

## 12. Matlab - Funkcie

Blaho Michal · MATLAB/Comsol

14.08.2009



V predchádzajúcej časti nášho seriálu sme si povedali čo je M-súbor, aké typy M-súborov môžeme vytvoriť a ukázali sme si ako ho vytvoriť a skontrolovať pomocou Editoru Matlabu. Dnes sa budeme venovať druhému typu M-súborov a to funkciám, ktoré ponúkajú viac možností ako samostatné skripty.

Ako sme si už povedali v minulej časti, funkcie sú M-súbory, ktoré môžu prijímať vstupné argumenty a vracajú výstupné argumenty. Funkcie pracujú s vlastným Workspace, ktorý je iný ako používa príkazový riadok. Meno M-súboru by sa malo zhodovať s menom funkcie, lebo toto meno sa stáva novým príkazom Matlabu. Treba si dávať pozor, aby sa meno nezačínalo na číslo a iné nepovolené znaky, prípadne sa nezhodovalo s menom inej funkcie v Matlabe. M-súbor, ktorý obsahuje funkciu má nasledovnú štruktúru:

```
function [A1,A2] = rozdel(A,N)
%Funkcia rozdeli vektor na dva
%rozdel(A,N) vracia dva vektory, ktore vzniknu
%rozelenim vektora A na pozici N
    A1=A(1:N);
    A2=A(N+1:length(A))
end
```

Prvý riadok M-súboru hovorí Matlabu, že obsahuje funkciu. Začína sa kľúčovým slovom **function**, nasledujú výstupné argumenty, meno funkcie a vstupné argumenty. Tento riadok nemusí byť prvý, ak sa pred ním nachádzajú nevykonateľné riadky - komentáre. Výstupné argumenty sa zadávajú v hranatých zátvorkách, pri jednom výstupnom argumente zátvorky netreba písať. Ak by funkcia nevracala žiadne hodnoty, stačí túto časť vynechať alebo napísať prázdne hranaté zátvorky. Meno funkcie sa musí začínať písmenom, nesmie obsahovať nepovolené znaky, prekročiť povolenú dĺžku a nesmie sa zhodovať s nejakým kľúčovým slovom Matlabu.

Prvý komentár za definíciou funkcie sa nazýva H1 riadok. Mal by byť čo najjednoduchší a obsahovať základný popis funkcie. Vypíše sa pri hľadaní príkazu príkazom **lookfor**. Ďalšie komentáre za H1 riadkom slúžia ako pomocný text (Help text), ktoré sa vypíšu pri zadaní príkazu **help menofunkcie**. Vypíšu sa však iba

komentáre po prvý vykonateľný príkaz. Zvyšok sa už vyhodnocuje ako telo funkcie:

```
>>lookfor rozdel
>>help rozdel
```

Samostatné telo funkcie pozostáva z bežných príkazov Matlabu ako sú výpočty, priradenia (napríklad výstupným argumentom hodnoty) a komentárov. Kľúčové slovo **end** funkcia obsahovať nemusí, ale je potrebná pri vnorených funkciách. Na rozdiel od iných programovacích jazykov, môžu funkcie v Matlabe prijímať neznámy počet argumentov. Ak nie je priradená žiadna výstupná premenná, výsledok je uložený do premennej **ans**. Počet vstupov a výstupov sa dá zistiť pomocou premenných **nargin** a **nargout**.

## Typy funkcií v Matlabe

Matlab poskytuje niekoľko typov funkcií:

- anonymné funkcie - funkcia v príkazovom riadku
- primárna funkcia - prvá funkcia v M-súbore
- subfunkcie - funkcie za primárnou funkciou
- vnorené funkcie - funkcie v ďalšej funkcii
- súkromné funkcie - funkcie pracovného adresára
- preťažené funkcie - funkcie s rovnakým menom a rôznymi argumentmi

## Anonymné funkcie

Anonymná funkcia je zjednodušená forma funkcie v Matlabe, ktorá nepotrebuje M-súbor. Pozostáva z jedného výrazu v Matlabe a rôzneho počtu vstupných alebo výstupných argumentov. Môže byť zadefinovaná v príkazovom riadku, ďalšej funkcii alebo skripte. Syntax anonymnej funkcie je nasledovný:

```
f = @(argumenty) výraz
```

Samozrejme môžeme použiť niekoľko vstupných argumentov alebo žiadne. Využitie anonymnej funkcie si môžeme ukázať na nasledujúcom príklade:

```
>>sqr = @(x) x.^2;
>>a = sqr(5)
```

## Primárna funkcie

Každá iná funkcia ako anonymná musí byť definovaná v M-súbore. Prvá funkcia v M-súbore sa nazýva primárna. Za primárnou funkciou môže nasledovať niekoľko subfunkcií, ktoré môže primárna funkcia volať. Vo väčšine prípadov bude však M-súbor obsahovať iba primárnu funkciu.

## Subfunkcie

M-súbor môže obsahovať viac ako jednu funkciu. Ďalšie funkcie okrem primárnej sa potom nazývajú subfunkcie (prípadne podfunkcie). Subfunkcie sú viditeľné iba vrámci M-súbora, teda ich môže volať iba primárna funkcia alebo iné subfunkcie v M-súbore.

Ak by sme chceli zavolať popis funkcie (help) mysubfun v súbore funkcie myfun, museli by sme použiť nasledujúcu konštrukciu:

```
>>help myfun>mysubfun
```

### Vnorené funkcie

Funkcie sa môžu definovať aj v rámci tela inej funkcie. Vnorená funkcia obsahuje všetky komponenty ako obyčajná funkcia a nachádza sa v tele inej funkcie. Vnorené funkcie sa definujú nasledovne:

```
function x = A(p1, p2)
    ...
    function y = B(p3)
        ...
    end
    ...
end
```

Samozrejme vnorené funkcie môžu byť aj v rámci inej vnorenej funkcie. Volanie vnorených funkcií má niekoľko pravidiel. Vnorenú funkciu môžeme zavolať z vyššej úrovne ako je vnorená funkcia, z inej vnorenej funkcie na rovnakej úrovni alebo z vnorenej funkcie na nižšej úrovni.

### Súkromné funkcie

Súkromné funkcie sú špeciálny typ funkcie. Sú uložené v adresároch so špeciálnym názvom **private**. Sú viditeľné iba pre funkcie v rodičovskom adresári. Pretože sú schované môžu sa nazývať ako funkcie v iných adresároch. Matlab najskôr hľadá súkromné funkcie pred štandardnými funkciami. Pomocníka ku súkromnej získame nasledovným príkazom:

```
>>help private/myprivfun
```

### Preťažené funkcie

Preťažovanie funkcií funguje ako v iných programovacích jazykoch. Preťažované funkcie slúžia na volanie rovnakých funkcií s rôznym počtom alebo typom argumentov. Matlab potom dokáže rozlíšiť, ktorú funkciu má použiť. Dajú sa vytvoriť v jednom M-súbore podobne ako subfunkcie funkcie s rovnakým menom, ale inými argumentami.

### Globálne premenné

Globálne premenné sa využívajú keď chceme, aby viaceré funkcie pristupovali k jednej premennej, prípadne aby mali prístup k premennej vo Workspace príkazového riadka. Premenná sa musí deklarovať slovom **global**. Globálne premenné sa väčšinou píšu veľkými písmenami na lepšie rozoznanie. Osobne neodporúčam používanie globálnych premenných, lebo môže dôjsť k prepísaniu ich hodnôt a chyby spôsobené zlým použitím globálnych premenných sa hľadajú ťažšie. Alternatíva ku globálnej premennej je predať hodnotu ako argument funkcie a jej novú hodnotu vrátiť ako návratovú

hodnotu funkcie. Príklad využitia globálnej premennej si môžeme ukázať nasledovne:

```
function h = falling(t)
    global GRAVITY
    h = 1/2*GRAVITY*t.^2
end
```

```
>>global GRAVITY
>>GRAVITY = 32;
>>y = falling(10);
```

### Predávanie hodnôt funkciám

Funkcie, ktoré majú ako argumenty reťazce, nepotrebujú dávať argumenty do zátvoriek a úvodzoviek. Nasledujúce dva príkazy sa vykonajú rovnako:

```
>>foo a b c
>>foo('a', 'b', 'c')
```

Pri takomto zápise však nemôžu funkcie vracaf návratové hodnoty. V takom prípade treba použiť klasický zápis volania funkcie. Pri volaní funkcie ako v prvom prípade s inými dátovými typmi ako znakové reťazce Matlab vyhlási chybu.

### Funkcia eval

Funkcia **eval** pracuje so znakovými reťazcami. Interpreter Matlabu vykoná text v príkaze eval ako samostatný príkaz. Napríklad chceme vytvoriť 5 nulových matic s menami A1 až A5. Bez funkcie **eval** by sme to museli robiť priradením do premenných osobitne, pomocou funkcie **eval** nám stačí cyklus for:

```
for k = 1:5
    s = ['A' int2str(k) '=zeros(3)'];
    eval(s)
end
```

Viac o funkciách a ich možnostiach nájdete v užívateľskej príručke Matlabu. Príkazy na riadenie behu programu (cykly, podmienky a iné) budú obsahom nasledujúcej časti.