo ovládacího prvku musí zůstat prázdné
-  **Popis:** Funkce vygeneruje HTML tag „<div id="{id}">“, resp. zahájení nového divu. ID tagu definuje parametr „id“.
-  **Příklad:**




```
DIV(D1)
```

void DIV(string id, bool visible)

-  **Použití:** Název ovládacího prvku HTML – tělo ovládacího prvku musí zůstat prázdné
-  **Popis:** Funkce vygeneruje HTML tag „<div id="{id}" style="display:{visible}">“, resp. zahájení nového divu. ID tagu definuje parametr „id“, jeho viditelnost parametr „visible“.
-  **Příklad:**

```
DIV(D1, false)
```


void /DIV

-  **Použití:** Název ovládacího prvku HTML – tělo ovládacího prvku musí zůstat prázdné
-  **Popis:** Funkce vygeneruje HTML tag „</div>“, resp. zakončení předchozího divu, který byl vygenerován pomocí funkce „DIV“.
-  **Příklad:**

```
/DIV
```

double[] DoubleArray(double number1; double number2; ...)

 **Použití:** Skript


 **Popis:** Funkce deklaruje skriptovou proměnnou typu pole hodnot „Double“ z jednotlivých hodnot zadaných parametrem „number1“, „number2“, atd. Parametry funkce se oddělují pomocí středníku nebo tabulátoru. Volání funkce „DoubleArray“ musí dodržovat velká a malá písmena v názvu funkce.

 **Příklad:**

```
DoubleArray(1.1;2.2;3.3)
```

string EMAIL(int id)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací e-mailovou adresu uživatele zadaného parametrem „id“.

 **Příklad:**

```
EMAIL(1)  
EMAIL(#userid#)
```

string EMAIL(string loginname)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací e-mailovou adresu uživatele zadaného parametrem „loginname“.

 **Příklad:**

```
EMAIL(Administrator)  
EMAIL(#loginname#)
```

string EMAILS(int id1 [, int id2])

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací seznam e-mailových adres uživatelů oddělených středníkem, jejichž ID je zadáno v parametrech funkce „id1“, „id2“ atd.

 **Příklad:**

```
EMAILS (1, 2, 3)
```

string EMAILS(string loginname1 [, string loginname2])

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací seznam e-mailových adres uživatelů oddělených středníkem, jejichž přihlašovací jméno je zadáno v parametrech funkce „loginname1“, „loginname2“ atd.

 **Příklad:**

```
EMAILS (loginname1, loginname2, loginname3)
```

string[] EmailArray(string email1; string email2; ...)

 **Použití:** Skript


 **Popis:** Funkce deklaruje skriptovou proměnnou typu pole hodnot „String“ z jednotlivých hodnot zadaných parametrem „email1“, „email2“, atd. Parametry funkce se oddělují pomocí středníku nebo tabulátoru. Volání funkce „EmailArray“ musí dodržovat velká a malá písmena v názvu funkce. Funkce „EmailArray“ se využívá u seznamu e-mailů, protože kontakt uvedený v e-mailové adrese může obsahovat středník.

 **Příklad:**

```
EmailArray("NetGenium; s.r.o." <info@netgenium.com>;"NetGenium; s.r.o." <info@netgenium.com>)
```

string EMAILLINK(string dbname, int id)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací HTML odkaz na databázový záznam zadaný parametrem „dbname“ a „id“. Parametr „dbname“ určuje identifikátor editačního formuláře, parametr „id“ je primární klíč databázového záznamu. Funkce se používá v textu e-mailu, do kterého je potřeba umístit odkaz na databázový záznam.

 **Příklad:**

```
EMAILLINK(susers, 1)
```

string EQUALS(string compare1, string compare2, string yes, string no)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací hodnotu parametru „yes“, je-li hodnota parametru „compare1“ rovna hodnotě parametru „compare2“. V opačném případě vrací hodnotu parametru „no“.

 **Příklad:**

```
EQUALS(#loginname#, Administrator, 0, -1)
```

void EVALATTACHMENTSTATUS(int file, string status, string message)

 **Použití:** Skript


 **Popis:** Funkce zjišťuje, zda souborová příloha zadaná parametrem „file“ nabývá stavu definovaného parametrem „status“. Pokud se příloha nachází v některém z definovaných stavů, funkce vyvolá přerušení skriptu se zprávou definovanou v parametru „message“. Typické využití je při kontrole přílohy antivirovým systémem.

 **Příklad:**

```
EVALATTACHMENTSTATUS(ng_priloha, 0, Právě probíhá antivirová kontrola přiložených dokumentů.  
Odešlete prosím formulář znovu za několik málo vteřin)  
EVALATTACHMENTSTATUS(ng_priloha, 1, V přiloženém dokumentu XYZ byla nalezena virová infekce.  
Tuto přílohu odstraňte nebo použijte jinou)
```


void EVALCAPTCHA(string captcha, string message)

 **Použití:** Skript


 **Popis:** Funkce testuje, zdali text zadaný parametrem „captcha“ odpovídá zobrazenému ověřovacímu kódu captcha, který byl v editačním formuláři zobrazen pomocí funkce „CAPTCHA“. Pokud se texty neshodují, funkce vyvolá přerušení skriptu s chybovou zprávou zadanou parametrem „message“. Funkce je svázána s funkcí „CAPTCHA“, bez níž by funkce „EVALCAPTCHA“ neměla smysl.

 **Příklad:**

```
EVALCAPTCHA(#ng_overovacikod#, Opište prosím ověřovací kód z obrázku.)
```

void EVALGOOGLECAPTCHA(string secretKey, string message)

 **Použití:** Skript


 **Popis:** Funkce testuje, zda je zaškrtnutý ovládací prvek, který byl zobrazen v editačním formuláři pomocí funkce „GOOGLECAPTCHA“ nebo „GOOGLECAPTCHAinvisible“. Jako první parametr se uvádí unikátní klíč „secretKey“, který lze získat společně s klíčem „siteKey“ při registraci prvku Google reCAPTCHA u společnosti Google. Pokud ovládací prvek není zaškrtnut, funkce vyvolá přerušení skriptu s chybovou zprávou zadanou parametrem „message“. Funkce je svázána s funkcí „GOOGLECAPTCHA“, bez níž by funkce „EVALGOOGLECAPTCHA“ neměla smysl.

 **Příklad:**

```
EVALGOOGLECAPTCHA(d5f545g5ccd5f2c5df1, Zaškrtněte prosím pole CAPTCHA.)
```

double EXP(double number)

 **Použití:** HTML, Skript


 **Popis:** Funkce vrací hodnotu Eulerova čísla umocněnou na hodnotu zadanou parametrem „number“.

 **Příklad:**

```
EXP(1.23)
// Vrací „3,42122953628967“
EXP(#A#)
```

string EXPORTIMAGES(string html)

 **Použití:** Skript


 **Popis:** Funkce vyexportuje inline obrázky z HTML kódu zadaného parametrem „html“, a nahradí všechny URL odkazy obrázků („Download.aspx?abc“) z HTML kódu za řetězec dat obrázků zakódovaných pomocí „base64“.

 **Příklad:**

```
A = String
A = EXPORTIMAGES(#ng_html#)
```

string EXPORTIMAGES(string html, int maxWidth, int maxHeight)

 **Použití:** Skript


 **Popis:** Funkce vyexportuje inline obrázky z HTML kódu zadaného parametrem „html“, a nahradí všechny URL odkazy obrázků („Download.aspx?abc“) z HTML kódu za řetězec dat obrázků zakódovaných pomocí „base64“. U všech zakódovaných obrázků budou proporcionálně nastaveny parametry „width“ a „height“ v závislosti na parametrech „maxWidth“ a „maxHeight“.

 **Příklad:**

```
A = String
A = EXPORTIMAGES(#ng_html#, 20, 15)
```

void EXPORTIMAGES(string html, string dir, string url)

 **Použití:** Skript


 **Popis:** Funkce vyexportuje inline obrázky z HTML kódu zadaného parametrem „html“ do adresáře na disku zadaného parametrem „dir“. Obrázky budou pojmenovány podle pořadí, jak jsou postupně použity v HTML kódu. První obrázek se bude jmenovat např. „img01.jpg“, druhý „img02.jpg“ atd. Parametr „url“ představuje veřejně dostupnou URL, ze které bude možné tyto obrázky stáhnout.

 **Příklad:**

```
A = String(#guid#)
EXPORTIMAGES(#ng_html#, #rootpath#Temp\#A#, https://www.netgenium.com/temp/#A#)
```

object FIRST(object[] array)

 **Použití:** Skript

 **Popis:** Funkce vrací první hodnotu skriptové proměnné zadané parametrem „array“, je-li proměnná polem hodnot. Není-li proměnná polem hodnot, pak funkce vrací hodnotu proměnné. Neobsahuje-li pole hodnot žádný prvek, vrací funkce prázdný řetězec.

 **Příklad:**

```
FIRST (#A#)
```

DateTime FIRSDAYINMONTH(DateTime date)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací datum prvního dne v měsíci z data zadaného parametrem „date“.

 **Příklad:**

```
FIRSTDAYINMONTH (15.2.2013)  
// Vrací „01.02.2013“  
FIRSTDAYINMONTH (#today#)
```

DateTime FIRSDAYINQUARTAL(DateTime date)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací datum prvního dne ve čtvrtletí (kvartálu) z data zadaného parametrem „date“.

 **Příklad:**

```
FIRSTDAYINQUARTAL (15.2.2013)  
// Vrací „01.01.2013“  
FIRSTDAYINQUARTAL (#today#)
```

DateTime FIRSDAYINWEEK(DateTime date)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací datum prvního dne v týdnu z data zadaného parametrem „date“.

 **Příklad:**

```
FIRSDAYINWEEK(15.2.2013)
// Vrací „11.02.2013“
FIRSDAYINWEEK(#today#)
```

DateTime FIRSDAYINYEAR(DateTime date)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací datum prvního dne v roce z data zadaného parametrem „date“.

 **Příklad:**

```
FIRSDAYINYEAR(15.2.2013)
// Vrací „01.01.2013“
FIRSDAYINYEAR(#today#)
```

double FLOOR(double number)

 **Použití:** HTML, Skript


 **Popis:** Funkce vrací nejvyšší celé číslo, které je menší nebo rovno hodnotě zadané parametrem „number“.

 **Příklad:**

```
FLOOR(-1.23)
// Vrací „-2“
FLOOR(1.23)
// Vrací „1“
```

string FORMATCURRENCY(double number)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce formátuje číslo zadané parametrem „number“ do textové podoby – s mezerou jako oddělovačem tisíců, s desetinnou čárkou nebo tečkou v závislosti na jazykovém nastavení aktuálně přihlášeného uživatele, a zaokrouhlené na dvě desetinná místa.

 **Příklad:**

```
FORMATCURRENCY(1100)
// Vrací „1 100,00“
FORMATCURRENCY(1100.5)
// Vrací „1 100,50“
FORMATCURRENCY(1100.5412)
// Vrací „1 100,54“
```

string FORMATCURRENCY(double number, string symbol)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce formátuje číslo zadané parametrem „number“ do textové podoby – s mezerou jako oddělovačem tisíců, s desetinnou čárkou nebo tečkou v závislosti na jazykovém nastavení aktuálně přihlášeného uživatele, zaokrouhlené na dvě desetinná místa, a se symbolem zadaným parametrem „symbol“.

 **Příklad:**

```
FORMATCURRENCY(1100, Kč)
// Vrací „1 100,00 Kč“
FORMATCURRENCY(1100.5, Kč)
// Vrací „1 100,50 Kč“
FORMATCURRENCY(1100.5412, Kč)
// Vrací „1 100,54 Kč“
```

string FORMATDATATABLE(string query)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací výsledek dotazu zadaného parametrem „query“ formátovaný do tabulky.

 **Příklad:**

```
FORMATDATATABLE(SELECT * FROM susers ORDER BY id)
```

string FORMATDATATABLE(string query, string title)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací výsledek dotazu zadaného parametrem „query“ formátovaný do tabulky, a s titulkem zadaným parametrem „title“.

 **Příklad:**

```
FORMATDATATABLE(SELECT * FROM susers ORDER BY id, Uživatelé)
```

string FORMATDATAVIEW(string query)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací výsledek dotazu zadaného parametrem „query“ formátovaný do tabulky.

 **Příklad:**

```
FORMATDATAVIEW(SELECT * FROM susers ORDER BY id)
```

string FORMATDATAVIEW(string query, string title)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací výsledek dotazu zadaného parametrem „query“ formátovaný do tabulky, a s titulkem zadaným parametrem „title“.

 **Příklad:**

```
FORMATDATAVIEW(SELECT * FROM susers ORDER BY id, Uživatelé)
```

string FORMATDATE(DateTime date)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce formátuje datum zadané parametrem „date“ do textové podoby na základě výchozí šablony „D“ a jazykového nastavení aktuálně přihlášeného uživatele.

 **Příklad:**

```
FORMATDATE(#today#)
// Vrací například „úterý 2. dubna 2019“
```

string FORMATDATE(DateTime date, string template)

 **Použití:** HTML, JavaScript, Skript

 **Popis:** Funkce formátuje datum zadané parametrem „date“ do textové podoby na základě šablony definované parametrem „template“ a jazykového nastavení aktuálně přihlášeného uživatele. Mezi šablony patří:


- „d“: 29.2.2000
- „D“: 29. února 2000
- „f“: 29. února 2000 12:34
- „F“: 29. února 2000 12:34:00
- „g“: 29.2.2000 12:34
- „G“: 29.2.2000 12:34:00
- „r“: Tue, 29 Feb 2000 12:34:00 GMT
- „s“: 2000-02-29T12:34:00
- „t“: 12:34
- „T“: 12:34:00
- „y“: únor 2000
- „dddd, MMMM dd yyyy“: úterý, února 29 2000
- „dd-MM-yy“: 29-02-00
- „w“: 9
- „ww“: 09
- „www“: 2000 / 09

 **Příklad:**

```
FORMATDATE(#today#, dd.MM.yyyy)
FORMATDATE(#today#, s)
```

string FORMATDATES(string value, bool dayPrefix, bool weekNumber)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce nahradí všechny výskyty datumů v textovém řetězci zadaném parametrem „value“ za formátovanou hodnotu do textové podoby na základě výchozí šablony „dd.MM.yyyy“ a jazykového nastavení aktuálně přihlášeného uživatele. Parametr „dayPrefix“ indikuje, zda se před každé datum vloží zkrácený název dne. Parametr „weekNumber“ indikuje, zda se za každé datum vloží číslo týdne.

 **Příklad:**

```
FORMATDATES(abc #today# def, true, true)
// Vrací „abc pá 14. 11. 2014 (46. týden) def“
FORMATDATES(abc #today# def, true, false)
// Vrací „abc pá 14. 11. 2014 def“
FORMATDATES(abc #today# def, false, true)
// Vrací „abc 14. 11. 2014 (46. týden) def“
```

string FORMATDATESQL(DateTime date)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce formátuje datum zadané parametrem „date“ do textové podoby určené pro použití v podmínce databázového dotazu například ve spojení s funkcí „SQL“, a na základě zvyklostí aktuální databáze.

 **Příklad:**

```
FORMATDATESQL(#today#)
SQL(SELECT name FROM sholiday WHERE date_ = FORMATDATESQL(1.1.2019))
```

string FORMATDIFFERENCES(string compare1, string compare2)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací tabulku se dvěma sloupci, kde v prvním sloupci jsou na jednotlivých řádcích zobrazeny hodnoty oddělené středníkem z parametru „compare1“, a ve druhém sloupci jsou stejným způsobem zobrazené hodnoty z parametru „compare2“. Každý řádek této tabulky je mezi sebou porovnán, a pokud hodnoty na jednotlivých řádcích nejsou stejné, jsou označené červeně. První sloupec má název „A“, druhý „B“.

 **Příklad:**

```
FORMATDIFFERENCES(Text1;Druhýtext2, Text1;Text2)
```


string FORMATDIFFERENCES(string compare1, string compare2, string header1, string header2)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací tabulku se dvěma sloupci, kde v prvním sloupci jsou na jednotlivých řádcích zobrazeny hodnoty oddělené středníkem z parametru „compare1“, a ve druhém sloupci jsou stejným způsobem zobrazené hodnoty z parametru „compare2“. Každý řádek této tabulky je mezi sebou porovnán, a pokud hodnoty na jednotlivých řádcích nejsou stejné, jsou označeny červeně. První sloupec má název definovaný parametrem „header1“, druhý parametrem „header2“.

 **Příklad:**

```
FORMATDIFFERENCES (Text1;Druhytext2, Text1;Text2, První, Druhý)
```

string FORMATDIFFERENCES(string compare1, string compare2, string header1, string header2, string title)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací tabulku se dvěma sloupci, kde v prvním sloupci jsou na jednotlivých řádcích zobrazeny hodnoty oddělené středníkem z parametru „compare1“, a ve druhém sloupci jsou stejným způsobem zobrazené hodnoty z parametru „compare2“. Každý řádek této tabulky je mezi sebou porovnán, a pokud hodnoty na jednotlivých řádcích nejsou stejné, jsou označeny červeně. První sloupec má název definovaný parametrem „header1“, druhý parametrem „header2“. Celá tabulka má nadpis definovaný parametrem „title“.

 **Příklad:**

```
FORMATDIFFERENCES (Text1;Druhytext2, Text1;Text2, První, Druhý, Rozdíly)
```

string FORMATDOUBLE(double number)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce formátuje číslo zadané parametrem „number“ do textové podoby – s mezerou jako oddělovačem tisíců, a s desetinnou čárkou nebo tečkou v závislosti na jazykovém nastavení aktuálně přihlášeného uživatele.

 **Příklad:**

```
FORMATDOUBLE(1100)
// Vrací „1 100“
FORMATDOUBLE(1100.5)
// Vrací „1 100,5“
FORMATDOUBLE(1100.5412)
// Vrací „1 100,5412“
```

string FORMATDOUBLE(double number, string symbol)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce formátuje číslo zadané parametrem „number“ do textové podoby – s mezerou jako oddělovačem tisíců, s desetinnou čárkou nebo tečkou v závislosti na jazykovém nastavení aktuálně přihlášeného uživatele, a se symbolem zadaným parametrem „symbol“.

 **Příklad:**

```
FORMATDOUBLE(1100, kg)
// Vrací „1 100 kg“
FORMATDOUBLE(1100.5, kg)
// Vrací „1 100,5 kg“
FORMATDOUBLE(1100.5412, kg)
// Vrací „1 100,5412 kg“
```

string FORMATDOUBLESQL(double number)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce formátuje číslo zadané parametrem „number“ do textové podoby určené pro použití v podmínce databázového dotazu například ve spojení s funkcí „SQL“, a na základě zvyklostí aktuální databáze.

 **Příklad:**

```
SQL(SELECT COUNT(*) FROM sstatistics2 WHERE session > FORMATDOUBLESQL(1000.1234))
```

string FORMATINTARRAYSQL(string array)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce formátuje seznam nenulových čísel oddělených středníkem zadaných parametrem „number“ do textové podoby určené pro použití v podmínce databázového dotazu například ve spojení s funkcí „SQL“, a na základě zvyklostí aktuální databáze. Pokud seznam čísel neobsahuje ani jednu nenulovou číselnou hodnotu, je vrácena hodnota „0“.

 **Příklad:**

```
SQL(SELECT loginname FROM susers WHERE id IN (FORMATINTARRAYSQL(1;2)))
```

string FORMATINTSQL(int number)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce formátuje číslo zadané parametrem „number“ do textové podoby určené pro použití v podmínce databázového dotazu například ve spojení s funkcí „SQL“, a na základě zvyklostí aktuální databáze.

 **Příklad:**

```
SQL(SELECT loginname FROM susers WHERE id = FORMATINTSQL(1))
```

string FORMATLONGARRAYSQL (string array)

 **Použití:** HTML, JavaScript, Skript

 **Popis:** Funkce formátuje seznam nenulových velkých čísel oddělených středníkem zadaných parametrem „number“ do textové podoby určené pro použití v podmínce databázového dotazu například ve spojení s funkcí „SQL“, a na základě zvyklostí aktuální databáze. Pokud seznam čísel neobsahuje ani jednu nenulovou číselnou hodnotu, je vrácena hodnota „0“.

 **Příklad:**

```
SQL(SELECT loginname FROM susers WHERE id IN (FORMATLONGARRAYSQL(1;2)))
```

string FORMATLONGSQL (long number)

Použití: HTML, JavaScript, Skript

Popis: Funkce formátuje číslo zadané parametrem „number“ do textové podoby určené pro použití v podmínce databázového dotazu například ve spojení s funkcí „SQL“, a na základě zvyklostí aktuální databáze.

Příklad:

```
SQL(SELECT loginname FROM susers WHERE id = FORMATLONGSQL(1))
```

string FORMATSTRING(string template, string variable0, ...[string variableXY])

Použití: HTML, JavaScript, Skript

Popis: Funkce formátuje text zadaný parametrem „template“ podle zvyklostí složeného formátování v jazyce C#. Proměnné, jejichž výskyt bude nahrazen za hodnoty proměnných, se deklarují ve složených závorkách, a jsou číslovány od nuly.

Příklad:

```
FORMATSTRING({0} + {1} = {2}, 1, 2, 3)  
// Vrací „1 + 2 = 3“
```

string FORMATSTRINGSQL(string text)

Použití: HTML, JavaScript, Skript


Popis: Funkce formátuje textový řetězec zadaný parametrem „text“ do textové podoby určené pro použití v podmínce databázového dotazu například ve spojení s funkcí „SQL“, a na základě zvyklostí aktuální databáze – začínající znakem „N“ v případě databáze MSSQL, umístěný v apostrofech, a s escapovanými apostrofy, které se vyskytují uvnitř textu za zdvojené apostrofy.

Příklad:

```
SQL(SELECT id FROM susers WHERE loginname = FORMATSTRINGSQL(Administrator))
```

string **FORMATTABLE**(string column1, string values1, [string column2, string values2...])

Použití:

 **Popis:** Funkce vrací HTML kód tabulky formátované v univerzálním vzhledu určeném například pro emailové zprávy. Parametr „column1“ definuje název prvního sloupce tabulky, parametr „values1“ definuje hodnoty oddělené středníkem určené pro tento sloupec tabulky. Počet sloupců tabulky není omezen, pouze je potřeba další parametry funkce uvádět v párech – název sloupce spolu s hodnotami v tomto sloupci. Jako oddělovač hodnot je použitý středník, proto jednotlivé hodnoty ve sloupci nemohou středník obsahovat jako podřetězec.


Příklad:

```
FORMATTABLE (A, 1;2;3, B, 4;5;6)
```

A	B
1	4
2	5
3	6

string **FORMATTIME**(int minutes)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce formátuje počet minut zadaný parametrem „minutes“ do textové podoby ve formátu „HH:mm“.

Příklad:

```
FORMATTIME (1156)  
// Vrací „19:16“  
FORMATTIME (-150)  
// Vrací „-02:30“
```

string FORMATTIME2(double seconds)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce formátuje počet vteřin zadaný parametrem „seconds“ do textové podoby ve formátu „HH:mm:ss“.

 **Příklad:**

```
FORMATTIME2(1156)
// Vrací „00:16:40“
FORMATTIME2(-150)
// Vrací „-00:02:30“
```

string FORMNAME(int form, string separator)


 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací celý název editačního formuláře zadaného parametrem „form“ včetně názvu aplikační skupiny a aplikace. Jako oddělovač mezi těmito třemi názvy se použije textový řetězec zadaný parametrem „separator“.

 **Příklad:**

```
FORMNAME(#form#, " | ")
```

string GANTT(DateTime from, DateTime to, int R, int G, int B)

 **Použití:** nahlížeč tabulka


 **Popis:** Funkce zobrazí Ganttův diagram na řádce nahlížeč tabulky. Počáteční datum diagramu určuje parametr „from“, koncové datum určuje parametr „to“. Zbarvení diagramu určují hodnoty barev parametrů „R“, „G“ a „B“. Pro zobrazení diagramu je nutné, aby volání funkce „GANTT“ bylo uloženo v hodnotě ovládacího prvku „TextBox“, nastaveného „Jen ke čtení“ a jako „Skryté pole“.

 **Příklad:**

```
GANTT(01.01.2013, 01.02.2013, 25, 25, 112)
```

string GETCOOKIE(string key)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací hodnotu cookie definovanou parametrem „key“.

 **Příklad:**

```
GETCOOKIE (abc)
```

string GOOGLECAPTCHA(string siteKey)

 **Použití:** HTML v editačním formuláři


 **Popis:** Funkce vloží do HTML kódu editačního formuláře ověřovací kód Google reCAPTCHA, který slouží pro odlišení skutečných uživatelů od robotů. Vložený ověřovací kód v sobě obsahuje unikátní klíč domény zadaný parametrem „siteKey“, který lze získat při registraci Google reCAPTCHA u společnosti Google. Funkce je svázána s funkcí „EVALGOOGLECAPTCHA“, bez níž by funkce „GOOGLECAPTCHA“ neměla smysl.

 **Příklad:**

```
GOOGLECAPTCHA (5ghjhddfg54cddry5tvvggfffg)
```

string GOOGLECAPTCHAinvisible(string siteKey, string button)

 **Použití:** HTML v editačním formuláři


 **Popis:** Funkce vloží do HTML kódu editačního formuláře ověřovací kód Google reCAPTCHA, který slouží pro odlišení skutečných uživatelů od robotů. Vložený ověřovací kód v sobě obsahuje unikátní klíč domény zadaný parametrem „siteKey“, který lze získat při registraci Google reCAPTCHA u společnosti Google. Parametr „button“ je identifikátor skrytého tlačítka, které obsahuje funkci „EVALGOOGLECAPTCHA“. Tlačítko je spuštěno na straně serveru při zpracování odpovědi od Google. Funkce je svázána s funkcí „EVALGOOGLECAPTCHA“, bez níž by funkce „GOOGLECAPTCHAinvisible“ neměla smysl.

 **Příklad:**

```
GOOGLECAPTCHAinvisible (5ghjhddfg54cddry5tvvggfffg, BT123)
```

object[] GROUP(object[] array)


 **Použití:** Skript


 **Popis:** Funkce deklaruje novou skriptovou proměnnou typu pole hodnot ze skriptové proměnné typu pole hodnot, ve které jsou odstraněny duplicity. Není-li proměnná polem hodnot, pak funkce vrátí hodnotu proměnné.

 **Příklad:**

```
GROUP (#A#)
```

void HIDEFORM(string message)

 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce skryje obsah editačního formuláře nebo nahlížecí stránky, a vypíše zprávu zadanou parametrem „message“.

 **Příklad:**

```
HIDEFORM("Děkujeme, že jste nám odeslali Vaši odpověď.")
```

string HOURCONDITION(string ng_date, int hour1, int hour2)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrátí SQL kód podmínky, která je určena na porovnávání, zda sloupec s názvem „ng_date“ (obsahem sloupce je datum) leží v časovém intervalu mezi hodinami „hour1“ a „hour2“.

 **Příklad:**

```
Použití v podmínce dotazu:  
[(ID)] [rovná se] [OK#crlf#HOURCONDITION(ng_date, 8, 20)]  
Před OK#crlf# nesmí být mezera
```


string HTMLDECODE(string text)

 **Použití:** Skript


 **Popis:** Funkce nahradí všechny sekvence znaků (character entities) obsažené v parametru „text“ za odpovídající HTML znaky.

 **Příklad:**

```
HTMLDECODE(&lt;h1&gt;Administrator &lt;h1&gt;)
// Vrací „<h1>Administrator<h1>“
HTMLDECODE(&lt;p&gt; Ahoj jmenuji se &lt;br&gt; NET Genium&lt;p&gt;)
// Vrací „<p> Ahoj jmenuji se <br> NET Genium<p>“
```

string HTMLENCODE(string text)

 **Použití:** Skript


 **Popis:** Funkce nahradí všechny HTML znaky obsažené v parametru „text“ za odpovídající sekvenci znaků (character entities).

 **Příklad:**

```
HTMLENCODE(<h1>Administrator<h1>)
// Vrací „&lt;h1&gt;Administrator &lt;h1&gt;“
HTMLENCODE(<p> Ahoj jmenuji se <br> NET Genium<p>)
// Vrací „&lt;p&gt; Ahoj jmenuji se &lt;br&gt; NET Genium&lt;p&gt;“
```

string HTML2TEXT(string html, bool simplehtml)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce převede textový řetězec ve formátu HTML do podoby prostého textu, který používá „\r\n“ jako oddělovač řádek. Volitelný parametr „simplehtml“ udává, zda se má zachovat základní formátování textu pomocí HTML značek „b“ a „i“.

 **Příklad:**

```
HTML2TEXT(<div><b>abc</b></div>, true)
// Vrací „<b>abc</b>“
```

string HTML2TEXT(<div>abc</div>)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce převede textový řetězec ve formátu HTML do podoby prostého textu, který používá „\r\n“ jako oddělovač řádek.

 **Příklad:**

```
HTML2TEXT(<div>abc</div>)  
// Vrací „abc“
```

string CHARS(string text)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací jednotlivé znaky oddělené středníkem z textového řetězce zadaného parametrem „text“. Funkce dále escape znaky „\r“ a „\n“ v parametru převede na „\|r“ a „\|n“.

 **Příklad:**

```
CHARS(Administrator)  
// Vrací „A;d;m;i;n;i;s;t;r;a;t;o;r“  
CHARS(Administrator#crlf#Anonymous)  
// Vrací „A;d;m;i;n;i;s;t;r;a;t;o;r;\r;\n;A;n;o;n;y;m;o;u;s“
```

string IFF(string expression, string yes, string no)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vyhodnotí, zda platí matematický výraz zadaný parametrem „expression“. Pokud výraz platí, je vrácena hodnota zadaná parametrem „yes“, v opačném případě je vrácena hodnota zadaná parametrem „no“. Funkce očekává porovnávání čísel, a umožňuje používání operátorů „<“, „<=“, „>=“ a „>“ a „=“.

 **Příklad:**

```
IFF(-1.5 <= 2, Platí, Neplatí)  
// Vrací „Platí“
```

int INDEXOF(string text, string subtext)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací pozici podřetězce zadaného parametrem „subtext“ v řetězci zadaném parametrem „text“ počínaje indexem „0“. Není-li takový výskyt nalezen, vrací hodnotu „-1“.

 **Příklad:**

```
INDEXOF(Dobrý den, den)
// Vrací „6“
```

string INT2TIME(int minutes)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací textový řetězec ve formátu „HH:mm“ vyjádřený z počtu minut zadaného parametrem „minutes“.

 **Příklad:**

```
INT2TIME(480)
// Vrací „08:00“
INT2TIME(0)
// Vrací „00:00“
```

int[] IntegerArray(int count)

 **Použití:** Skript


 **Popis:** Funkce deklaruje skriptovou proměnnou typu pole hodnot „Integer“, a vytvoří pole o velikosti tolika prvků, kolik je zadáno parametrem „count“. Tyto prvky naplní hodnotami postupně od jedničky, každý prvek má v sobě uloženou hodnotu o jedna vyšší než předcházející prvek. Volání funkce „IntegerArray“ musí dodržovat velká a malá písmena v názvu funkce.

 **Příklad:**

```
IntegerArray(5)
```

string JTSK2WGS(double x, double y)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací konverzi hodnot souřadného systému JTSK zadaných parametry „x“ a „y“ do systému WGS. Výsledné hodnoty jsou odděleny středníkem.

 **Příklad:**

```
JTSK2WGS (x, y)
```

string LANGUAGE(string loginname)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací jazykové nastavení uživatele zadaného parametrem „loginname“. Funkce nabývá hodnoty „cs“, „de“, „en“, „fr“ nebo „sk“.

 **Příklad:**

```
LANGUAGE (Administrator)  
LANGUAGE (#loginname#)
```

object LAST(object[] array)

 **Použití:** Skript


 **Popis:** Funkce vrací poslední hodnotu skriptové proměnné zadané parametrem „array“, je-li proměnná polem hodnot. Není-li proměnná polem hodnot, pak funkce vrací hodnotu proměnné. Neobsahuje-li pole hodnot žádný prvek, funkce vrací prázdný řetězec.

 **Příklad:**

```
LAST (#A#)
```

DateTime LASTDAYINMONTH(DateTime date)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací datum posledního dne v měsíci z data zadaného parametrem „date“.

 **Příklad:**

```
LASTDAYINMONTH(15.2.2013)
// Vrací „29.02.2013“
LASTDAYINMONTH(#today#)
```

DateTime LASTDAYINQUARTAL(DateTime date)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací datum posledního dne ve čtvrtletí (kvartálu) z data zadaného parametrem „date“.

 **Příklad:**

```
LASTDAYINQUARTAL(15.2.2013)
// Vrací „31.03.2013“
LASTDAYINQUARTAL(#today#)
```

DateTime LASTDAYINWEEK(DateTime date)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací datum posledního dne v týdnu z data zadaného parametrem „date“.

 **Příklad:**

```
LASTDAYINWEEK(15.2.2013)
// Vrací „17.02.2013“
LASTDAYINWEEK(#today#)
```

DateTime LASTDAYINYEAR(DateTime date)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací datum posledního dne v roce z data zadaného parametrem „date“.

 **Příklad:**

```
LASTDAYINYEAR(15.2.2013)
// Vrací „31.12.2013“
LASTDAYINYEAR(#today#)
```

void LEFTALIGNMENT

 **Použití:** Název ovládacího prvku HTML – tělo ovládacího prvku musí zůstat prázdné


 **Popis:** Funkce zahájí blok ovládacích prvků, které mají název ovládacího prvku v levém sloupci zarovnaný doleva.

 **Příklad:**

```
LEFTALIGNMENT
```

void /LEFTALIGNMENT

 **Použití:** Název ovládacího prvku HTML – tělo ovládacího prvku musí zůstat prázdné


 **Popis:** Funkce zakončí blok ovládacích prvků, které mají název ovládacího prvku v levém sloupci zarovnaný doleva.

 **Příklad:**

```
/LEFTALIGNMENT
```

int LENGTH(string text)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací délku (počet znaků) textového řetězce zadaného parametrem „text“.

 **Příklad:**

```
LENGTH (Administrator)
// Vrací „13“
LENGTH (#loginname#)
LENGTH (#A#)
```

double LOG10(double number)

 **Použití:** HTML, Skript


 **Popis:** Funkce vrací logaritmus o základu 10 z hodnoty zadané parametrem „number“. Hodnoty zadané parametrem „number“ mohou být jakákoliv nezáporná čísla (i desetinná).

 **Příklad:**

```
LOG10 (1.123)
// Vrací „0,0503797562614578“
LOG10 (0)
// Vrací „-nekonečno“
```

void LOGSERVICE(string service)

 **Použití:** Skript


 **Popis:** Funkce zalogue úspěšné provedení služby s názvem „service“ do databázové tabulky „ng_sluzbawindows“. Funkce nalezne využití převážně při naplánovaném spouštění skriptů pomocí naplánovaných úloh systému Windows. Výchozí interval, po jehož uplynutí by se měla služba opět spustit, je nastaven na 1 den (1440 minut).

 **Příklad:**

```
LOGSERVICE (RunScript)
```

void LOGSERVICE(string service, int interval)

 **Použití:** Skript


 **Popis:** Funkce zaloguje úspěšné provedení služby s názvem „service“ do databázové tabulky „ng_sluzbawindows“. Funkce nalezne využití převážně při naplánovaném spouštění skriptů pomocí naplánovaných úloh systému Windows. Interval, po jehož uplynutí by se měla služba opět spustit, je definován v minutách parametrem „interval“.

 **Příklad:**

```
LOGSERVICE (RunScript, 720)
```

string LOWER(string text)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací textový řetězec zadaný parametrem „text“, ve kterém jsou všechny znaky převedeny na malá písmena.

 **Příklad:**

```
LOWER (NET Genium)  
// Vrací „net genium“
```

string LOWRES(string text1, string text2)


 **Použití:** JavaScript, Skript


 **Popis:** Funkce vrací textový řetězec zadaný parametrem „text1“, pokud je uživatel přihlášen do NET Genia na mobilním zařízení s nízkým rozlišením (při přihlašování je zaškrtnuto pole „Malý display“). V případě, že není splněna výše uvedená podmínka, funkce vrací řetězec zadaný parametrem „text2“. Funkce má využití při nastavování rozměrů ovládacích prvků v editačním formuláři pro různé typy zařízení (mobilní a nemobilní).

 **Příklad:**

```
LOWRES (abc, def)  
// Vrací „abc“ pro mobilní zařízení s nízkým rozlišením  
// Vrací „def“ pro nemobilní zařízení nebo mobilní zařízení s velkým rozlišením
```


void LOWRESOLUTION()

 **Použití:** Poznámka editačního formuláře


 **Popis:** Funkce nastaví nízké rozlišení mobilního zařízení tak, aby se editační formuláře zobrazovaly optimalizované pro nízké rozlišení i v anonymním režimu. Standardně je možnost výběru nízkého resp. velkého rozlišení umístěna na hlavní stránce v přihlašovací okně, u anonymních editačních formulářů ale tento způsob není relevantní.

 **Příklad:**

```
LOWRESOLUTION ()
```

double MEASUREWORKDAYS(DateTime startdate, DateTime enddate)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrátí počet pracovních dnů z časového úseku zadaného parametry „startdate“ a „enddate“. Funkce zohledňuje státní svátky evidované v základní aplikaci „Nastavení“.

 **Příklad:**

```
MEASUREWORKDAYS (1.1.2000, 1.2.2000)  
// Vrací „22“
```

double MEASUREWORKHOURS(DateTime startdate, DateTime enddate)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrátí počet pracovních hodin z časového úseku zadaného parametry „startdate“ a „enddate“. Funkce zohledňuje státní svátky evidované v základní aplikaci „Nastavení“, pracovní doba je stanovena na 24 hodině denně.

 **Příklad:**

```
MEASUREWORKHOURS (1.1.2000, 1.2.2000)  
// Vrací „528“
```

double MEASUREWORKHOURS(DateTime startdate, DateTime enddate, int starthour, int endhour)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací počet pracovních hodin z časového úseku zadaného parametry „startdate“ a „enddate“. Funkce zohledňuje státní svátky evidované v základní aplikaci „Nastavení“, a pracovní dobu zadanou parametry „starthour“ a „endhour“.

 **Příklad:**

```
MEASUREWORKHOURS(1.1.2000, 1.2.2000, 8, 18)
// Vrací „220“
```

string MINUTES2TIME(int minutes)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací textový řetězec ve formátu „HH:mm“ vyjádřený z počtu minut zadaného parametrem „minutes“.

 **Příklad:**

```
MINUTES2TIME(480)
// Vrací „08:00“
MINUTES2TIME(0)
// Vrací „00:00“
```

int MONTH(DateTime date)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací číslo měsíce z data zadaného parametrem „date“.

 **Příklad:**

```
MONTH(15.2.2013)
// Vrací „2“
MONTH(#now#)
```

string NGDPH()

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací aktuální sazby DPH oddělené středníkem.

 **Příklad:**

```
NGDPH ()
```

double NGDPH(int rate)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací sazbu DPH definovanou parametrem „rate“. Parametr je definován číslem 2 pro zvýšenou sazbu DPH, číslem 1 pro sníženou sazbu DPH a číslem 0 pro nulovou sazbu DPH.

 **Příklad:**

```
NGDPH (1)  
// Vrací aktuální sníženou sazbu DPH, např. „15“
```

string NGDPH(DateTime date)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací sazby DPH oddělené středníkem ke dni uvedeném v parametru „date“.

 **Příklad:**

```
NGDPH (1.1.2000)  
// Vrací sazby DPH oddělené středníkem platné k danému dni, např. „22;5;0“
```

double NGDPH(int rate, DateTime date)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací sazbu DPH definovanou parametrem „rate“. Parametr je definován číslem 2 pro zvýšenou sazbu DPH, číslem 1 pro sníženou sazbu DPH a číslem 0 pro nulovou sazbu DPH. Sazba je určena ke dni uvedeném v parametru „date“.

 **Příklad:**

```
NGDPH(2, 1.1.2000)
// Vrací zvýšenou sazbu DPH k danému dni, např. „22“
```

double NGDPH(double base, string type1, double rate, string type2)

 **Použití:** HTML, JavaScript, Skript

 **Popis:** Funkce spočítá v závislosti na parametru „type2“ buď částku, nebo hodnotu DPH. Pro výpočet je použita základní částka definovaná parametrem „base“ a sazba DPH definovaná parametrem „rate“. Parametr „type1“ specifikuje blíže hodnotu parametru „base“.

Možné hodnoty parametru „type1“:

- bezdph – parametr 1 je suma bez DPH
- sdph – parametr 1 je suma s DPH

Možné hodnoty parametru „type2“:


- bezdph – vypočítá základ daně
- dph – vypočítá hodnotu DPH
- zdph – vypočítá hodnotu DPH zaokrouhlenou na celé číslo
- dph1 – vypočítá hodnotu DPH snížené sazby
- zdph1 – vypočítá hodnotu DPH snížené sazby zaokrouhlenou na celé číslo
- dph2 – vypočítá hodnotu DPH zvýšené sazby
- zdph2 – vypočítá hodnotu DPH zvýšené sazby zaokrouhlenou na celé číslo
- sdph – vypočítá částku s DPH
- zsdph – vypočítá částku s DPH zaokrouhlenou nahoru na celé číslo
- zsdph-1 – vypočítá částku s DPH zaokrouhlenou matematicky na celé číslo
- zsdph-2 – vypočítá částku s DPH zaokrouhlenou na 50 haléřů

 **Příklad:**

```
NGDPH(-10.55, bezdph, 19, sdph)
// Vrací „-12,55“
NGDPH(100, sdph, NGDPH(1), dph)
```

string ngef(string id, string arg0, string arg1, string arg2, ...)


 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce spustí externí funkci zadanou parametrem „id“, a vrátí její návratovou hodnotu. Parametry „arg0“, „arg1“, „arg2“ atd. jsou do externí funkce předány jako „string[] args“.

 **Příklad:**

```
ngef(MyFirstFunction, A, B, C, 1, 2, 3, 1.1.2000, ...)
```

string ngef2(string id, string arg0, string arg1, string arg2, ...)

 **Použití:** Vizuální ovládací prvky


 **Popis:** Funkce spustí externí funkci zadanou parametrem „id“ ve vizuálním ovládacím prvku, a vrátí její návratovou hodnotu. Parametry „arg0“, „arg1“, „arg2“ atd. jsou do externí funkce předány jako „string[] args“. Pro samotné spuštění externí funkce je nutné, aby volání funkce „ngef2“ bylo uloženo v hodnotě ovládacího prvku „TextBox“, nastaveného „Jen ke čtení“ a jako „Skryté pole“. Textboxy, které mají jako výchozí hodnotu nastavené volání externí funkce „ngef2“, a zároveň mají zaškrtnuté vyplnění výchozí hodnoty při každém otevření editačního formuláře nebo před každým uložením záznamu do databáze, mohou mít v databázi prázdnou nebo nevyplněnou hodnotu, a přesto se při pokusu o zobrazení této hodnoty všude v aplikaci kromě editačního formuláře vyhodnotí externí funkce nastavená jako výchozí hodnota. Při volání funkce „ngef2“ se automaticky přidá poslední parametr s ID zobrazovaného záznamu.

 **Příklad:**

```
ngef2(MyFirstFunction, A, B, C, 1, 2, 3, 1.1.2000, 123, ...)
```

string NGVAT()

 **Použití:** HTML, JavaScript, Skript

 **Popis:** Funkce vrátí seznam všech států oddělených středníkem, pro které lze použít funkci „NGVAT“ s parametry.


 **Příklad:**

```
NGVAT()  
// Vrací „Afghanistan;Albania;Algeria;American  
Samoa;Andorra;Angola;Anguilla;Antarctica;Antigua and  
Barbuda;Argentina;Armenia;Aruba;Australia;Austria;Azerbaijan;Bahamas;Bahrain;Bangladesh;Barbad  
os;Belarus;Belgium;Belize;Benin;Bermuda;Bhutan;Bolivia;Bosnia and Herzegovina;Botswana;Bouvet  
Island;Brazil;British Indian Ocean Territories;Darussalam;Bulgaria;Burkina  
Faso;Burundi;Cambodia;Cameroon;Canada;Cape Verde;Cayman Islands;Central African  
Republic;Chad;Chile;China, People's Republic of;Christmas Island;Cocos  
Islands;Colombia;Comoros;Congo;Cook Islands;Costa Rica;Cote D'ivoire;Croatia;Cuba;Cyprus;Czech  
Republic;Denmark;Djibouti;Dominica;Dominican Republic;East Timor;Ecuador;Egypt;El
```

Salvador;Equatorial Guinea;Eritrea;Estonia;Ethiopia;Falkland Islands;Faroe Islands;Fiji;Finland;France;French Guiana;French Polynesia;French Southern Territories;FYROM;Gabon;Gambia;Georgia;Germany;Ghana;Gibraltar;Greece;Greenland;Grenada;Guadeloupe;Guam;Guatemala;Guinea;Guinea-Bissau;Guyana;Haiti;Heard Island And Mcdonald Islands;Honduras;Hong Kong;Hungary;Iceland;India;Indonesia;Iran;Iraq;Ireland;Israel;Italy;Jamaica;Japan;Jordan;Kazakhstan;Kenya;Kiribati;Korea, Democratic People's Republic of;Korea, Republic of;Kuwait;Kyrgyzstan;Lao Peoples Democratic Republic;Latvia;Lebanon;Lesotho;Liberia;Libyan Arab Jamahiriya;Liechtenstein;Lithuania;Luxembourg;Macau;Madagascar;Malawi;Malaysia;Maldives;Mali;Marshall Islands;Martinique;Mauritania;Mauritius;Mayotte;Mexico;Micronesia;Moldova;Monaco;Mongolia;Montserrat;Morocco;Mozambique;Myanmar;Namibia;Nauru;Nepal;Netherlands;Netherlands Antilles;New Caledonia;New Zealand;Nicaragua;Niger;Nigeria;Niue;Norfolk Island;Northern Mariana Islands;Norway;Oman;Pakistan;Palau;Panama;Papua New Guinea;Paraguay;Peru;Philippines;Pitcairn;Poland;Portugal;Puerto Rico;Qatar;Reunion;Romania;Russian Federation;Rwanda;Saint Helena;Saint Kitts and Nevis;Saint Lucia;Saint Pierre and Miquelon;Saint Vincent and The Grenadines;Samoa;San Marino;Sao Tome and Principe;Saudi Arabia;Senegal;Seychelles;Sierra Leone;Singapore;Slovakia;Slovenia;Solomon Islands;Somalia;South Africa;South Georgia and Sandwich Islands;Spain;Sri Lanka;Sudan;Suriname;Svalbard and Jan Mayen;Swaziland;Sweden;Switzerland;Syrian Arab Republic;Taiwan;Tajikistan;Tanzania;Thailand;Togo;Tokelau;Tonga;Trinidad and Tobago;Tunisia;Turkey;Turkmenistan;Turks and Caicos Islands;Tuvalu;Uganda;Ukraine;United Arab Emirates;United Kingdom;USA;Uruguay;Uzbekistan;Vanuatu;Vatican City State;Venezuela;Vietnam;Virgin Islands (British);Virgin Islands (U.S.);Wallis And Futuna Islands;Western Sahara;Yemen;Yugoslavia;Zaire;Zambia;Zimbabwe"

double NGVAT(double base, DateTime date, string country, string dic)

 **Použití:** HTML, JavaScript, Skript

 **Popis:** Funkce vrací vypočtenou částku DPH, kterou je třeba zaplatit při dodávce zboží do státu zadaného parametrem „country“. Parametr „base“ definuje peněžní základ. Parametr „date“ definuje datum zdanitelného plnění a parametr „dic“ definuje DIČ. Pokud dodavatel nemá přidělené DIČ, parametr bude prázdný.

 **Příklad:**

```
NGVAT(1500.20, #today#, Germany, CZ27092381)
```

string NGVATRATES(string country, DateTime date)

Použití: HTML, JavaScript, Skript

Popis: Funkce vrací sazby DPH oddělené středníkem v dané zemi (CZ, DE, HU, SK, ...) zadané parametrem „country“, a k danému datu zadaném parametrem „date“.

Příklad:

```
NGVATRATES(DE, 5.5.2012)
// Vrací „19;7;0“
```

void NOBACKGROUND

Použití: Název ovládacího prvku HTML – tělo ovládacího prvku musí zůstat prázdné

Popis: Funkce zahájí blok ovládacích prvků, které nepoužívají barevné pozadí v levém sloupci s názvem ovládacího prvku. Zahájený blok, který není ukončen pomocí funkce „/NOBACKGROUND“, ukotví tlačítka „Uložit“, „Smazat“ a „Zpět“. Tyto tlačítka pak nebudou plovoucí, a budou se zobrazovat vždy až na konci editačního formuláře.

Příklad:

```
NOBACKGROUND
```

void /NOBACKGROUND

Použití: Název ovládacího prvku HTML – tělo ovládacího prvku musí zůstat prázdné


Popis: Funkce zakončí blok ovládacích prvků, které nepoužívají barevné pozadí v levém sloupci s názvem ovládacího prvku.

Příklad:

```
/NOBACKGROUND
```

int OCR2FILE(string path, string type)

 **Použití:** Skript


 **Popis:** Funkce převede metodou OCR text ze souboru zadaného parametrem „path“, a vytvoří z něho novou souborovou přílohu definovanou parametrem „type“. Převod ze souborů MS Word nebo Excel je prováděn nativně, v ostatních případech využívá OCR server. Zdrojový soubor je definován parametrem „path“ – plnou cestou k souboru na disku. Výsledkem funkce je ID nově vytvořené souborové přílohy. Podporované typy výstupů jsou „doc“, „docx“, „htm“, „pdf“, „rtf“, „txt“, „xlsx“, „xml“.

 **Příklad:**

```
OCR2FILE(#rootpath#Temp\zdroj.xlsx, pdf)
```

int OCR2FILE(string path, string type, string url)

 **Použití:** Skript


 **Popis:** Funkce převede metodou OCR text ze souboru zadaného parametrem „path“, a vytvoří z něho novou souborovou přílohu definovanou parametrem „type“. Převod ze souborů MS Word nebo Excel je prováděn nativně, v ostatních případech využívá OCR server. Zdrojový soubor je definován parametrem „path“ – plnou cestou k souboru na disku. Výsledkem funkce je ID nově vytvořené souborové přílohy. Podporované typy výstupů jsou „doc“, „docx“, „htm“, „pdf“, „rtf“, „txt“, „xlsx“, „xml“. Samotné OCR rozpoznávání je prováděno samostatnou webovou aplikací na adrese definované parametrem „url“.

 **Příklad:**

```
OCR2FILE(#rootpath#Temp\zdroj.xlsx, xls, https://www.netgenium.com/netgenium_with_ocr)
```

int OCR2FILE(int file, string type)

 **Použití:** Skript


 **Popis:** Funkce převede metodou OCR text ze souborové přílohy zadané parametrem „file“, a vytvoří z ní novou souborovou přílohu definovanou parametrem „type“. Převod ze souborů MS Word nebo Excel je prováděn nativně, v ostatních případech využívá OCR server. Zdrojová souborová příloha je definována parametrem „file“ – ID souborové přílohy. Výsledkem funkce je ID nově vytvořené souborové přílohy. Podporované typy výstupů jsou „doc“, „docx“, „htm“, „pdf“, „rtf“, „txt“, „xlsx“, „xml“.

 **Příklad:**

```
OCR2FILE(#ng_priloha#, docx)
```


int OCR2FILE(int file, string type, string url)

 **Použití:** Skript


 **Popis:** Funkce převede metodou OCR text ze souborové přílohy zadané parametrem „file“, a vytvoří z ní novou souborovou přílohu definovanou parametrem „type“. Převod ze souborů MS Word nebo Excel je prováděn nativně, v ostatních případech využívá OCR server. Zdrojová souborová příloha je definována parametrem „file“ – ID souborové přílohy. Výsledkem funkce je ID nově vytvořené souborové přílohy. Podporované typy výstupů jsou „doc“, „docx“, „htm“, „pdf“, „rtf“, „txt“, „xlsx“, „xml“. Samotné OCR rozpoznávání je prováděno samostatnou webovou aplikací na adrese definované parametrem „url“.

 **Příklad:**

```
OCR2FILE(#ng_priloha#, pdf, https://www.netgenium.com/netgenium_with_ocr)
```

string OCR2HTML(string path)

 **Použití:** Skript


 **Popis:** Funkce převede metodou OCR text ze souboru zadaného parametrem „path“, a vrací ho jako textový řetězec ve formátu HTML. Převod ze souborů MS Word nebo Excel je prováděn nativně, v ostatních případech využívá OCR server. Zdrojový soubor je definován parametrem „path“ – plnou cestou k souboru na disku.

 **Příklad:**

```
OCR2HTML(#rootpath#Temp\zdroj.xlsx)
```

string OCR2HTML(string path, string url)

 **Použití:** Skript


 **Popis:** Funkce převede metodou OCR text ze souboru zadaného parametrem „path“, a vrací ho jako textový řetězec ve formátu HTML. Převod ze souborů MS Word nebo Excel je prováděn nativně, v ostatních případech využívá OCR server. Zdrojový soubor je definován parametrem „path“ – plnou cestou k souboru na disku. Samotné OCR rozpoznávání je prováděno samostatnou webovou aplikací na adrese definované parametrem „url“.

 **Příklad:**

```
OCR2HTML(#rootpath#Temp\zdroj.xlsx, https://www.netgenium.com/netgenium_with_ocr)
```

string OCR2HTML(int file)

 **Použití:** Skript


 **Popis:** Funkce převede metodou OCR text ze souborové přílohy zadané parametrem „file“, a vrací ho jako textový řetězec ve formátu HTML. Převod ze souborů MS Word nebo Excel je prováděn nativně, v ostatních případech využívá OCR server. Zdrojová souborová příloha je definována parametrem „file“ – ID souborové přílohy.

 **Příklad:**

```
OCR2HTML(#ng_priloha#)
```

string OCR2HTML(int file, string url)

 **Použití:** Skript


 **Popis:** Funkce převede metodou OCR text ze souborové přílohy zadané parametrem „file“, a vrací ho jako textový řetězec ve formátu HTML. Převod ze souborů MS Word nebo Excel je prováděn nativně, v ostatních případech využívá OCR server. Zdrojová souborová příloha je definována parametrem „file“ – ID souborové přílohy. Samotné OCR rozpoznávání je prováděno samostatnou webovou aplikací na adrese definované parametrem „url“.

 **Příklad:**

```
OCR2HTML(#ng_priloha#, https://www.netgenium.com/netgenium_with_ocr)
```

string OCR2TEXT(string path)

 **Použití:** Skript


 **Popis:** Funkce převede metodou OCR text ze souboru zadaného parametrem „path“, a vrací ho jako prostý textový řetězec bez HTML formátování. Převod ze souborů MS Word nebo Excel je prováděn nativně, v ostatních případech využívá OCR server. Zdrojový soubor je definován parametrem „path“ – plnou cestou k souboru na disku.

 **Příklad:**

```
OCR2TEXT(#rootpath#Temp\zdroj.xlsx)
```

string OCR2TEXT(string path, string url)

 **Použití:** Skript


 **Popis:** Funkce převede metodou OCR text ze souboru zadaného parametrem „path“, a vrací ho jako prostý textový řetězec bez HTML formátování. Převod ze souborů MS Word nebo Excel je prováděn nativně, v ostatních případech využívá OCR server. Zdrojový soubor je definován parametrem „path“ – plnou cestou k souboru na disku. Samotné OCR rozpoznávání je prováděno samostatnou webovou aplikací na adrese definované parametrem „url“.

 **Příklad:**

```
OCR2TEXT(#rootpath#Temp\zdroj.xlsx, https://www.netgenium.com/netgenium_with_ocr)
```

string OCR2TEXT(int file)

 **Použití:** Skript


 **Popis:** Funkce převede metodou OCR text ze souborové přílohy zadané parametrem „file“, a vrací ho jako prostý textový řetězec bez HTML formátování. Převod ze souborů MS Word nebo Excel je prováděn nativně, v ostatních případech využívá OCR server. Zdrojová souborová příloha je definována parametrem „file“ – ID souborové přílohy.

 **Příklad:**

```
OCR2TEXT(#ng_priloha#)
```

string OCR2TEXT(int file, string url)

 **Použití:** Skript


 **Popis:** Funkce převede metodou OCR text ze souborové přílohy zadané parametrem „file“, a vrací ho jako prostý textový řetězec bez HTML formátování. Převod ze souborů MS Word nebo Excel je prováděn nativně, v ostatních případech využívá OCR server. Zdrojová souborová příloha je definována parametrem „file“ – ID souborové přílohy. Samotné OCR rozpoznávání je prováděno samostatnou webovou aplikací na adrese definované parametrem „url“.

 **Příklad:**

```
OCR2TEXT(#ng_priloha#, https://www.netgenium.com/netgenium_with_ocr)
```

string PADLEFT(string text, int totalwidth, string pad)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce doplní textový řetězec zadaný parametrem „text“ zleva znakem zadaným parametrem „pad“ na délku zadanou parametrem „totalwidth“.

 **Příklad:**

```
PADLEFT(x, 5, 0)
// Vrací „0000x“
```

string PADRIGHT(string text, int totalwidth, string pad)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce doplní řetězec zadaný parametrem „text“ zprava znakem zadaným parametrem „pad“ na délku zadanou parametrem „totalWidth“.

 **Příklad:**

```
PADRIGHT(x, 5, 0)
// Vrací „x0000“
```

string PAGEIDENTIFIER(string page)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací identifikátor webové stránky zadané parametrem „page“.

 **Příklad:**

```
PAGEIDENTIFIER(Kalendář akcí)
// Vrací „kalendar-akci“
```

string PAGEIDENTIFIER2(string page)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací identifikátor webové stránky zadané parametrem „page“.

 **Příklad:**

```
PAGEIDENTIFIER2(Kalendář akcí)
// Vrací „KalendarAkci“
```

string PLACESONMAP(int id)

 **Použití:** Skript


 **Popis:** Funkce vrací HTML kód mapy zadané parametrem „id“ včetně všech míst na mapě definovaných v aplikaci „Mapy“.

 **Příklad:**

```
PLACESONMAP (14)
```

double POWER(double number)

 **Použití:** HTML, Skript


 **Popis:** Funkce vrací druhou mocninu čísla zadaného parametrem „number“.

 **Příklad:**

```
POWER(-1.23)
// Vrací „1,5129“
POWER(5)
// Vrací „25“
POWER(#A#)
```

int PRINTATTACHMENT(int button, int id)


 **Použití:** Skript


 **Popis:** Funkce spustí tisk definovaný na ovládacím prvku zadaném parametrem „button“, výslednou tiskovou sestavu uloží jako novou souborovou přílohu, a vrátí ID této přílohy. Parametr „button“ značí ID tlačítka, které obsluhuje tisk dané tiskové šablony, a parametr „id“ značí ID databázového záznamu, který má být vytištěn.

 **Příklad:**

```
PRINTATTACHMENT(123, 1)
```

void PROTECTFORM(int form, int id, string message)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce ověří, zda je otevřen záznam zadaný parametrem „id“ v editačním formuláři zadaném parametrem „form“. Pokud ano, vyvolá přerušení zadané parametrem „message“. Za parametr „message“ se automaticky doplní text „Záznam je uzamčen uživatelem {0} od {1}.“, kde {0} představuje jméno uživatele, který má záznam otevřený a {1} představuje čas, do kterého je záznam uzamčen. Volání funkce má smysl pouze v případě, kdy je formulář nastaven pro uzamykání záznamů.

 **Příklad:**

```
PROTECTFORM(1, 1, Není dovolena editace záznamu.)  
// Vrací například „Není dovolena editace záznamu. Záznam je uzamčen uživatelem Administrator  
od 12:30.“
```

void PROTECTSCRIPT(int form, int n, string message)

 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce ověří, zda je spuštěn skript zadaný parametrem „form“ a „n“. Pokud ano, vyvolá přerušení zadané parametrem „message“. Parametry „form“ a „n“ je možné zjistit v administrátorském režimu v tooltipu, který se zobrazí po zastavení ukazatele myši nad odkazem „Upravit skript“.

 **Příklad:**

```
PROTECTSCRIPT(1, 10, Není dovoleno spuštění skriptu ABC. Vyčkejte prosím na jeho dokončení.)
```

string READURL(string text)

 **Použití:** HTML


 **Popis:** Funkce načte obsah webové stránky definované parametrem „text“, a vrátí její obsah.

 **Příklad:**

```
READURL (https://www.netgenium.com)
```

void RECYCLEPOOL(string applicationpool)

 **Použití:** Skript


 **Popis:** Funkce recykluje aplikační pool zadaný parametrem „applicationpool“. Skript, ve kterém je uvedena funkce „RECYCLEPOOL“, musí být spuštěn výhradně pomocí programu „RunScript.exe“.

 **Příklad:**

```
RECYCLEPOOL (DefaultAppPool)
```

void REDUCEATTACHMENT(int file, int maxwidth, int maxheight)

 **Použití:** Skript


 **Popis:** Funkce zmenší obrázek zadaný parametrem „file“ na maximální velikost o šířce zadané parametrem „maxwidth“ a výšce zadané parametrem „maxheight“, a uloží do formátu „jpg“. Pro zmenšení obrázku bude použita nejvyšší možná kvalita.

 **Příklad:**

```
REDUCEATTACHMENT (#ng_obrazek#, 500, 500)
```

void REDUCEATTACHMENT(int file, int maxwidth, int maxheight, bool highquality)

 **Použití:** Skript


 **Popis:** Funkce zmenší obrázek zadaný parametrem „file“ na maximální velikost o šířce zadané parametrem „maxwidth“ a výšce zadané parametrem „maxheight“, a uloží do formátu „jpg“. Parametr „highquality“ nastavený na „true“ značí, že pro zmenšení obrázku bude použita nejvyšší možná kvalita. V opačném případě bude použita standardní kvalita.

 **Příklad:**

```
REDUCEATTACHMENT(#ng_obrazek#, 500, 500, true)
```

string REMOVEDIACRITICS(string text)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce odebere diakritiku z textového řetězce zadaného parametrem „text“.

 **Příklad:**

```
REMOVEDIACRITICS(Přiliš žlutoučký kůň pěl ďábelské ódy)  
// Vrací „Prilis zlutoucky kun pel dabelske ody“
```

string REMOVEIFCONTAINS(string text, string search1[, search2, ...])

 **Použití:** HTML, JavaScript, Skript

 **Popis:** Funkce vynuluje parametr „text“, pokud obsahuje podřetězec „search1“ nebo „search2“ nebo „search3“, atd.

 **Příklad:**

```
REMOVEIFCONTAINS(Faktura je zaevidována, je)  
// Vrací „“ (prázdný řetězec)
```


string REMOVEIFENDSWITH(string text, string search1[, search2, ...])

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vynuluje parametr „text“, pokud končí podřetězcem „search1“ nebo „search2“ nebo „search3“, atd.

 **Příklad:**

```
REMOVEIFENDSWITH(Faktura je zaevidována, váno, vány, vána)
// Vrací „“ (prázdný řetězec)
```

string REMOVEIFEQUALS(string text, string search1[, search2, ...])

 **Použití:** HTML, JavaScript, Skript

 **Popis:** Funkce vynuluje parametr „text“, pokud se shoduje s podřetězcem „search1“ nebo „search2“ nebo „search3“, atd.

 **Příklad:**

```
REMOVEIFEQUALS(Schváleno, Schváleno)
// Vrací „“ (prázdný řetězec)
```

string REMOVEIFSTARTSWITH(string text, string search1[, search2, ...])

 **Použití:** HTML, JavaScript, Skript

 **Popis:** Funkce vynuluje parametr „text“, pokud začíná podřetězcem „search1“ nebo „search2“ nebo „search3“, atd.

 **Příklad:**

```
REMOVEIFSTARTSWITH(Faktura je zaevidována, Fact, Fakt, Faak)
// Vrací „“ (prázdný řetězec)
```

string REMOVETAGS(string html)

 **Použití:** HTML, JavaScript, Skript

 **Popis:** Funkce odstraní všechny tagy z řetězce zadaného parametrem „html“.

 **Příklad:**

```
REMOVETAGS (<p>abc</p><p>abc</p>)  
// Vrací „abcabc“
```

string REMOVEWHITESPACES(string value)


 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce odstraní z řetězce zadaného parametrem „value“ všechna prázdná místa (mezery a tabulátory).

 **Příklad:**

```
REMOVEWHITESPACES (a b c);  
// Vrací „abc“
```

void RENAMEATTACHMENT(int file, string name)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce přejmenuje souborovou přílohu zadanou parametrem „file“ na jméno zadané parametrem „name“ bez přípony.

 **Příklad:**

```
RENAMEATTACHMENT(#ng_priloha#, Nové jméno souboru)
```

void RENAMEATTACHMENT(int file, string prefix, string postfix)

 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce přejmenuje souborovou přílohu zadanou parametrem „file“ na nové jméno, které vznikne spojením hodnoty parametru „prefix“, původního názvu souboru bez přípony, hodnoty parametru „postfix“, a následně přípony.

 **Příklad:**

```
RENAMEATTACHMENT(#ng_priloha#, A-, -B)
```

string REPLACE(string value, string oldvalue, string newvalue)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací textový řetězec zadaný parametrem „value“, ve kterém byly nahrazeny všechny výskyty řetězce uloženého v parametru „oldvalue“ (vzor) řetězcem uloženým v parametru „newvalue“ (náhrada).

 **Příklad:**

```
REPLACE(Dobrý den NET Genium, e, é)  
// Vrací „Dobrý dén NET Génium“
```

string REPLACEIFEQUALS(string compare1, string compare2, string text)

 **Použití:** HTML, JavaScript, Skript

 **Popis:** Funkce vrací hodnotu parametru „text“, je-li hodnota parametru „compare1“ rovna hodnotě parametru „compare2“. V opačném případě vrací hodnotu parametru „compare1“.

 **Příklad:**

```
REPLACEIFEQUALS(abc, abc, def)  
// Vrací „def“
```

string REPLACETAB(string items, string olditem, string newitem)

Použití: HTML, JavaScript, Skript

Popis: Funkce vrací seznam položek oddělených tabulátorem zadaný parametrem „items“, ve kterém byly nahrazeny všechny položky rovné parametru „oldvalue“ (vzor) za hodnotu zadanou parametrem „newvalue“ (náhrada).

Příklad:

```
REPLACETAB(abc#tab#abcd#tab#def, abc, def)
// Vrací „def#tab#abcd#tab#def“
```

string REPLYBODY(string from, string to, DateTime date, string body)

Použití: HTML, JavaScript, Skript

Popis: Funkce vrací hodnotu parametru „body“ jako naformátovanou emailovou odpověď. Tělo odesílané emailové zprávy začíná volným řádkem, kam lze napsat text jako odpověď na doručený e-mail. Na dalším řádku je zobrazen aktuální čas odpovědi definovaný parametrem „date“ a e-mailová adresa uložená v parametru „from“. Tento parametr představuje e-mailovou adresu odesílatele původní zprávy. Na dalších řádcích se nachází text původní zprávy (parametr „body“) ohraničený po levé straně svislou čárou. Parametr „to“ je emailová adresa příjemce původní zprávy a momentálně s ním funkce „REPLYBODY“ nepracuje.

Příklad:

```
REPLYBODY(info@netgenium.com, info@netgenium.com, #now#, body)
```

string REQUESTFORM(string name)

Použití: Skript

Popis: Funkce při uskutečnění HTML postbacku vrací hodnotu formulářového prvku zadaného parametrem „name“.

Příklad:

```
<input name="C5$tb" type="text" value="Nová faktura" id="C5_tb" style="width:110px;">
REQUESTFORM(C5$tb)
// Vrací „Nová faktura“
```

string REQUESTQUERYSTRING(string attribute)


 **Použití:** Skript


 **Popis:** Funkce vrací hodnotu atributu z aktuální URL. Název atributu je specifikován parametrem „attribute“.

 **Příklad:**

```
https://www.netgenium.com?id=123&next=true  
REQUESTQUERYSTRING(id)  
// Vrací „123“
```

void RESETDATAGRIDSETTINGS(int query)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce resetuje nastavení datagridu zadaného parametrem „query“ tím, že zruší hledaný výraz, zruší zaškrtnutí hledání v jednotlivých sloupcích, a nastaví hodnotu všech filtrů na „(Vše)“.

 **Příklad:**

```
RESETDATAGRIDSETTINGS (123)
```

void RESETUSERPASSWORD(string ng_checkbox, string from, string subject, string message)

 **Použití:** Skript (na místě, kde se očekává zadání komentáře)

 **Popis:** Funkce resetuje heslo uživateli, který má označen checkbox zadaný parametrem „ng_checkbox“. Uživatel je o tomto resetu obeznámen emailem, který mu přijde z adresy zadané parametrem „from“, s předmětem zadaným parametrem „subject“ a textem zadaným parametrem „message“. Uživatel je po následném přihlášení vyzván ke změně hesla.

 **Příklad:**

```
RESETUSERPASSWORD(ng_oznacetproodeslanihesla, #ng_odesilatel#, #ng_predmet#, #ng_zprava#)  
RESETUSERPASSWORD(ng_oznacetproodeslanihesla, noreply@netgenium.com, Expirace hesla, Změňte prosím své heslo.)
```

 **Příklad e-mailu:**


Změňte prosím své heslo.

Přihlašovací jméno: novak.petr

Heslo: kR2FP50h

string RESOURCE(int index)

 **Použití:** HTML, JavaScript, Skript

 **Popis:** Funkce vrací položku ze seznamu zdrojů zadanou jejím indexem, resp. parametrem „index“. Zdroj je třeba nadefinovat v souboru „resources.txt“ v adresáři „NETGenium\Config\Resources\jazykové_nastavení“. Index používá číslování od nuly.

Příklad obsahu souboru:


- 0="10 minut"
- 1="10 vteřin"
- 2="1 desetinné místo"

 **Příklad:**

```
RESOURCE (2)
```

void RESTARTSERVICE(string service)


 **Použití:** Skript


 **Popis:** Funkce restartuje službu zadanou parametrem „service“. Skript, ve kterém je uvedena funkce „RESTARTSERVICE“, musí být spuštěn výhradně pomocí programu „RunScript.exe“.

 **Příklad:**

```
RESTARTSERVICE (CRMService)
```

void RESTOREFROMHISTORY(int form, int id)

 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce obnoví smazaný záznam z historie zadaný parametrem „id“ v editačním formuláři zadaném parametrem „form“.

 **Příklad:**

```
RESTOREFROMHISTORY (16, 6620)
```

double ROUND(double number)

 **Použití:** HTML, Skript


 **Popis:** Funkce vrací zaokrouhlenou hodnotu čísla zadaného parametrem „number“.

 **Příklad:**

```
ROUND (-1.63)  
// Vrací „-2“  
ROUND (1.63)  
// Vrací „2“
```

double ROUND(double number, double decimals)

 **Použití:** HTML, ve skriptu není funkce není dostupná (je nutné použít například „ROUND(... * 100)/100“)


 **Popis:** Funkce vrací číslo zadané parametrem „number“ zaokrouhlené na počet desetinných míst zadaných parametrem „decimals“.

 **Příklad:**

```
ROUND (-1.636, 2)  
// Vrací „-1,64“  
ROUND (1.636, 2)  
// Vrací „1,64“
```

string SCALEIMAGES(string html, int maxWidth, int maxHeight)

 **Použití:** Skript


 **Popis:** Funkce vrátí obrázky s upravenými rozměry zadané v parametru „html“. U všech upravených obrázků budou proporciónálně nastaveny parametry „width“ a „height“ v závislosti na parametrech „maxWidth“ a „maxHeight“.

 **Příklad:**

```
SCALEIMAGES(#ng_html#, 20, 15)
```

string SEARCHFILES(string subject, string content, int results)

 **Použití:** HTML


 **Popis:** Funkce vyhledá souborové přílohy dle zadaných kritérií, a vrátí tabulku se třemi sloupci, kde v prvním sloupci se nachází název souborové přílohy, ve druhém její velikost, a ve třetím sloupci datum vytvoření. Souborové přílohy jsou hledány fulltextově dle jejich názvu definovaného parametrem „subject“ a dle obsahu, který definuje parametr „content“. Parametr „results“ určuje, kolik nalezených výskytů souborových příloh se má v tabulce zobrazit. Použití parametru „content“ má smysl pouze v případě, jsou-li souborové přílohy indexovány pomocí služby Indexing Service nebo Windows Search.

 **Příklad:**

```
SEARCHFILES(NETGenium.gif, , 100)  
SEARCHFILES(Název, Obsah, 100)
```

string SEARCHFILES(string subject, string content, int results, DateTime date1)

 **Použití:** HTML


 **Popis:** Funkce vyhledá souborové přílohy dle zadaných kritérií a vrátí tabulku se třemi sloupci, kde v prvním sloupci se nachází název souborové přílohy, ve druhém její velikost, a ve třetím sloupci datum vytvoření. Souborové přílohy jsou hledány fulltextově dle jejich názvu definovaného parametrem „subject“ a dle obsahu, který definuje parametr „content“. Parametr „results“ určuje, kolik nalezených výskytů souborových příloh se má v tabulce zobrazit. Parametr „date1“ dále určuje, že se mají vyhledávat pouze přílohy s datem vytvoření větším nebo rovno než zadané datum. Použití parametrů „content“ a „date1“ má smysl pouze v případě, jsou-li souborové přílohy indexovány pomocí služby Indexing Service nebo Windows Search.

 **Příklad:**

```
SEARCHFILES(NETGenium.gif, , 100, 1.1.2013)  
SEARCHFILES(Název, Obsah, 100, 1.1.2013)
```


string SEARCHFILES(string subject, string content, int results, DateTime date1, DateTime date2)

 **Použití:** HTML


 **Popis:** Funkce vyhledá souborové přílohy dle zadaných kritérií a vrátí tabulku se třemi sloupci, kde v prvním sloupci se nachází název souborové přílohy, ve druhém její velikost, a ve třetím sloupci datum vytvoření. Souborové přílohy jsou hledány fulltextově dle jejich názvu definovaného parametrem „subject“ a dle obsahu, který definuje parametr „content“. Parametr „results“ určuje, kolik nalezených výskytů souborových příloh se má v tabulce zobrazit. Parametry „date1“ a „date2“ dále určují, že se budou vyhledávat pouze přílohy s datem vytvoření mezi těmito dvěma zadanými daty včetně. Použití parametrů „content“, „date1“ a „date2“ má smysl pouze v případě, jsou-li souborové přílohy indexovány pomocí služby Indexing Service nebo Windows Search.

 **Příklad:**

```
SEARCHFILES(NETGenium.gif, , 100, 1.1.2013, 31.12.2013)
SEARCHFILES(Název, Obsah, 100, 1.1.2013, 31.12.2013)
```

string SEARCHNEXTID(string array, string id)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce přijímá pole hodnot oddělených středníkem, které je zadané parametrem „array“. V tomto poli je vyhledána hodnota zadaná parametrem „id“, a funkce poté vrátí hodnotu, která se v zadaném poli nachází bezprostředně za touto hodnotou. Pokud se zadaná hodnota „id“ v poli „array“ nenachází nebo se za zadanou hodnotou již žádná hodnota v poli nenachází, funkce vrátí „0“. Pokud je hodnota „id“ rovna „0“, vrátí funkce první hodnotu zadaného pole.

 **Příklad:**

```
SEARCHNEXTID(1;2;3, 2)
// Vrací „3“
SEARCHNEXTID(1;2;3, 0)
// Vrací „1“
SEARCHNEXTID(dva;čtyři;šest, čtyři)
// Vrací „šest“
```

string SEARCHPREVID(string array, string id)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce přijímá pole hodnot oddělených středníkem, které je zadané parametrem „array“. V tomto poli je vyhledána hodnota zadaná parametrem „id“, a funkce poté vrátí hodnotu, která se v zadaném poli nachází bezprostředně před touto hodnotou. Pokud se zadaná hodnota „id“ v poli „array“ nenachází nebo se před zadanou hodnotou již žádná hodnota v poli nenachází, funkce vrátí „0“. Pokud je hodnota „id“ rovna „0“, vrátí funkce poslední hodnotu zadaného pole.

 **Příklad:**

```
SEARCHPREVID(1;2;3, 2)
// Vrací „1“
SEARCHPREVID(1;2;3, 0)
// Vrací „3“
SEARCHPREVID(dva;čtyři;šest, čtyři)
// Vrací „dva“
```

string SECONDS2TIME(double seconds)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrátí textový řetězec ve formátu „HH:mm:ss“ vyjádřený z počtu vteřin zadaného parametrem „seconds“.

 **Příklad:**

```
SECONDS2TIME(480)
// Vrací „00:08:00“
SECONDS2TIME(0)
// Vrací „00:00:00“
```

void SENDCHATMESSAGE(string message)

 **Použití:** Skript


 **Popis:** Funkce odešle chatovou zprávu, jejíž text je definován parametrem „message“. Zpráva je odeslána všem aktuálně přihlášeným uživatelům.

 **Příklad:**

```
SENDCHATMESSAGE>Hello)
```

void SENDCHATMESSAGE(string message, string loginname)


 **Použití:** Skript


 **Popis:** Funkce odešle chatovou zprávu, jejíž text je definován parametrem „message“. Zpráva je odeslána uživateli, který je zadán parametrem „loginname“.

 **Příklad:**

```
SENDCHATMESSAGE (Hello, Administrator)
```

string SENDMAILINVITATION(string from, string replyto, string to, string cc, string bcc, string subject, string body, string attachments, bool save, string summary, string location, DateTime start, DateTime end, string description [, string c0, string v0 [, string c1, string v1...]])


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce odešle pozvánku na schůzku prostřednictvím e-mailové adresy zadané parametrem „from“ na e-mailovou adresu zadanou parametrem „to“, a vrátí UID pozvánky. Parametr „replyto“ značí, na kterou e-mailovou adresu se má na tuto pozvánku defaultně odpovědět. Parametr „summary“ definuje předmět schůzky, parametr „location“ pak místo konání schůzky. Parametry „start“ a „end“ označují začátek a konec schůzky. Parametr „description“ nabízí možnost vložení podrobnějšího i víceřádkového popisu dané schůzky. Ostatní parametry jsou vysvětleny u funkce „SCHEDULEMAILMESSAGE“.

 **Příklad:**

```
SENDMAILINVITATION(info@mojefirma.cz, , info@netgenium.com, , , E-mailová zpráva, Vážený pane..., 0, True, Předmět schůzky, Místo konání, 25.4.2013 15:00, 25.4.2013 20:00, Popis schůzky)
```

void SENDMAILMESSAGE(string from, string replyto, string to, string cc, string bcc, string subject, string body, string attachments, bool save [, string c0, string v0 [, string c1, string v1...]])

 **Použití:** Skript (na místě, kde se očekává zadání komentáře)

 **Popis:** Funkce odešle e-mailovou zprávu z e-mailové adresy zadané parametrem „from“ na e-mailovou adresu zadanou parametrem „to“. Parametr „replyto“ značí, na kterou e-mailovou adresu se má na tento e-mail defaultně odpovědět. Ostatní parametry jsou vysvětleny u funkce „SCHEDULEMAILMESSAGE“.


 **Příklad:**


```
SENDMAILMESSAGE(info@mojefirma.cz, , info@netgenium.com, , , E-mailová zpráva, Vážený pane..., 0, True)
SENDMAILMESSAGE(info@mojefirma.com, , info@netgenium.com, , , E-mailová zpráva, Vážený pane..., 1;#ng_priloha#;D:\priloha.txt, true)
```

```
void SENDMAILMESSAGE2(string from, string replyto, string to, string cc, string bcc, string subject, string body, string attachments, bool save [, string c0, string v0 [, string c1, string v1...]])
```

Funkce je identická k funkci SENDMAILMESSAGE, avšak využívá druhý SMTP server - v nastavení NET Genia „SMTP Server 2“.

void SETCOOKIE(string key, string value, DateTime expires)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce nastaví hodnotu cookie zadané parametrem „key“ na hodnotu definovanou parametrem „value“, s datem expirace definovaným parametrem „expires“. Parametr „SameSite“ bude nastaven na hodnotu „Strict“, parametr „Secure“ bude zvolen podle konfigurace NET Genia (konfigurační soubor „SecureCookies.txt“, „SSL.txt“, nebo zabezpečená komunikace přes protokol „https“), a parametr „HttpOnly“ bude nastaven na hodnotu „False“.

 **Příklad:**

```
SETCOOKIE(abc, def, #today# + 6m)
```

void SETVARIABLE(string variable, string value)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce nastaví hodnotu proměnné zadané parametrem „variable“ na hodnotu definovanou parametrem „value“.

 **Příklad:**

```
SETVARIABLE(A, abc)
```

void SCHEDULEBACK()


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce naplánuje otevření předchozí stránky ihned po úspěšném provedení skriptu.

 **Příklad:**

```
SCHEDULEBACK ()
```

void SCHEDULEBACK(string message)

 **Použití:** Skript (na místě, kde se očekává zadání komentáře)

 **Popis:** Funkce naplánuje otevření předchozí stránky ihned po úspěšném provedení skriptu. Po otevření předchozí stránky je uživateli zobrazena zpráva v dialogovém okně zadaná parametrem „message“.

 **Příklad:**

```
SCHEDULEBACK(abc)  
SCHEDULEBACK("Zpráva, která obsahuje čárku.")
```

void SCHEDULEBACK(string message, bool dialog)

Použití: Skript (na místě, kde se očekává zadání komentáře)

Popis: Funkce naplánuje otevření předchozí stránky ihned po úspěšném provedení skriptu. Po otevření předchozí stránky je uživateli zobrazena zpráva zadaná parametrem „message“ buď v dialogovém okně, nebo pouze jako červeně zvýrazněný text zobrazený ve spodní části stránky.

Příklad:

```
SCHEDULEBACK(abc, false)
SCHEDULEBACK("Zpráva, která obsahuje čárku.", false)
```

void SCHEDULEDEFAULTVALUE(string dbname, string value)

Použití: Skript (na místě, kde se očekává zadání komentáře)

Popis: Funkce naplánuje vyplnění hodnoty zadané parametrem „value“ do ovládacího prvku definovaného identifikátorem „dbname“ při otevření editačního formuláře ihned po úspěšném provedení skriptu.

Příklad:

```
SCHEDULEDEFAULTVALUE (ng_stav, Dokončeno)
```

void SCHEDULEDEFAULTVALUE(string dbname, string dbtype, string value)


Použití: Skript (na místě, kde se očekává zadání komentáře)


Popis: Funkce naplánuje vyplnění hodnoty zadané parametrem „value“ do ovládacího prvku definovaného identifikátorem „dbname“ při otevření editačního formuláře ihned po úspěšném provedení skriptu. Parametr „dbtype“ určuje databázový typ hodnoty zadané parametrem „value“. Databázový typ je nutné uvádět u hodnot, které jsou typu „Integer“, „Double“ nebo „Date“. V případě, že není zadán, bude použit databázový typ „String“.

Příklad:

```
SCHEDULEDEFAULTVALUE (ng_datum, Date, #today# - 20)
```

void SCHEDULEDEFAULTVALUES(string dbname1, string dbname2, ... , string values)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce naplňuje vyplnění hodnot oddělených středníkem zadaných parametrem „values“ do ovládacích prvků zadaných identifikátory „dbname1“, „dbname2“, atd... při otevření editačního formuláře ihned po úspěšném provedení skriptu.

 **Příklad:**

```
SCHEDULEDEFAULTVALUES (ng_text1, ng_text2, #A#)
SCHEDULEDEFAULTVALUES (ng_text1, ng_text2, hodnota1;hodnota2)
```

void SCHEDULEENTRY(int id)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce naplňuje otevření záznamu s primárním klíčem zadaným parametrem „id“ v aktuálně otevřeném editačním formuláři ihned po úspěšném provedení skriptu.

 **Příklad:**

```
SCHEDULEENTRY (123)
```

void SCHEDULEENTRY(string column, int id)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce naplňuje otevření vyhledaného záznamu v aktuálně otevřeném editačním formuláři ihned po úspěšném provedení skriptu. Záznam je vyhledán v přidružené databázové tabulce ze všech záznamů, které mají ve sloupci zadaném parametrem „column“ uloženou hodnotu zadanou parametrem „id“. Pokud nebude dohledán žádný záznam odpovídající této podmínce, bude vytvořen nový záznam, do sloupce „column“ bude uložena hodnota „id“, a tento záznam bude následně v editačním formuláři otevřen.

 **Příklad:**

```
SCHEDULEENTRY (pid, 123)
```

void SCHEDULEFORM(int id)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce naplánuje otevření editačního formuláře zadaného parametrem „id“. Parametr „id“ je možné zjistit v administrátorském režimu v tooltipu, který se zobrazí po zastavení ukazatele myši nad odkazem „Upravit editační formulář“. Bude otevřen prázdný editační formulář za účelem vytvoření nového záznamu.

 **Příklad:**

```
SCHEDULEFORM(1)
SCHEDULEFORM(#form#)
```

void SCHEDULEFORM(string dbname)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce naplánuje otevření editačního formuláře zadaného parametrem „dbname“. Bude otevřen prázdný editační formulář za účelem vytvoření nového záznamu.

 **Příklad:**

```
SCHEDULEFORM(susers)
```

void SCHEDULEFORM(int id, int record)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce naplánuje otevření editačního formuláře zadaného parametrem „id“, a otevření záznamu s primárním klíčem zadaným parametrem „record“ ihned po úspěšném provedení skriptu. Parametr „id“ je možné zjistit v administrátorském režimu v tooltipu, který se zobrazí po zastavení ukazatele myši nad odkazem „Upravit editační formulář“.

 **Příklad:**

```
SCHEDULEFORM(1, 1)
```


void SCHEDULEFORM(string dbname, int record)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce naplánuje otevření editačního formuláře zadaného parametrem „dbname“, a otevření záznamu s primárním klíčem zadaným parametrem „record“ ihned po úspěšném provedení skriptu.

 **Příklad:**

```
SCHEDULEFORM(susers, 1)
```

void SCHEDULEFORM(int id, int record, bool back)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce naplánuje otevření editačního formuláře zadaného parametrem „id“, a otevření záznamu s primárním klíčem zadaným parametrem „record“ ihned po úspěšném provedení skriptu. Parametr „id“ je možné zjistit v administrátorském režimu v tooltipu, který se zobrazí po zastavení ukazatele myši nad odkazem „Upravit editační formulář“. Parametr „back“ definuje, zda se okno prohlížeče přesměruje na hlavní stránku (nahližecí stránku/portlet) při hodnotě „true“ nebo pouze zpět do nadřazeného formuláře při hodnotě „false“.

 **Příklad:**

```
SCHEDULEFORM(1, 1, true)
```

void SCHEDULEFORM(string dbname, int record, bool back)

 **Použití:** Skript (na místě, kde se očekává zadání komentáře)

 **Popis:** Funkce naplánuje otevření editačního formuláře zadaného parametrem „dbname“, a otevření záznamu s primárním klíčem zadaným parametrem „record“ ihned po úspěšném provedení skriptu. Parametr „back“ definuje, zda se okno prohlížeče přesměruje na hlavní stránku (nahližecí stránku/portlet) při hodnotě „true“ nebo pouze zpět do nadřazeného formuláře při hodnotě „false“.

 **Příklad:**

```
SCHEDULEFORM(susers, 1, false)
```

void SCHEDULEMAILMESSAGE(string to, string cc, string bcc, string subject, string body, string attachments, bool save [, string c0, string v0 [, string c1, string v1...]])

Použití: Skript (na místě, kde se očekává zadání komentáře)

Popis: Funkce naplánuje otevření stránky pro zadávání nové e-mailové zprávy ihned po úspěšném provedení skriptu. Funkce na stránce nové e-mailové zprávy vyplní zadávací pole:

- „Komu“ parametrem „to“,
- „Kopie“ parametrem „cc“,
- „Skrytá“ parametrem „bcc“,
- „Předmět“ parametrem „subject“,
- „Zpráva“ parametrem „body“.

Dále ke zprávě přiloží souborové přílohy zadané parametrem „attachments“, a zaškrtně zaškrťovací tlačítko „Uložit“, pokud je hodnota parametru „save“ rovna hodnotě „True“. „Attachments“ může být seznam ID nebo cest na disku oddělených středníkem. Parametry „c0“ resp. „v0“ značí tabulku odeslané pošty – identifikátor databázového ovládacího prvku, resp. hodnotu, která bude do databázového ovládacího prvku vložena, pokud uživatel před odesláním e-mailové zprávy zatrhne volbu „Uložit“. Tím zajistí uložení odeslané zprávy do databázové tabulky, která může tyto ovládací prvky obsahovat. Nastavení databázové tabulky, do které se odeslaná pošta ukládá, se provádí u ovládacího prvku „E-Mail“.

Příklad:

```
SCHEDULEMAILMESSAGE(info@netgenium.com, , , E-mailová zpráva, Vážený pane..., 0, true)
SCHEDULEMAILMESSAGE(info@netgenium.com, , , E-mailová zpráva, Vážený pane...,
1;#ng_priloha#;D:\priloha.txt, true)
```

void SCHEDULEPRINT(int button, int id)


Použití: Skript (na místě, kde se očekává zadání komentáře)


Popis: Funkce naplánuje tisk do tiskové šablony ihned po úspěšném provedení skriptu. Parametr „button“ značí ID tlačítka, které obsluhuje tisk dané šablony, a parametr „id“ značí ID záznamu, který se bude tisknout. Tisk je prováděn do souboru určeného tiskovou šablonou. Pokud je ve skriptu použito více volání funkcí „SCHEDULEPRINT“, je výstupem více vytištěných souborů zabalených do archivu „zip“, který je následně nabídnut ke stažení.

Příklad:

```
SCHEDULEPRINT(123, 1)
```

void SCHEDULEPRINT(int button, int id, bool joinxls)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce naplánuje tisk do tiskové šablony ihned po úspěšném provedení skriptu. Parametr „button“ značí ID tlačítka, které obsluhuje tisk dané šablony, a parametr „id“ značí ID záznamu, který se bude tisknout. Tisk je prováděn do souboru určeného tiskovou šablonou. Pokud je ve skriptu použito více volání funkcí „SCHEDULEPRINT“, je výstupem více vytištěných souborů zabalených do archivu „zip“, který je následně nabídnut ke stažení. Jsou-li všechny vytištěné soubory výsledkem tisku do tiskových šablon aplikace MS Excel, rozhoduje o výstupu funkce parametr „joinxls“. Pokud je hodnota parametru „joinxls“ nastavena na „true“, dojde ke spojení všech excelových souborů do jednoho souboru s příponou „xlsx“ s více listy.

 **Příklad:**

```
SCHEDULEPRINT(123, 1, true)
```

void SCHEDULEPRINTOFATTACHMENT(int id)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce naplánuje tisk souborové přílohy ihned po úspěšném provedení skriptu. Parametr „id“ značí ID souborové přílohy, která se bude tisknout. Pokud je ve skriptu použito více volání funkcí „SCHEDULEPRINTOFATTACHMENT“, je výstupem více vytištěných souborů zabalených do archivu „zip“, který je následně nabídnut ke stažení. Volání funkce lze také libovolně kombinovat s funkcí „SCHEDULEPRINT“.

 **Příklad:**

```
SCHEDULEPRINTOFATTACHMENT(1)
```

void SCHEDULEPRINTOFATTACHMENT(int id, string filename)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce naplánuje tisk souborové přílohy ihned po úspěšném provedení skriptu. Parametr „id“ značí ID souborové přílohy, která se bude tisknout. Tisk je prováděn do souboru pod názvem určeným parametrem „filename“. Přípona souboru zůstane přejmenováním zachována. Pokud je ve skriptu použito více volání funkcí „SCHEDULEPRINTOFATTACHMENT“, je výstupem více vytištěných souborů zabalených do archivu „zip“, který je následně nabídnut ke stažení. Volání funkce lze také libovolně kombinovat s funkcí „SCHEDULEPRINT“.

 **Příklad:**

```
SCHEDULEPRINTOFATTACHMENT(1, Příloha)
```

void SCHEDULEPRINTNAME(string filename)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce naplánuje název souboru tiskové sestavy zadaný parametrem „filename“. Použití funkce má význam pouze ve skriptech, které slouží k tisku do tiskové šablony.

 **Příklad:**

```
SCHEDULEPRINTNAME (Invoice - #ng_cislofv#)
```

void SCHEDULESAVE()


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce naplánuje uložení právě otevřeného záznamu v editačním formuláři ihned po úspěšném provedení skriptu.

 **Příklad:**

```
SCHEDULESAVE ()
```

void SCHEDULESAVE(string dbname)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce naplánuje uložení hodnoty ovládacího prvku právě otevřeného záznamu v editačním formuláři ihned po úspěšném provedení skriptu.

 **Příklad:**

```
SCHEDULESAVE (ng_textbox)
```

void SCHEDULESCRIPT(int script)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce naplánuje spuštění skriptu přiřazeného tlačítku v nahlížečí stránce zadaného parametrem „script“ ihned po úspěšném provedení skriptu. Parametr „script“ je možné zjistit v administrátorském režimu v tooltipu, který se zobrazí po zastavení ukazatele myši nad odkazem „Upravit skript“.

 **Příklad:**

```
SCHEDULESCRIPT (1)
```

void SCHEDULEURL(string url)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce naplánuje otevření webové stránky zadané parametrem „url“ ihned po úspěšném provedení skriptu.

 **Příklad:**

```
SCHEDULEURL (https://www.netgenium.com)
```

void SCHEDULEVIEWPAGE(int viewpage)

 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce naplánuje otevření nahlížečí stránky zadané parametrem „viewpage“ ihned po úspěšném provedení skriptu. Parametr „viewpage“ je možné zjistit v administrátorském režimu v tooltipu, který se zobrazí po zastavení ukazatele myši nad odkazem „Upravit nahlížečí stránku“.

 **Příklad:**

```
SCHEDULEVIEWPAGE (1)
```

double SIN(double number)


 **Použití:** HTML, Skript

 **Popis:** Funkce vrací sinus z hodnoty zadané parametrem „number“.

 **Příklad:**

```
SIN(-1.23)
// Vrací „-0,942488801931697“
SIN(1.23)
// Vrací „0,942488801931697“
SIN(#A#)
```

void SKIPDELETE()


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)

 **Popis:** Funkce zabrání smazání právě otevřeného záznamu v editačním formuláři.

 **Příklad:**

```
SKIPDELETE ()
```

void SKIPSAVE()


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce zabrání uložení právě otevřeného záznamu v editačním formuláři.

 **Příklad:**

```
SKIPSAVE ()
```

void SLEEP(int milliseconds)

 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce pozastaví běh skriptu na „milliseconds“ milisekund.

 **Příklad:**

```
SLEEP(1000)
```

int SPLIT(string value, string separator)

 **Použití:** Skript


 **Popis:** Funkce vrací počet prvků v textovém řetězci zadaném parametrem „value“ rozděleném podle hodnoty definované parametrem „separator“.

 **Příklad:**

```
SPLIT(A;B;C, ";") // Vrací „3“  
SPLIT(1;2;5;6;8, ";") // Vrací „5“
```

string SPLIT(string value, string separator, int index)

 **Použití:** Skript


 **Popis:** Funkce vrací prvek, který je na pozici zadané parametrem „index“ v textovém řetězci zadaném parametrem „value“ rozděleném podle hodnoty definované parametrem „separator“. Index používá číslování od nuly.

 **Příklad:**

```
SPLIT(A;B;C, ";", 0) // Vrací „A“  
SPLIT(1;2;5;6;8, ";", 3) // Vrací „6“
```

string SQL(string query)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací hodnotu z prvního řádku a prvního sloupce databázového dotazu zadaného parametrem „query“. Pokud jsou součástí databázového dotazu čárky, je nutné umístit celý dotaz mezi dvojité uvozovky.

 **Příklad:**

```
SQL(SELECT * FROM susers)
```

string SQL(string query, string separator)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací výsledek databázového dotazu zadaného parametrem „query“. Pokud jsou součástí databázového dotazu čárky, je nutné umístit celý dotaz mezi dvojité uvozovky. Hodnoty výsledku dotazu jsou odděleny oddělovačem zadaným parametrem „separator“. Oddělovač může být definován pro každý jednotlivý sloupec zvlášť jako seznam hodnot oddělených středníkem. V takovém případě musí být definováno tolik oddělovačů, kolik je počet sloupců – 1.

 **Příklad:**

```
SQL("SELECT id, loginname, lng FROM susers", " - ")
SQL("SELECT id, loginname, lng FROM susers", " - ; -- ")
```

string SQL(string query, string columnSeparator, string lineSeparator)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací výsledek databázového dotazu zadaného parametrem „query“. Pokud jsou součástí databázového dotazu čárky, je nutné umístit celý dotaz mezi dvojité uvozovky. Sloupce hodnot výsledku dotazu jsou odděleny parametrem „columnSeparator“, a jednotlivé řádky jsou odděleny parametrem „lineSeparator“. Oddělovač „columnSeparator“ může být definován pro každý jednotlivý sloupec zvlášť jako seznam hodnot oddělených středníkem. V takovém případě musí být definováno tolik oddělovačů, kolik je počet sloupců – 1.

 **Příklad:**

```
SQL("SELECT id, loginname, lng FROM susers", " - ", <br>)
SQL("SELECT id, loginname, lng FROM susers", " - ; -- ", <br>)
```


string SQLARRAY(string query)

 **Použití:** JavaScript

 **Popis:** Funkce vrací výsledek databázového dotazu zadaného parametrem „query“. Pokud jsou součástí databázového dotazu čárky, je nutné umístit celý dotaz mezi dvojité uvozovky. Databázový dotaz vrací pole polí ve formě javascriptu.

 **Příklad:**


```
SQLARRAY(SELECT * FROM susers)
```

 **Příklad výsledku:**

```
[  
  [1, 'True', 0, 0, 0, '', '', '', '', 'Administrator', 'pn1Xly6NKqI=', 'cs', '', ''],  
  [2, 'True', 0, 0, 0, '', '', '', '', 'Anonymous', 'pn1Xly6NKqI=', 'cs', '', '']  
]
```

string SQLQUERY(string control)

 **Použití:** Skript

 **Popis:** Funkce vrací SQL dotaz ovládacího prvku zadaného parametrem „control“.

 **Příklad:**

```
SQLQUERY(datagrid1)
```

string SQLTABLE(string query, string key1, [string key2, string key3...])

Použití: Výraz ve virtuálním sloupci statistického dotazu. Funkce „SQLTABLE“ se používá jako optimalizovaná varianta k funkci „SQL“, jejíž volání v každém řádku statistického dotazu vede k problémům s rychlostí načítání výsledného statistického dotazu – funkce „SQL“ je spuštěna tolikrát, kolik je v načteném dotazu řádků.

Popis: Funkce načte databázový dotaz zadaný parametrem „query“, který je definován pomocí seskupování dat klauzulí „GROUP BY“, a vrací hodnotu posledního sloupce z „query“ pro řádek definovaný klíčem „key1“, „key2“, atd... Výsledek dotazu je načten do virtuální tabulky v paměti, a z jednotlivých řádků tabulky je sestaven slovník typu klíč-hodnota. Počet sloupců tabulky, které definují unikátní klíč, je určen počtem použitých klíčů v parametrech „key1“, „key2“ atd. Sloupec, který definuje hodnotu, je vždy poslední sloupec z dotazu.

Příklad – neoptimalizovaná varianta pomocí funkce „SQL“:

```
Počet položek@SQL("SELECT COUNT(*) FROM ng_polozka WHERE pid = FORMATINTSQL(#c0#)")
```

Příklad – optimalizovaná varianta pomocí funkce „SQLTABLE“:

```
Počet položek@SQLTABLE("SELECT pid, COUNT(*) FROM ng_polozka GROUP BY pid", #c0#)
```

double SQUARE(double number)

Použití: HTML, Skript


Popis: Funkce vrací druhou odmocninu kladného čísla zadaného parametrem „number“.

Příklad:

```
SQUARE (-5.22)
// Vrací „Není číslo“
SQUARE (5.22)
// Vrací „2,28473193175917“
SQUARE (#A#)
```

int SUBSTRACTDATES(DateTime date1, DateTime date2)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací počet minut definovaný jako rozdíl mezi datem zadaným parametrem „date1“ a datem zadaným parametrem „date2“.

 **Příklad:**

```
SUBSTRACTDATES(#today# + 14, #today#)
// Vrací „20160“ (14 * 60 * 24)
```

double SUBSTRACTDATES2(DateTime date1, DateTime date2)

 **Použití:** HTML, JavaScript, Skript

 **Popis:** Funkce vrací počet vteřin definovaný jako rozdíl mezi datem zadaným parametrem „date1“ a datem zadaným parametrem „date2“.

 **Příklad:**

```
SUBSTRACTDATES2(#today# + 14, #today#)
// Vrací „1209600“ (14 * 60 * 24 * 60)
```

int SUBSTRACTINTEGERS(int value1, in value2)

 **Použití:** HTML, JavaScript, Skript

 **Popis:** Funkce vrací rozdíl dvou integerů zadaných parametrem „value1“ a „value2“.

 **Příklad:**

```
SUBSTRACTINTEGERS(10, 1)
// Vrací „9“
```

string SUBSTRING(string text, string startindex, int length)

Použití: HTML, JavaScript, Skript

Popis: Funkce vrací podřetězec z řetězce zadaného parametrem „text“ počínaje znakem na pozici uložené v parametru „startindex“ o délce uložené v parametru „length“. Je-li hodnota parametru „length“ rovna „-1“, vrací podřetězec počínaje znakem na pozici uložené v parametru „startindex“ až na konec řetězce.

Příklad:

```
SUBSTRING(Vážený pane Nováku, 12, 5)
// Vrací „Novák“
SUBSTRING(Vážený pane Nováku, 12, -1)
// Vrací „Nováku“
```

double SUM(object[] array)

Použití: Skript

Popis: Funkce vrací součet hodnot skriptové proměnné zadané parametrem „array“, je-li proměnná polem hodnot. Není-li proměnná polem hodnot, vrací funkce hodnotu proměnné.

Příklad:

```
SUM(#A#)
```

string[] StringArray(string text1; string text2; ...)




Použití: Skript

Popis: Funkce deklaruje skriptovou proměnnou typu pole hodnot „String“ z jednotlivých hodnot zadaných parametrem „text1“, „text2“, atd. Parametry funkce se oddělují pomocí středníku nebo tabulátoru. Volání funkce „StringArray“ musí dodržovat velká a malá písmena v názvu funkce.

Příklad:




```
StringArray(a;b;c)
StringArray(a#tab#b#tab#c)
```

void TABLE(string id)

-  **Použití:** Název ovládacího prvku HTML – tělo ovládacího prvku musí zůstat prázdné
-  **Popis:** Funkce vygeneruje HTML tag „<table id="{id}" cellspacing="0" cellpadding="0"><tr>“, resp. zahájení nové tabulky. ID tagu definuje parametr „id“.
-  **Příklad:**




```
TABLE (T1)
```

void /TABLE

-  **Použití:** Název ovládacího prvku HTML – tělo ovládacího prvku musí zůstat prázdné
-  **Popis:** Funkce vygeneruje HTML tag „</table>“, resp. zakončení předchozí tabulky, která byla vygenerována pomocí funkce „TABLE“.
-  **Příklad:**

```
/TABLE
```


double TAN(double number)

-  **Použití:** HTML, Skript
-  **Popis:** Funkce vrací tangens z hodnoty zadané parametrem „number“.
-  **Příklad:**

```
TAN(-0.56)
// Vrací „-0,626949535052698“
TAN(1)
// Vrací „1,557407724654902“
TAN(#A#)
```

void TD

 **Použití:** *Název ovládacího prvku HTML – tělo ovládacího prvku musí zůstat prázdné*


 **Popis:** *Funkce vygeneruje HTML tag „<td>“, resp. zahájení nové buňky tabulky.*

 **Příklad:**

```
TD
```

void TEFUIG(string message, string group1, string group2, ...)

 **Použití:** *Skript*

 **Popis:** *Funkce vyvolá přerušení s chybovou zprávou zadanou parametrem „message“ všem uživatelům, kteří jsou členy skupin „group1“, „group2“..., a ukončí provádění skriptu. Počet parametrů se seznamem skupin je libovolný.*

 **Příklad:**

```
TEFUIG(Nemáte oprávnění k provedení této činnosti., 6, 21)  
TEFUIG("Zpráva, která obsahuje čárku.", 6, 21)
```

string TEMPPASSWORD()

 **Použití:** *HTML, JavaScript, Skript*


 **Popis:** *Funkce vygeneruje náhodné číslo, které se používá jako dočasné heslo.*

 **Příklad:**

```
TEMPPASSWORD ()
```

void TESUIG(string message, string group1, string group2, ...)

 **Použití:** Skript

 **Popis:** Funkce vyvolá přerušení s chybovou zprávou zadanou parametrem „message“ všem uživatelům, kteří nejsou členy skupin „group1“, „group2“..., a ukončí provádění skriptu. Počet parametrů se seznamem skupin je libovolný.

 **Příklad:**

```
TESUIG(Nemáte oprávnění k provedení této činnosti., 9, 3)
TESUIG("Zpráva, která obsahuje čárku.", 9, 3)
```

void THROWEXCEPTION(string exception)

 **Použití:** HTML, Skript


 **Popis:** Funkce vyvolá přerušení s chybovou zprávou zadanou parametrem „exception“.

 **Příklad:**

```
THROWEXCEPTION(abc)
```

int TIME2INT(string time)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací počet minut konvertovaný z textového řetězce – času – zadaného parametrem „time“ ve formátu „HH:mm“.

 **Příklad:**

```
TIME2INT(08:00)
// Vrací „480“
TIME2INT(00:00)
// Vrací „0“
```

int TIME2MINUTES(string time)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací počet minut konvertovaný z textového řetězce – času – zadaného parametrem „time“ ve formátu „HH:mm“.

 **Příklad:**

```
TIME2MINUTES(08:00)
// Vrací „480“
TIME2MINUTES(00:00)
// Vrací „0“
```

double TIME2SECONDS(string time)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací počet vteřin konvertovaný z textového řetězce – času – zadaného parametrem „time“ ve formátu „HH:mm:ss“.

 **Příklad:**

```
TIME2SECONDS(00:08:00)
// Vrací „480“
TIME2SECONDS(00:00:00)
// Vrací „0“
```

void TR

 **Použití:** Název ovládacího prvku HTML – tělo ovládacího prvku musí zůstat prázdné


 **Popis:** Funkce vygeneruje HTML tag „<tr>“, resp. zahájení nové řádky tabulky.

 **Příklad:**

```
TR
```


string TRANSLATE(string text, string language)

 **Použití:** JavaScript


 **Popis:** Funkce vrací jazykový překlad vícejazyčného termínu zadaného parametrem „text“. Text je přeložen dle jazykového nastavení definovaného parametrem „language“.

 **Příklad:**

```
TRANSLATE(Schvalování bylo stornováno#en:Approval has been canceled,  
#language#)
```

string TRIM(string text)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací textový řetězec zadaný parametrem „text“, ve kterém byly na začátku a na konci odstraněny mezery a netisknutelné znaky.

 **Příklad:**

```
TRIM( Dobrý den )  
// Vrací „Dobrý den“
```

void UNLOCKPASSWORDS()


 **Použití:** Skript


 **Popis:** Funkce povolí nastavení nového hesla v editačním formuláři uživatele.

 **Příklad:**

```
UNLOCKPASSWORDS ()
```

void UPDATEAUTOSUM(int id, int form)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce aktualizuje všechny sumy vnořených záznamů, které jsou součástí záznamu zadaného parametrem „id“ v editačním formuláři zadaném parametrem „form“.

 **Příklad:**

```
UPDATEAUTOSUM(1, 1)
UPDATEAUTOSUM(#id#, #form#)
```

void UPDATEAUTOSUM(string id, int form)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce aktualizuje všechny sumy vnořených záznamů, které jsou součástí záznamů zadaných parametrem „id“ (pole hodnot oddělených středníkem) v editačním formuláři zadaném parametrem „form“.

 **Příklad:**

```
UPDATEAUTOSUM(1;2;3, 1)
```

void UPDATEAUTOSUM(int id, string dbname)


 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce aktualizuje všechny sumy vnořených záznamů, které jsou součástí záznamu zadaného parametrem „id“ v editačním formuláři zadaném parametrem „dbname“.

 **Příklad:**

```
UPDATEAUTOSUM(1, susers)
```

void UPDATEAUTOSUM(string id, string dbname)

 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce aktualizuje všechny sumy vnořených záznamů, které jsou součástí záznamů zadaných parametrem „id“ (pole hodnot oddělených středníkem) v editačním formuláři zadaném parametrem „dbname“.

 **Příklad:**

```
UPDATEAUTOSUM(1;2;3, susers)
```

string UPPER(string text)

 **Použití:** HTML, JavaScript, Skript

 **Popis:** Funkce vrací textový řetězec zadaný parametrem „text“, ve kterém jsou všechny znaky převedeny na velká písmena.

 **Příklad:**

```
UPPER(abc)  
// Vrací „ABC“
```

string URLDECODE(string text)

 **Použití:** Skript


 **Popis:** Funkce dekóduje všechny URL hexadecimální znaky v textovém řetězci zadaném parametrem „text“.

 **Příklad:**

```
URLDECODE(https%3a%2f%2fwww.netgenium.com)  
// Vrací „https://www.netgenium.com“
```

string URLENCODE(string text)

 **Použití:** Skript


 **Popis:** Funkce zakóduje všechny URL nevalidní znaky (např. mezera) v textovém řetězci zadaném parametrem „text“ za odpovídající sekvenci znaků v hexadecimální podobě.

 **Příklad:**

```
URLENCODE(https://www.netgenium.com)
// Vrací „https%3a%2f%2fwww.netgenium.com“
```

string USERDETAILS(string columnname)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací hodnotu uloženou v databázové tabulce uživatelů ve sloupci zadaném parametrem „columnname“ na řádce, který se vztahuje k aktuálně přihlášenému uživateli.

 **Příklad:**

```
USERDETAILS(id)
USERDETAILS(loginname)
```

string USERGROUP(string loginname)

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací seznam všech názvů uživatelských skupin oddělených středníkem, v nichž je uživatel zadaný parametrem „loginname“ členem.

 **Příklad:**

```
USERGROUP(Administrator)
USERGROUP(#loginname#)
```

string USERINGROUP(string group1 [... string group2]))

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací hodnotu „x“, je-li aktuálně přihlášený uživatel členem uživatelské skupiny zadané parametrem „group1“, popřípadě členem více skupin zadaných dalšími parametry. Funkce vrací hodnotu „“ pokud členem této uživatelské skupiny není. „Group“ může být ID uživatelské skupiny nebo její jméno.

 **Příklad:**

```
USERINGROUP(1)
USERINGROUP(Administrators)
USERINGROUP(1, 2)
USERINGROUP(Administrators, Users)
// Vrací „“ nebo „x“
```

string USERSINGROUP(int group1 [... int group2])

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací seznam všech ID uživatelů oddělených středníkem, kteří jsou členy skupiny se zadaným ID v parametru „group1“, popřípadě více skupin zadaných dalšími parametry.

 **Příklad:**

```
USERSINGROUP(1)
USERSINGROUP(1, 2)
// Vrací například „1“, „1;2“ atp.
```

string USERSINGROUP(string group1 [... string group2])

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací seznam všech ID uživatelů oddělených středníkem, kteří jsou členy skupiny se zadaným ID v parametru „group1“, popřípadě více skupin zadaných dalšími parametry.

 **Příklad:**

```
USERSINGROUP(Administrators)
USERSINGROUP(Administrators, Users)
// Vrací například „1“, „1;2“ atp.
```

string USERSINGROUP2(int group1 [,... int group2])

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací seznam všech přihlašovacích jmen uživatelů oddělených středníkem, kteří jsou členy skupiny se zadaným ID v parametru „group1“, popřípadě více skupin zadaných dalšími parametry.

 **Příklad:**

```
USERSINGROUP2(1)
USERSINGROUP2(1, 2)
// Vrací například „Administrator“, „Administrator;Anonymous“ atp.
```

string USERSINGROUP2(string group1 [,... string group2])

 **Použití:** HTML, JavaScript, Skript


 **Popis:** Funkce vrací seznam všech přihlašovacích jmen uživatelů oddělených středníkem, kteří jsou členy skupiny se zadaným ID v parametru „group1“, popřípadě více skupin zadaných dalšími parametry.

 **Příklad:**

```
USERSINGROUP2(Administrators)
USERSINGROUP2(Administrators, Users)
// Vrací například „Administrator“, „Administrator;Anonymous“ atp.
```

string VALIDEMAIL(string email)



 **Použití:** HTML, JavaScript, Skript

 **Popis:** Funkce otestuje správnost formátu e-mailové adresy zadané parametrem „email“. Pokud ano, funkce vrací „x“. Pokud e-mailová adresa není ve správném formátu, funkce vrací prázdný řetězec.

 **Příklad:**

```
VALIDEMAIL(info@netgenium.com)
// Vrací „x“
```



void WEBFORM

-  **Použití:** *Název ovládacího prvku HTML – tělo ovládacího prvku musí zůstat prázdné*
-  **Popis:** *Funkce zahájí blok ovládacích prvků, které nepoužívají barevné pozadí v levém sloupci s názvem ovládacího prvku, a které používají reverzní design – ovládací prvky nemají rámeček, mají nastavené barevné pozadí, a při focusu zobrazují barevnou lištu pod ovládacím prvkem.
Zahájený blok, který není ukončen pomocí funkce „/WEBFROM“, ukotví tlačítka „Uložit“, „Smazat“ a „Zpět“. Tyto tlačítka pak nebudou plovoucí, a budou se zobrazovat vždy až na konci editačního formuláře.*

 **Příklad:**

```
WEBFORM
```



void /WEBFORM

-  **Použití:** *Název ovládacího prvku HTML – tělo ovládacího prvku musí zůstat prázdné*
-  **Popis:** *Funkce zakončí blok ovládacích prvků, které nepoužívají barevné pozadí v levém sloupci s názvem ovládacího prvku, a které používají reverzní design – ovládací prvky nemají rámeček, mají nastavené barevné pozadí, a při focusu zobrazují barevnou lištu pod ovládacím prvkem.*

 **Příklad:**

```
/WEBFORM
```


void WRITEMESSAGE(string message)


-  **Použití:** *Skript (na místě, kde se očekává zadání komentáře)*
-  **Popis:** *Funkce vypíše zprávu zadanou parametrem „message“ ihned po úspěšném provedení skriptu. Funkci nelze využít ve skriptu „OnBeforeSave“ a „OnAfterSave“.*

 **Příklad:**

```
WRITEMESSAGE(Provádění skriptu bylo úspěšně dokončeno.)  
WRITEMESSAGE("Zpráva, která obsahuje čárku.")
```

void WRITEMESSAGE(string message, bool dialog)

 **Použití:** Skript (na místě, kde se očekává zadání komentáře)


 **Popis:** Funkce vypíše zprávu zadanou parametrem „message“ ihned po úspěšném provedení skriptu, a zobrazí ji také v dialogovém okně. Funkci nelze využít ve skriptu „OnBeforeSave“ a „OnAfterSave“.

 **Příklad:**

```
WRITEMESSAGE(Provádění skriptu bylo úspěšně dokončeno., true)
WRITEMESSAGE("Zpráva, která obsahuje čárku.", true)
```

int YEAR(DateTime date)

 **Použití:** HTML, JavaScript, Skript

 **Popis:** Funkce vrací číslo roku z data zadaného parametrem „date“.

 **Příklad:**

```
YEAR(15.2.2013)
// Vrací „2013“
YEAR(#now#)
```