

# Vytěžování Dat

## Cvičení 1 – Úvod do Matlabu

Miroslav Čepek  
Michael Anděl



Evropský sociální fond Praha & EU:  
Investujeme do vaší budoucnosti

Fakulta Elektrotechnická, ČVUT

24.9.2013

# Kontakty na přednášející a cvičící

**Miroslav Čepek**

**email** : cepekmir@fel.cvut.cz

**Filip Železný**

**email** : zelezny@fel.cvut.cz

---

**Michael Anděl**

**email** : andelmi2@fel.cvut.cz

Konzultace jsou možné po předchozí domluvě. Preferovaná forma konzultací je přes diskusní fórum předmětu na <https://cw.felk.cvut.cz/forum/forum-164.html>.

# Obsah dnešního cvičení

Úvod do předmětu

Proč Matlab?

Programování pro Matlab

Domácí úloha

Byli bychom rádi, kdybyste si z předmětu odnesli

- ▶ jaké jsou základní úlohy ve vytěžování dat (tím nemyslíme tento předmět),
- ▶ jak základní metody pro tvorbu modelů,
- ▶ představu o vyhodnocování přesnosti a úspěšnosti modelů,
- ▶ širší povědomí o data miningu.

# Organizace cvičení a požadavky na zápočet

- ▶ **Stránky předmětu:** <https://cw.felk.cvut.cz/doku.php/courses/a7b36vyd/start>
- ▶ Náplň cvičení bude odevzdání domácí úlohy a konzultace k zadané domácí úloze. Tj. jejich obsah budou tvořit především dotazy na probranou látku na přednáškách a zadané úlohy.
- ▶ Na začátku každého cvičení bude zadána jedna zápočtová úloha týkající se látky z poslední přednášky.
- ▶ Z každé úlohy se vypracovává krátký protokol, rozsahem přibližně jedna strana A4.
- ▶ Celkem bude 11 zápočtových úloh a všechny je musíte odevzdat. Za každou úlohu můžete získat 3–5 bodů.
- ▶ Dohromady musíte za úlohy získat alepoň 30 bodů z 50.

# Organizace cvičení a požadavky na zápočet

- ▶ **Stránky předmětu:** <https://cw.felk.cvut.cz/doku.php/courses/a7b36vyd/start>
- ▶ **Náplň cvičení bude odevzdání domácích úloh a konzultace k zadané domácí úloze.** Tj. jejich obsah budou tvořit především dotazy na probranou látku na přednáškách a zadané úlohy.
- ▶ **Na začátku každého cvičení bude zadána jedna zápočtová úloha týkající se látky z poslední přednášky.**
- ▶ **Z každé úlohy se vypracovává krátký protokol, rozsahem přibližně jedna strana A4.**
- ▶ Celkem bude 11 zápočtových úloh a všechny je musíte odevzdat. Za každou úlohu můžete získat 3–5 bodů.
- ▶ Dohromady musíte za úlohy získat alespoň 30 bodů z 50.

# Organizace cvičení a požadavky na zápočet

- ▶ **Stránky předmětu:** <https://cw.felk.cvut.cz/doku.php/courses/a7b36vyd/start>
- ▶ Náplň cvičení bude odevzdání domácí úlohy a konzultace k zadané domácí úloze. Tj. jejich obsah budou tvořit především dotazy na probranou látku na přednáškách a zadané úlohy.
- ▶ Na začátku každého cvičení bude zadána jedna zápočtová úloha týkající se látky z poslední přednášky.
- ▶ Z každé úlohy se vypracovává krátký protokol, rozsahem přibližně jedna strana A4.
- ▶ Celkem bude 11 zápočtových úloh a všechny je musíte odevzdat. Za každou úlohu můžete získat 3–5 bodů.
- ▶ Dohromady musíte za úlohy získat alepoň 30 bodů z 50.

# Domácí úkoly

- ▶ Každý domácí úkol (kromě prvních dvou) bude na procvičení/vyzkoušení látky probírané na přednášce ⇒ choďte na přednášky!
- ▶ K získání zápočtu musíte odevzdat všech 11 domácích úkolů.
  - ▶ Na vypracování každého úkolu bude jeden týden, pokud nebude řečeno jinak.
  - ▶ Zprávu budete odevzdávat přes upload systém na stránkách předmětu.
  - ▶ Deadline na odevzdání je půlnoc z pondělí na úterý daného týdne.
- ▶ V případě **závažných** důvodů může cvičící povolit pozdní odevzdání. Problémy řešte co nejdřív!! (A pokud možno, dopředu).



- ▶ Zpráva musí být v upload systému na stránkách předmětu do zadaného deadlinu.
- ▶ Pokud má úloha i programovací část, můžete ji ladit ještě i po deadlinu až do odevzdání na cvičení.
- ▶ Stejně tak, pokud při odevzdání na cvičení vyplují na povrch drobné nedostatky, máte šanci je opravit přímo na cvičení bez ztráty bodů.
- ▶ Body se udělují stylem SPLNĚNO/NESPLNĚNO.
  - ▶ Splnění a odevzdání včas – 3–5 bodů (viz tabulka na stránkách cvičení). V případě, že protokol splňuje obsahové požadavky, ale nesplňuje požadavky formální, může cvičící protokol přijmout se sníženým bodovým ohodnocením.
  - ▶ Splnění a odevzdání s týdením zpožděním – 2 body.
  - ▶ Splnění a odevzdání s dvoutýdenním zpožděním – 0 body (ale stále nárok na zápočet).
  - ▶ Později – neudělení zápočtu.

# Zkouška

- ▶ Body ze cvičení si ponesete ke zkoušce, kde vám budou k užítku :).
- ▶ Zkouška bude hlavně písemná – z písemky můžete získat až 50 bodů.

Výsledná známka bude dána součtem bodů ze zkouškové písemky a cvičení:

ECTS známka	A	B	C	D	E	F
Počet bodů	100-90	89-80	79-70	69-60	59-50	50-0

# Proč proboha Matlab?

- ▶ Matlab je SW pro vědecko-technické výpočty a de facto průmyslový standard v mnoha odvětvích.
- ▶ Vhodný pro rychlé prototypování a zkoušení aplikací.
- ▶ Obsahuje skriptovací jazyk se spoustou knihoven pro různé oblasti (včetně vytěžování dat).
- ▶ Matlab je v tomto předmětu kompromis mezi "klikacími" nástroji typu RapidMiner a programováním v jazycích typu Java.
- ▶ Dovolí nám hrát si s implementací algoritům, ale spoutu větší řeší interně za vás.

# Kde získat Matlab?

- ▶ ČVUT má multilicenci, kterou můžete využít.
- ▶ Kopii instalačního DVD můžete získat na `download.cvut.cz`
- ▶ Zde se musíte přihlásit hlavním přístupovým heslem a pak už můžete stahovat a instalovat.

# Základní uživatelské rozhraní

The screenshot displays the MATLAB 7.11.0 (R2010b) environment. The main window is titled "MATLAB 7.11.0 (R2010b)" and has a menu bar with "File", "Edit", "Debug", "Desktop", "Window", and "Help". The "Current Folder" is set to "/Users/cepekml/Documents/MATLAB".

**Command Window:** Shows the following commands and output:

```
>> x = [1 1 2 2 4 5 6];  
>> mode(x)  
  
ans =  
  
1  
  
>> x = [1 1 2 2 4 5 6 6];  
>> mode(x)  
  
ans =  
  
6  
  
MODE Return the mode value in time series data  
  
MODE(TS) returns the median of TS.Data  
  
MODE(TS,'PropertyName', PropertyValue,...) includes optional arguments:  
    'MissingData': 'remove' (default) or 'interpolate'  
    indicates how to treat missing data during the calculation  
    'Quality': a vector of integers  
    indicates which quality codes represent missing sample (vector case) or missing observations (>2 dimensional case)  
    'Weighting': 'none' (default) or 'time'  
    When 'time' is used, large time values correspond to large weights  
  
See also timeseries/mean, timeseries/igr, timeseries/std  
  
fx >>
```

**Workspace:** Shows a table of variables:

Name	Value	Min	Max
A	<10x5 double>	0.01...	10
B	[1,2,3,4,5,6]	1	6
C	[1,2,3,4,5,6]	7	12
ans	[1,2,3,4,5,6]	1	6
i	[1,2,3,4,5,6]	1	6
row	[1,2,3,4,5,6]	1	3
sl	[1,2,3,4,5,6]	1	3
str	[1,2,3,4,5,6]	1	6
x	[1,1,2,2,4,5,6,6]	1	6
y	[2,4]	2	4

**Command History:** Shows the following commands:

```
A(LAB)(> 0.5);  
fprintf('%d', 5)  
fprintf('%d\n', 5)  
diap(5)  
diap(x)  
for i = [1 6 10000 4 3.1415 2.7 1.415]  
diap(i); end  
factorial(5)  
help  
A > 0.5  
help mode  
x = [1 1 2 2 4 5 6];  
mode(x)  
x = [1 1 2 2 4 5 6 6];  
mode(x)
```

**Annotations:**

- A red box highlights the "Current Folder" path: "/Users/cepekml/Documents/MATLAB".
- A red box highlights the "Current Folder" dropdown menu.
- A red box highlights the "Command Window" output area.
- A red box highlights the "Workspace" table.
- A red box highlights the "Command History" window.

**Text Labels:**

- "Aktuální pracovní adresář" (Current working directory) points to the "Current Folder" dropdown.
- "Obsah aktuálního adresáře" (Content of current directory) points to the "Current Folder" pane.
- "Příkazový řádek pro zadávání příkazů a jejich výstup" (Command line for entering commands and their output) points to the "Command Window".
- "Aktuálně definované proměnné a jejich hodnoty (zkuste dvojklik)" (Currently defined variables and their values (try double-click)) points to the "Workspace" table.
- "Historie příkazů" (Command history) points to the "Command History" window.

# Jednoduché příkazy

Do příkazového okna zkuste zadat následující příkazy. Jaký je jejich význam a co je výsledkem?

- ▶  $1 + 5$
- ▶ Ukládá se někam výsledek?
- ▶  $x = 3$
- ▶  $y = 1 * 6$
- ▶  $x + y$
- ▶  $z = x + y$
- ▶  $z = x + y;$
- ▶ Jaký je rozdíl mezi předchozími dvěma příkazy?

# Jednoduché příkazy

Do příkazového okna zkuste zadat následující příkazy. Jaký je jejich význam a co je výsledkem?

- ▶  $1 + 5$
- ▶ Ukládá se někam výsledek?
- ▶  $x = 3$
- ▶  $y = 1 * 6$
- ▶  $x + y$
- ▶  $z = x + y$
- ▶  $z = x + y;$
- ▶ Jaký je rozdíl mezi předchozími dvěma příkazy?

# Jednoduché příkazy

Do příkazového okna zkuste zadat následující příkazy. Jaký je jejich význam a co je výsledkem?

▶  $1 + 5$

▶ Ukládá se někam výsledek?

▶  $x = 3$

▶  $y = 1 * 6$

▶  $x + y$

▶  $z = x + y$

▶  $z = x + y;$

▶ Jaký je rozdíl mezi předchozími dvěma příkazy?



# Jednoduché příkazy

Do příkazového okna zkuste zadat následující příkazy. Jaký je jejich význam a co je výsledkem?

▶  $1 + 5$

▶ Ukládá se někam výsledek?

▶  $x = 3$

▶  $y = 1 * 6$

▶  $x + y$

▶  $z = x + y$

▶  $z = x + y;$

▶ Jaký je rozdíl mezi předchozími dvěma příkazy?

# Jednoduché příkazy

Do příkazového okna zkuste zadat následující příkazy. Jaký je jejich význam a co je výsledkem?

- ▶  $1 + 5$
- ▶ Ukládá se někam výsledek?
- ▶  $x = 3$
- ▶  $y = 1 * 6$
- ▶  $x + y$
- ▶  $z = x + y$
- ▶  $z = x + y;$
- ▶ Jaký je rozdíl mezi předchozími dvěma příkazy?

# Matice

- ▶ Hlavní síla Matlabu spočívá v práci s maticemi.
- ▶ Jak vypadají matice z následujících příkazů?

```
A = [1 2; 3 4; 5 6]
```

```
B = [1 2 3; 4 5 6]
```

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$$

$$B = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

# Matice

- ▶ Hlavní síla Matlabu spočívá v práci s maticemi.
- ▶ Jak vypadají matice z následujících příkazů?

```
A = [1 2; 3 4; 5 6]
```

```
B = [1 2 3; 4 5 6]
```

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$$

$$B = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

# Matcové operace

- ▶ **Jaké znáte maticové operace?**
- ▶ Maticové sčítání, odčítání, násobení, dělení, transpozice
- ▶  $C = [7 \ 8; \ 9 \ 10; \ 11 \ 12]$
- ▶  $A + C$
- ▶  $A * B$
- ▶  $A .* C$
- ▶ Jak se liší výsledky posledních dvou příkazů?
- ▶ Co dělá  $A'$  ?

# Matcové operace

- ▶ Jaké znáte maticové operace?
- ▶ Maticové sčítání, odčítání, násobení, dělení, transpozice
- ▶  $C = [7 \ 8; \ 9 \ 10; \ 11 \ 12]$
- ▶  $A + C$
- ▶  $A * B$
- ▶  $A .* C$
- ▶ Jak se liší výsledky posledních dvou příkazů?
- ▶ Co dělá  $A'$  ?

# Matcové operace

- ▶ Jaké znáte maticové operace?
- ▶ Maticové sčítání, odčítání, násobení, dělení, transpozice
- ▶  $C = [7 \ 8; \ 9 \ 10; \ 11 \ 12]$
- ▶  $A + C$
- ▶  $A * B$
- ▶  $A .* C$
- ▶ Jak se liší výsledky posledních dvou příkazů?
- ▶ Co dělá  $A'$  ?

# Matcové operace

- ▶ Jaké znáte maticové operace?
- ▶ Maticové sčítání, odčítání, násobení, dělení, transpozice
- ▶  $C = [7 \ 8; \ 9 \ 10; \ 11 \ 12]$
- ▶  $A + C$
- ▶  $A * B$
- ▶  $A .* C$
- ▶ Jak se liší výsledky posledních dvou příkazů?
- ▶ Co dělá  $A'$  ?



# Funkce pro práci s maticemi

- ▶ **Matice náhodných čísel** `rand(<počet prvků v 1. dimenzi>, <v 2. dimenzi>, ...)`
  - ▶ například `A = rand(10, 5)`
- ▶ **Velikost matice, počet prvků v jednotlivých dimenzích** – `size(A)`
- ▶ **Matice samých nul** – `zeros(10, 6)`
- ▶ **Inverzní matice** – `inv(A)`
- ▶ **Vlastní čísla** – `eig(A)`
- ▶ **Determinant** – `det(A)`
- ▶ **Indikace splnění podmínky** – `A > 0.5`
- ▶ **Vybere indexy z matice na základě podmínky** – `find(A > 0.5)`

Existuje několik možností získání nápovědy pro příkaz Matlabu.

- ▶ Jednoduchá textová nápověda – příkaz `help <příkaz>`
- ▶ Hypertextová nápověda – příkaz `doc <příkaz>`
- ▶ Mathworld Knowledge Base –  
<http://www.mathworks.com/help/techdoc/>

# A co vektory?

- ▶ Vektory jsou matice, které mají jen jeden řádek/sloupec.
- ▶ Vytvořte řádkový a sloupcový vektor hodnot 1, 2, 3, 4, 5, 6.

- ▶ `row = [1 2 3 4 5 6]`

- ▶ `sl = [1; 2; 3; 4; 5; 6]`

- ▶ Jak vytvořím z řádkového vektoru sloupcový?

- ▶ Transpozicí – `row'`

# A co vektory?

- ▶ Vektory jsou matice, které mají jen jeden řádek/sloupec.
- ▶ Vytvořte řádkový a sloupcový vektor hodnot 1, 2, 3, 4, 5, 6.
- ▶ `row = [1 2 3 4 5 6]`
- ▶ `sl = [1; 2; 3; 4; 5; 6]`
- ▶ Jak vytvořím z řádkového vektoru sloupcový?
- ▶ Transpozicí – `row'`

# A co vektory?

- ▶ Vektory jsou matice, které mají jen jeden řádek/sloupec.
- ▶ Vytvořte řádkový a sloupcový vektor hodnot 1, 2, 3, 4, 5, 6.
- ▶ `row = [1 2 3 4 5 6]`
- ▶ `sl = [1; 2; 3; 4; 5; 6]`
- ▶ Jak vytvořím z řádkového vektoru sloupcový?
- ▶ Transpozicí – `row'`

# A co vektory?

- ▶ Vektory jsou matice, které mají jen jeden řádek/sloupec.
- ▶ Vytvořte řádkový a sloupcový vektor hodnot 1, 2, 3, 4, 5, 6.
- ▶ `row = [1 2 3 4 5 6]`
- ▶ `sl = [1; 2; 3; 4; 5; 6]`
- ▶ Jak vytvořím z řádkového vektoru sloupcový?
- ▶ Transpozicí – `row'`

# Generování posloupností čísel

Pokud potřebujete vytvořit posloupnost čísel, můžete využít příslušný operátor ":" (dvojtečka).

Pomocí něj vygeneruje vektor, který obsahuje posloupnost čísel v zadaném rozmezí – zkuste

- ▶ `[1:10]; [-5:5]; [5.4:15.6]`

- ▶ můžete zadat i krok, se kterým se posloupnost mění.

Zkuste: `[1:0.5:10]; [5:-1:5]; [5.4:0.2:15.6]`

# Generování posloupností čísel

Pokud potřebujete vytvořit posloupnost čísel, můžete využít příslušný operátor ":" (dvojtečka).

Pomocí něj vygeneruje vektor, který obsahuje posloupnost čísel v zadaném rozmezí – zkuste

- ▶ `[1:10]; [-5:5]; [5.4:15.6]`

- ▶ můžete zadat i krok, se kterým se posloupnost mění.

Zkuste: `[1:0.5:10]; [5:-1:5]; [5.4:0.2:15.6]`



# Přístup k prvkům matice

Zopakujte `A = rand(10, 5)`, ať máme všichni stejné rozměry matice.

- ▶ Přístup k jednomu konkrétnímu prvku – `A(1, 2)`
- ▶ **POZOR** – indexy se číslují od **1**!
- ▶ Přístup k podmaticím – uvedu rozsahy indexů, které chci v mít podmatici `A(3:5, 1:3)`
- ▶ Místo čísel můžu uvést i vektory indexů, které chci zobrazit.
- ▶ Zkuste zobrazit prvky ve řádcích 1., 5., 3., 8. a sloupcích 3., 2., 1.
- ▶ `x = [1 5 3 8]; y = [3 2 1]; A(x,y)`

# Přístup k prvkům matice

Zopakujte `A = rand(10, 5)`, ať máme všichni stejné rozměry matice.

- ▶ Přístup k jednomu konkrétnímu prvku – `A(1, 2)`
- ▶ **POZOR** – indexy se číslují od **1**!
- ▶ Přístup k podmaticím – uvedu rozsahy indexů, které chci v mít podmatici `A(3:5, 1:3)`
- ▶ Místo čísel můžu uvést i vektory indexů, které chci zobrazit.
- ▶ Zkuste zobrazit prvky ve řádcích 1., 5., 3., 8. a sloupcích 3., 2., 1.
- ▶ `x = [1 5 3 8]; y = [3 2 1]; A(x,y)`

# Přístup k prvkům matice

Zopakujte `A = rand(10, 5)`, ať máme všichni stejné rozměry matice.

- ▶ Přístup k jednomu konkrétnímu prvku – `A(1, 2)`
- ▶ **POZOR** – indexy se číslují od **1**!
- ▶ Přístup k podmaticím – uvedu rozsahy indexů, které chci v mít podmatici `A(3:5, 1:3)`
- ▶ Místo čísel můžu uvést i vektory indexů, které chci zobrazit.
- ▶ Zkuste zobrazit prvky ve řádcích 1., 5., 3., 8. a sloupcích 3., 2., 1.

```
▶ x = [1 5 3 8]; y = [3 2 1]; A(x,y)
```

# Přístup k prvkům matice

Zopakujte `A = rand(10, 5)`, ať máme všichni stejné rozměry matice.

- ▶ Přístup k jednomu konkrétnímu prvku – `A(1, 2)`
- ▶ **POZOR** – indexy se číslují od **1**!
- ▶ Přístup k podmaticím – uvedu rozsahy indexů, které chci v mít podmatici `A(3:5, 1:3)`
- ▶ Místo čísel můžu uvést i vektory indexů, které chci zobrazit.
- ▶ Zkuste zobrazit prvky ve řádcích 1., 5., 3., 8. a sloupcích 3., 2., 1.
- ▶ `x = [1 5 3 8]; y = [3 2 1]; A(x, y)`

# Přiřazování hodnot do matice

- ▶ Úplně stejně jako výběr prvků na minulém slajdu, jen výběr umístím na levou stranu přiřazení.
- ▶ Přiřazení jedné hodnoty – přiřaďte 10 do levého horního rohu matice.
  - ▶  $A(1,1) = 10$
- ▶ Přiřazení do podmatice – přiřaďte hodnoty 1, 2, 3, 4 na souřadnice (2,2), (2,4), (4,2), (4,4)
  - ▶  $x=[2\ 4]; y=[2\ 4]; A(x,y) = [1\ 3; 2\ 4]$

# Přiřazování hodnot do matice

- ▶ Úplně stejně jako výběr prvků na minulém slajdu, jen výběr umístím na levou stranu přiřazení.
- ▶ Přiřazení jedné hodnoty – přiřaďte 10 do levého horního rohu matice.
- ▶  $A(1, 1) = 10$
- ▶ Přiřazení do podmatice – přiřaďte hodnoty 1, 2, 3, 4 na souřadnice (2,2), (2,4), (4,2), (4,4)
- ▶  $x=[2\ 4]; y=[2\ 4]; A(x,y) = [1\ 3; 2\ 4]$

# Přiřazování hodnot do matice

- ▶ Úplně stejně jako výběr prvků na minulém slajdu, jen výběr umístím na levou stranu přiřazení.
- ▶ Přiřazení jedné hodnoty – přiřaďte 10 do levého horního rohu matice.
- ▶  $A(1, 1) = 10$
- ▶ Přiřazení do podmatice – přiřaďte hodnoty 1, 2, 3, 4 na souřadnice (2,2), (2,4), (4,2), (4,4)
- ▶  $x=[2\ 4]; y=[2\ 4]; A(x,y) = [1\ 3; 2\ 4]$

# Přiřazování hodnot do matice

- ▶ Úplně stejně jako výběr prvků na minulém slajdu, jen výběr umístím na levou stranu přiřazení.
- ▶ Přiřazení jedné hodnoty – přiřaďte 10 do levého horního rohu matice.
- ▶  $A(1, 1) = 10$
- ▶ Přiřazení do podmatice – přiřaďte hodnoty 1, 2, 3, 4 na souřadnice (2,2), (2,4), (4,2), (4,4)
- ▶  $x=[2 \ 4]; \ y=[2 \ 4]; \ A(x, y) = [1 \ 3; \ 2 \ 4]$



# Datové typy

- ▶ V Matlabu, stejně jako v jiných skriptovacích jazycích, proměnné nemají pevný datový typ.
- ▶ Základní datové typy jsou:
  - ▶ Čísla, Logické hodnoty, Řetězce
  - ▶ Matice – matice hodnot jednoho datového typu
  - ▶ Struktury – skupina několika pojmenovaných hodnot zabalených do jedné proměnné
  - ▶ Buňková pole (Cell arrays) – pole hodnotu různých datových typů
  - ▶ Odkazy (Handles)
  - ▶ Objekty

Více o datových typech se lze dozvědět na

[http://www.mathworks.com/help/techdoc/matlab\\_prog/f2-43934.html](http://www.mathworks.com/help/techdoc/matlab_prog/f2-43934.html)

# Programování pro Matlab

Doteď jsme zkoušeli interaktivní práci se systémem Matlab. Teď zkusíme programování.

Programy se zadávají do tzv. M-souborů (M-file) což je jen textový soubor s koncovkou `.m`. Existují dva typy M-souborů

- ▶ Skripty – obsahují jen posloupnost příkazů Matlabu,
- ▶ Soubory definující funkci – obsahuje definici funkce, kterou lze využít při interaktivní práci s Matlabem nebo v jiných M-souborech.

Pro editaci obou typů M-souborů můžete použít:

- ▶ libovolný textový editor,
- ▶ editor integrovaný přímo do Matlabu.

My použijeme druhou možnost. Interní editor se spouští buď příkazem `edit` nebo z menu `File > New Script` resp. `File > New Function`.

# Struktura funkce

Základní struktura funkce vypadá takto:

```
function [soucet, rozdil] = SlozitaFunkce(a,b)
%
% Funkce se jmenuje SlozitaFunkce a ma dva
vstupni
% parametry - a, b. A dva vystupni parametry
-
% soucet, rozdil.
% Tohle je help k funkci SlozitaFunkce.
Vypisuje se
% zadanim prikazu 'help SlozitaFunkce'.
%
%Naplneni vystupniho parametru se deje
prirazanim
    soucet = a+b;
    rozdil = a-b;
end % Nepovinne end
```

Funkce **musí** být uložena v souboru slozitafunkce.m.

- ▶ Funkci můžete zavolat pomocí jejího jména.
- ▶ `[s, r] = slozitafunkce(10, 4)`
- ▶ Jak Matlab zjistí, kterou funkci voláte?
  - ▶ Matlab se podívá do aktuálního adresáře, zda v aktuálním adresáři existuje soubor `slozitafunkce.m`
  - ▶ Pokud Matlab funkci nenajde v aktuálním adresáři, podívá se na vnitřní proměnné `PATH` na seznam adresářů, které se mají prohledávat a zkusí, zda některý z nich funkci neobsahuje.
- ▶ Z toho plyne, že je důležité, ve kterém adresáři se nacházíte :).

# Volání funkce

- ▶ Funkci můžete zavolat pomocí jejího jména.
- ▶ `[s, r] = slozitafunkce(10, 4)`
- ▶ Jak Matlab zjistí, kterou funkci voláte?
  - ▶ Matlab se podívá do aktuálního adresáře, zda v aktuálním adresáři existuje soubor `slozitafunkce.m`
  - ▶ Pokud Matlab funkci nenajde v aktuálním adresáři, podívá se na vnitřní proměnné `PATH` na seznam adresářů, které se mají prohledávat a zkusí, zda některý z nich funkci neobsahuje.
- ▶ Z toho plyne, že je důležité, ve kterém adresáři se nacházíte :).

- ▶ Funkci můžete zavolat pomocí jejího jména.
- ▶ `[s, r] = slozitafunkce(10, 4)`
- ▶ Jak Matlab zjistí, kterou funkci voláte?
  - ▶ Matlab se podívá do aktuálního adresáře, zda v aktuálním adresáři existuje soubor `slozitafunkce.m`
  - ▶ Pokud Matlab funkci nenajde v aktuálním adresáři, podívá se na vnitřní proměnné `PATH` na seznam adresářů, které se mají prohledávat a zkusí, zda některý z nich funkci neobsahuje.
- ▶ Z toho plyne, že je důležité, ve kterém adresáři se nacházíte :).

# Příkazy řízení programu

Provádění funkce můžete ovlivnit pomocí standardních konstrukcí, které znáte z jiných programovacích jazyků.

- ▶ Podmínky
  - ▶ if – then – else
  - ▶ switch – case
- ▶ Cykly
  - ▶ for
  - ▶ while

## **if – then – else**

```
if x > 11
    disp('x je vetsi nez 11.');
```

elseif x < 5

```
    disp('x je mensi nez 11 a take mensi nez
5.')
```

else

```
    disp('Je to nejak uplne jinak.');
```

end



## **switch – case**

```
switch x
    case {0, 1, 2, 3}
        disp('x je v intervalu 0-3.');
```

case 4

```
        disp('x je 4.');
```

otherwise

```
        disp('x je uplne jine.')
```

end

# Cykly – for cyklus

For cyklus funguje trochu jinak než jste zvyklí z Javy. For cyklus v Matlabu iteruje přes všechny hodnoty vektoru.

```
for i = 1:10
    fprintf('%d ', i); %muzete pouzit take
disp(i)
end
```

```
for i = [1 6 10000 4 3.1415 2.7 1.41]
    disp(i);
end
```

**POZOR** – použití cyklů je extrémně nevýhodné, protože Matlab parsuje každý řádek znovu ⇒ pokud můžete zkuste cykly obejít (například vektorovými nebo maticovými operacemi).

# Cykly – While cyklus

While cyklus je mnohem blíž tomu, co znáte z jiných jazyků.

```
i = 1;
while i < 10
    fprintf('%d', i);
    i = i+1;
end
```

Existují i příkazy na přerušení cyklů

- ▶ `break` – ukončuje cyklus if `i = 5, break, end`
- ▶ `continue` – přeskakuje zbytek iterace if `i = 5, continue, end`

## Příklad - faktorial

```
function fac = factorial(n)
if n < 0
    disp('n musi byt vetsi nez 0!');
    fac = 0;
    return;
end
if n == 0
    fac = 1;
    return;
end
fac = 1;
for i = 1:n
    fac = fac * i;
end
end
```

# 1. zápočtová úloha (I)

**Termín odevzdání:** 24.9. 2012 nejpozději v 23:59:59 (SELČ).  
Do upload systému na stránkách předmětu.

**Zadání** – vytvořte v Matlabu:

- ▶ funkci `fib_arr(array)`, která pro všechny prvky ve vektoru `array` spočítá fibonacciho hodnotu a vrátí pole těchto hodnot.
- ▶ funkci `stat_props(filename)`, která načte zadaný soubor ve formátu CSV a pro každý **řádek** hodnot vypíše maximum, minimum, medián, průměr, rozptyl, stření hodnotu.

Použití všech funkcí Matlabu je samozřejmě dovoleno!

# 1. zápočtová úloha (II)

Protokol z první úlohy, který nahrajete na web, bude PDF s textovou zprávou. Bude obsahovat:

- ▶ stručný popis vašeho řešení a okomentované nejpodstatnější části kódu vašich funkcí,
- ▶ **krátká** ukázková data a výsledky,
- ▶ **jednovětný** popis termínů uvedených v zadání (fibonacciho posloupnost, medián, rozptyl, ...).

Funkčnost vašeho kódu předvedete na cvičení.

*Poznámka:* Pro načítání souboru ve formátu CSV můžete použít funkci `csvread`.

## Zde jsou odkazy na další výukové materiály o Matlabu

- ▶ <http://labe.felk.cvut.cz/~posik/y33aui/uvod-do-matlabu/>
- ▶ [http://www.mathworks.com/help/techdoc/matlab\\_product\\_page.html](http://www.mathworks.com/help/techdoc/matlab_product_page.html)
- ▶ <http://www.mathworks.com/moler/intro.pdf>
- ▶ <http://www.maths.dundee.ac.uk/~ftp/na-reports/MatlabNotes.pdf>
- ▶ a mnoho dalších...