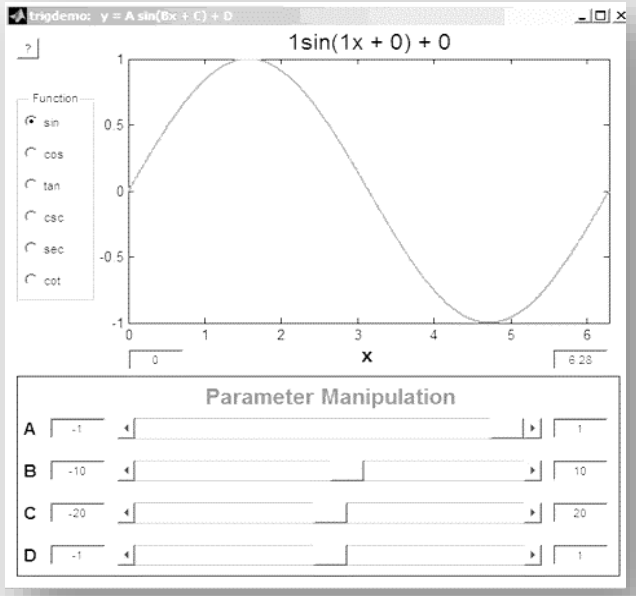
# Počítačové aplikácie

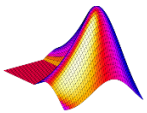
## Pr.8 – Tvorba funkcií




1. roč. AES, LS 2018/19

# Náplň

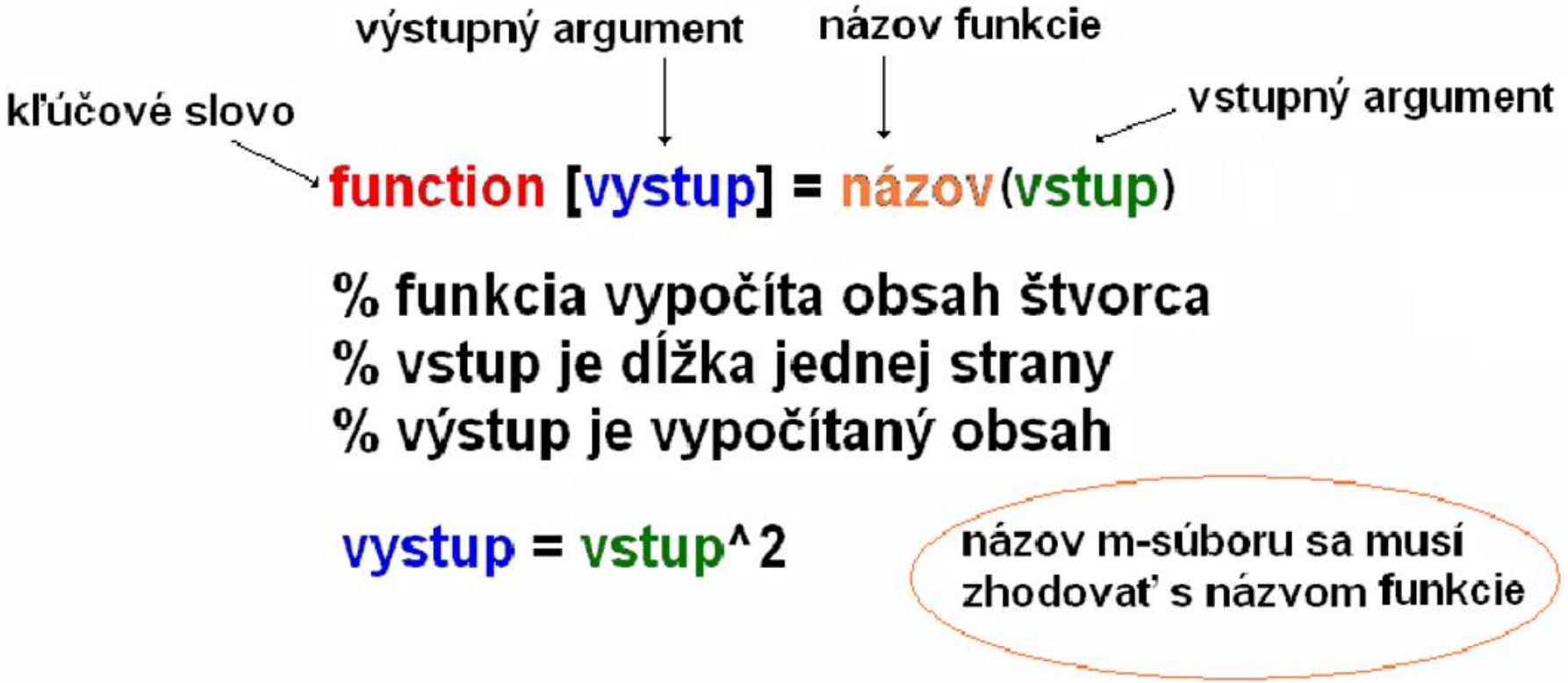
1. Tvorba funkcií
2. Funkcie bez vstupných argumentov
3. Globálne  a lokálne premenné
4. F. s jedným vstupným argumentom
5. F. s dvoma alebo viacerými vstupnými argumentami
6. F. s jedným výstupným argumentum
7. F. s dvoma alebo viacerými výstupnými argumentami
8. F. s výstupnými aj vstupnými argumentami
9. Volanie funkcie v rámci inej funkcie



# 1. Tvorba funkcií

Funkcie sú m-súbory, ktoré programátorovi ponúkajú väčšie možnosti využitia – môžu obsahovať (ale nemusia) vstupné a výstupné argumenty, čo umožňuje vytvárať zložitejšia aplikácie.

## syntax funkcie vo všeobecnosti



# 2. Funkcie bez vstupných argumentov

Funkcie bez vstupných argumentov sa správajú ako skript.

**Rozdiel** je v dostupnosti premenných, pretože
– vo funkcii sú všetky premenné lokálnymi premennými
– a ich hodnoty sa neukladajú do Workspace.

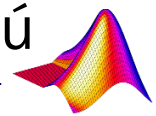
**Príklad:**

Funkcia s názvom parabola vždy nakreslí graf paraboly na danom intervale <-1, 1>
(nezadávame vstupné argumenty)

```
function parabola
    x = -1:0.02:1;
    y = x.^2;
    plot(x, y, 'ro-')
    whos
```

Premenné x, y sú tu lokálnymi premennými (neukladajú sa do Workspace)

Inštrukcia whos je tu použitá len pre zistenie obsahu pamäti – na demonštrovanie toho, aké lokálne veličiny sa tam nachádzajú

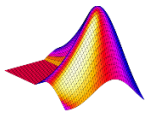


# 3. Globálne  a lokálne premenné

Ak chceme, aby premenné používané vo funkcii boli prístupné aj iným funkciám, potom ich musíme deklarovať ako globálne pomocou príkazu:

global premenna

Ak chceme, aby boli prístupné aj vo Workspace, potom musíme použiť rovnaký príkaz v príkazovom riadku okna Command Window



# Globálne a lokálne premenné

Obsah globálnych premenných je prístupný všade tam, kde sa vopred deklarujú ako global.

## Príklad:

zostaviť funkciu generovanie vektora náhodných celočíselných čísiel sinus na intervale <-5; 5>.

```
function generuj
global a
a = round(rand(1)*10-5)
```

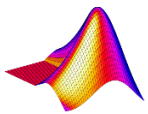
## Použitie:

Premennú a môžeme používať aj v inej funkcii:

```
function nasob
global a
global b
b = 10*a
```

Ak chceme premenné a, b používať v Command Window, tak ich takisto najprv musíme deklarovať ako **globálne** v príkazovom riadku

```
>> global a
>> global b
```



# 4. Funkcie s jedným vstupným argumentom

syntax: function nazov (vstup)

## Príklad:

zostaviť funkciu pre kreslenie grafu goniometrickej funkcie **sinus** na intervale zadanom parametrom **vstup**.

```
function graf_sin(vstup)
y = sin(vstup);
plot(vstup,y,'g*-')
```

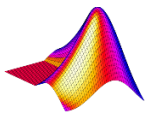
Vstupné parametre sa píšu do okrúhlych zátvoriek!

## Použitie:

Command Window

```
>> x = -2*pi:2*pi/100:2*pi;
>> graf_sin(x)

>> y = linspace(-1,1);
>> graf_sin(y)
```



# 5. Funkcie s dvoma alebo viacerými vstupnými argumentami

syntax: function nazov (vstup1, vstup2, ...)

## Príklad:

zostaviť funkciu pre kreslenie grafu funkcie sinus na danom intervale parametrom a s rôznym typom, prípadne aj farbou čiary.

```
function graf_sinus(interval, farba)
y = sin(interval);
plot(interval, y, farba)
```

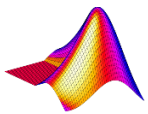
Pri volaní funkcie s daným počtom argumentov je potrebné pri volaní funkcie dodržať tento počet

## Použitie:

**Command Window**

```
>> y = linspace(-1,1);
>> graf_sinus(y,'r')

>> z = -pi:pi/50:pi
>> color ='b'
>> graf_sinus(z, color)
```



# 6. Funkcie s jedným výstupným argumentom

syntax: function vystup = nazov

## Príklad:

Zostavte funkciu pre generovanie vektora náhodných čísel z intervalu <-10; 10>.

```
function cislo = generuj_n
cislo = round(rand(10,1)*20-10);
```

**Názov výstupného argumentu** sa musí nachádzať v tele funkcie ako premenná, ktorej sa priradí vypočítaná hodnota.

Hodnota výstupného argumenta je prístupný vo Workspace

## Použitie:

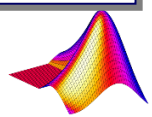
```
>> j =
generuj_n
j =
     3
     6
     9
     5
    -6
    -1
     9
     9
    -1
     8
```

**Command Window**

```
>> generuj_n
ans =
    10
    -5
     3
     0
     8
     6
     0
    -9
     7
    -1
```

Funkciu môžeme volať ako skript.

Výstup sa priradí premennej ans.



# 7. Funkcie s dvoma alebo viacerými výstupnými argumentami

syntax: function [vyst1,vyst2,..] = názov

## Príklad:

```
function [cislo1,cislo2] = generuj_c
cislo1 = magic(3);
cislo2 = round(rand(10,1)*20-10);
```

Výstupné parametre sa píšu do hranatých zátvoriek.

**Výstupné parametre** sa musia nachádzať v tele funkcie. Môžeme deklarovať ľubovoľný počet výstupných parametrov, ktoré sú prístupné vo Workspace.

## Použitie:

```
>> [A, a] = generuj_c
A =
     8     1     6
     3     5     7
     4     9     2
a =
   -10
     5
    -1
     9
    -1
    -2
     7
     1
    -6
     3
```

**Command Window**

```
>> generuj_c
ans =
    -9
    -3
     6
   -10
    -7
    -6
    -6
     2
    -5
    -6
```

**Ak pri volaní funkcie neuvedieme výstupné parametre, potom sa len jeden výstup priradí premennej ans.**

# 8. Funkcie s výstupnými aj vstupnými argumentami

syntax: function [vystupy] = názov (vstupy)

## Príklad:

Zostaviť funkciu pre výpočet priemernej hodnoty prvkov vektora.

```
function priem = priemer(vektor)
[r,s] = size(vektor);
if (r==1 & s~=1)|(r~=1 & s==1)
    priem=sum(vektor)/length(vektor)
else
    error('Vstup musí byť vektor')
end
```

Vstupom do funkcie pre výpočet priemeru musí byť vektor, ktorý obsahuje minimálne dve čísla.

## Použitie:

Command Window

```
>>  y = 1:20
>>  priemer(y)
      ans = 10.5

>>  x = priemer(y)
      x = 10.5

>>  z = 5
>>  priemer(z)
    ??? Error using ==> priemer
    Vstup musí byť vektor

>>  d = ones(2)
>>  priemer(d)
    ??? Error using ==> priemer
    Vstup musí byť vektor
```

# Funkcie s výstupnými aj vstupnými argumentami

## Príklad:

Zostaviť funkciu pre výpočet derivácie, integrálu a hodnôt prvkov vektora na danom intervale (a pritom súčasne nakresliť graf priebehu polynómu špecifikovaným typom čiary!

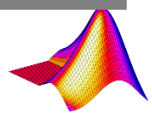
```
function [der,int,hod] = polynom(p,interval)
der = polyder(p);
int = polyint(p);
hod = polyval(p,interval);
plot(interval, hod, 'r*-')
```

## Použitie:

**Command Window**

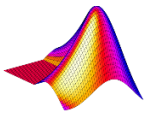
```
>> k = [1 5 8]
>> t = 0:0.1:10
>> [x,y,z] = polynom(k,t)


>> [d,i,h] = polynom([1 5 8],0:0.1:10)
```



# 9. Prehľad syntaxe preberaných funkcií

| Funkcia | Syntax |
|---|---|
| bez vstupných argumentov | function názov |
| s jedným vstupným argumentom | function názov (vstup) |
| s dvoma alebo viacerými vstupnými argumentami | function názov (vstup1, vstup2, ...) |
| s jedným výstupným argumentom | function vystup = názov |
| s dvoma alebo viacerými výstupnými argumentami | function [vyst1,vyst2,...] = názov |
| s výstupnými aj vstupnými argumentami | function [vystupy] = názov (vstupy) |



# 8. Volanie funkcie v rámci inej funkcie

## Príklad:

Z jednej funkcie pre vykonanie inštrukcií zavoláme druhú funkciu, z ktorej po výpočte volaných parametrov sa vrátime do prvej funkcie.

```
function [vysledok] = podiel
a = input('Zadaj prve cislo: ')
b = input('Zadaj druhe cislo :')
if b ~= 0
    vysledok=delenie(a,b);
else
    error('Druhe cislo = 0')
end
```

```
function [vysl] = delenie(c1,c2)
vysl = c1/c2;
```

## Použitie:

Command Window

```
>> d = podiel
Zadaj prve cislo: 5
Zadaj druhe cislo 1
d =
     5
```

# Linky, odporúčaná literatúra

1. Hans-Petter Halvorsen: MATLAB Part I: Introduction to MATLAB, 2016, s. 37 – 40 (literatúra označená na portáli ako **x0**)
2. Learning MATLAB Language,
   https://riptuzorial.com/Download/matlab-language.pdf
3. MATLAB - Functions,
   https://www.tutorialspoint.com/matlab/matlab_functions.htm
4. Learning MATLAB language
   https://riptutorial.com/Download/matlab-language.pdf
5. Introduction to Computer Programming
   http://www.shahmoradi.org/ICP2017F/