

# 1 Úvod do programování v jazyce SciLab

## 1.1 Úvodní poznámky

1. Všechny proměnné jsou matice. Skalár je  $a(1,1)$ . Vektor v první řádce je  $a(1,5)$ , v prvním sloupci  $a(5,1)$ . Celý sloupec je  $a(:,1)$ , řádek  $a(2,:)$ . Dvojtečka znamená „všechno“.
2. Středník za zadanou proměnnou  $a=5$ ; způsobí, že po zadání není žádná odezva (tisk). Naopak po zadání proměnné s čárkou nebo bez ničeho, tj.  $a=5$ , nebo  $a=5$  se hodnota proměnné vypíše na obrazovku. Pozn.: musí být zavolán příkaz `mode(0)`.
3. `help „objekt“` zobrazí nápovědu k objektu.  
Ikona `?` zavolá interaktivní stránku HELP.
4. Komentář je text začínající `//`. Nehraje žádnou roli ve výpočtech programu.

## 1.2 Proměnné a operace

Následně jsou zmíněny základní typy proměnných, se kterými budeme pracovat. Jsou však jen malým zlomkem z existující množiny proměnných.

- **čísla (matice)**

*Definice:*

- skalár  $a=5$ ;
- řádkový vektor  $a=[3\ 5\ 1]$ ;
- sloupcový vektor  $a=[3; 5; 1]$ , což je totéž jako  $a=[3\ 5\ 1]'$
- matice  $a=[2\ 3\ 4; 8\ 7\ 6]$ ;
- příkaz  $a=5:8$  vytvoří vektor  $[5\ 6\ 7\ 8]$ ;  $5:2:13 = [5\ 7\ 9\ 11\ 13]$
- příkaz  $a=zeros(2,3)$  vytvoří matici  $2 \times 3$  z nul
- příkaz  $a=ones(2,3)$  vytvoří matici  $2 \times 3$  z jedniček
- transpozice se provede pomocí apostrofu `'`.
- matici  $b$  ( $3 \times 3$ ) lze vyskládat:  $b=[a; 2*a; 5*a]$ ;

*Operace:*

- násobení matic `*` dělení `/` umocňování `^` nebo `**` odmocnina `sqrt()`
- tečkové operace `.*` `./` `.^` se provádějí prvek po prvku
- v operaci `*` platí pravidla pro násobení matic
- operace  $a/b$  je násobení  $a$  inverzí  $b$  (lze také  $a*inv(b)$ )

- **text:**  $a='ahoj'$ . Texty lze spojovat do vektorů:  
je-li  $a='dobry'$ ;  $b='den'$  a  $c=a+b$ , pak  $c='dobry den'$ .  
Užitečná je konverze číslo  $\rightarrow$  řetězec:  $s=string(a)$ , kde  $s$  je string a  $a$  double.
- **logické proměnné** jejichž hodnotami je „true“ (při počítání 1) a „false“ (při počítání 0).  
Logické operace: `==` `~=` `<` `<=` `>` `>=` `&` (and) `|` (or) `~` (not)

## Příklady

Zadejte:

```
a=[1 2 3]  b=[8; 9]  c=[11 12 13; 21 22 23; 31 32 33];
```

Zkuste a výsledek zdůvodněte:

```
x1=a*a'  x2=a'*a  y=[[a;5*a] b]  c(2,:).*a  c(1,2:3)*b  
c(3,:).^c(1,:)  c(3,:)**2  d1=c(:)  dd=c'; d2=dd(:)  d2(3:2:7)
```

Zadejte:

```
u='prvni'  v='pokus'  x=%t (zadání hodnoty true)  y=5==5  z=5>5
```

Zkuste a výsledek zdůvodněte:

```
u+' '+v  x & y  x & z  x | y  x | z
```

### 1.3 Práce s proměnnými

- Příkazy `who_user()`; dává informaci o definovaných proměnných.
- `[m,n]=size(a)`, `m=size(a,1)`, `n=size(a,2)` dávají informaci o rozměrech matice `a`, resp. počtu řádků, resp. počtu sloupců. Místo 1 a 2 lze psát 'r' a 'c'.
- `n=length(a)` udává informaci o počtu všech prvků objektu.
- `n=max(size(a))` dá délku vektoru (větší z rozměrů `a`)
- `clc` maže konzoli (obrazovku)
- `clear` uvolňuje paměť
- `close(winsid())` maže všechny grafy (`close` maže poslední graf)

### 1.4 Programovací příkazy

- **Podmínka if**

```
if b>c,  
    a=5;  
else  
    a=0;  
end
```

Je-li splněna podmínka `b>c`, provede se příkaz `a=5`; jinak příkaz `a=0`;

#### Příklad

```
// Určete c = jako větší z čísel a, b  
a=rand(1,1,'n'); b=rand(1,1,'n');  
if a>b, c=a;  
else c=b;  
end  
printf('a = %g, b = %g, c = %g\n',a,b,c)
```

- **Větvení programu**

```
select i,
    case 1, prikaz_A;
    case 2, prikaz_B;
    else prikaz_D
end
```

Podle hodnoty i se provede příslušný příkaz

- **Příklad**

```
// Zvolte operaci na vektorech
// zadejte své vektory
a=[1 3 5]; b=[2 4 6];
// 1 - součet
// 2 - skalární součin
// 3 - tenzorový součin
// zadejte číslo operace
```

- **pokračování**

```
i=2;
select i
case 1, d=a+b;
case 2, d=a*b';
case 3, d=a'*b;
end
disp(d,'vysledek')
```

- **Cyklus for**

```
for i=1:5
    a(i)=2*i;
end
```

Pro  $i=1,2,3,4,5$  se provede příkaz  $a(i)=2*i$ ; . Výsledek:  $a=[2, 4, 6, 8, 10]$  .

- **Příklad 1**

```
// Určete vážený průměr
x=[1 2 3 4 5 6]; // čísla
p=[.1 .3 .2 .1 .2 .1]; // váhy
n=length(x);
s=0;
for i=1:n
s=s+x(i)*p(i);
end
disp(s,'prumer je')
```

- **Příklad 2**

```
// Serad'te čísla podle velikosti
n=10; // počet čísel
a=fix(100*rand(1,n,'u')); // čísla
disp(a,'puvodni cisla')
b=[];
for i=1:n
[x,j]=min(a);
b=[b x];
a(j)=%nan;
end
disp(b,'serazena cisla')
end
```

- **Ovládání běhu programu**

pause zastaví běh programu a předá řízení klávesnici.  
resume pokračování běhu programu  
abort - ukončí běh programu.

- **Zadání příkazu stringem**

eval(string) spustí příkaz, který je textově zapsán v proměnné string.  
eval('x=1;') dá příkaz  $x=1$ ;

- **Volání podprogramu**

exec('muj\_program',-1) spustí program muj\_program (-1 potlačí odezvu)

- **Natažení funkcí do paměti**  
`getd('moje_adresa')` natáhne do paměti všechny funkce v adresáři `moje_adresa`  
`getd()` natáhne do paměti funkce z aktuálního adresáře.  
(Pozor, SciLab nemá `path`, a zná jen svoje funkce a ty, které natáhneme)

## 1.5 Tiskový výstup

Pro tisk proměnných slouží příkazy `disp` (bez formátu) a `fprintf` (s formátem).

- `disp(a)` zobrazí hodnotu proměnné `a`.
- `disp(a,'popis')` vypíše proměnnou `a` s popisem
- `printf('prvek %d vektoru a je %g\n',i,a(i));`  
vypíše např.: *prvek 5 vektoru a je 4.12*

## 1.6 Grafický výstup

Dvojměrný graf je možno vykreslit příkazem `plot`.

Příklady:

- `plot(y)` vykreslí hodnoty vektoru `y`. Na ose `x` je pořadí složky vektoru (index).
- `plot(x,y)` vykreslí hodnoty vektoru `y` proti hodnotám vektoru `x` – tzv. xy-graf.
- `plot(a)` vykreslí sloupce matice `a`.

Formátování grafu:

Typ čar	Typ bodů	Barva
- (plná)	. (bod)	r (červená)
: (tečkovaná)	+ (plus)	g (zelená)
-. (čerchovaná)	o (kroužek)	b (modrá)
- (čárkovaná)	x (křížek)	w (bílá)

Pro další možnosti zadejte: `help plot` nebo jděte do Nápovědy SciLabu: `SCILAB HELP >> GRAPHICS > GLOBALPROPERTY`

Příklady:

- `plot(x,'or')` vykreslí vektor `x` pomocí červených kroužků.
- `plot(x,y,'r-+',u,v,'b-x')` vykreslí dvě křivky `(x,y)` a `(u,v)`; první červenou plnou čarou s plusy, druhou modrou plnou čarou s křížky.