Fakulta elektrotechniky
a informatiky

Úvod do Matlabu

Prednáška č. 3

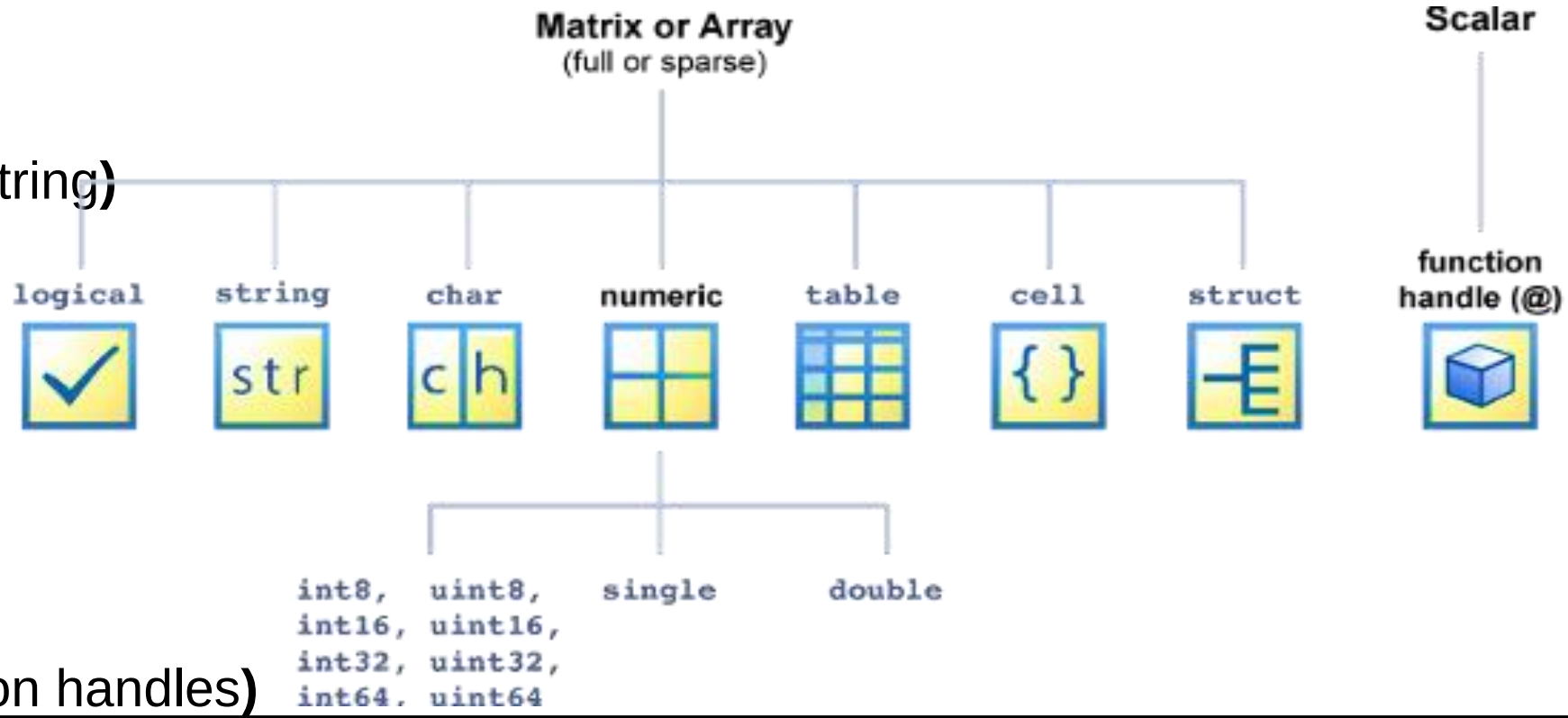
- **Údajové typy (string, struct, cell, table)**
- Príkazový riadok – základné príkazy
- Práca s priečinkom a súborom
- Prvý skript
- Debugging

Ing. Ondrej Kováč, PhD.

Údajové typy – čo už vieme z minulej prednášky

- V Matlabe sa každá premenná považuje za maticu. Dokonca i jedno číslo, napr. a = 5 bude považované za maticu s rozmerom 1x1.
- Medzi najčastejšie používané údajové typy patria:

- **Číselné (**double, single, int8, int16, int32, int64, uint8, uint16, uint32, uint64**)**
- **Logické (**logical**)**
- **Znaky a reťazce (**char, string**)**
- **Polia buniek (**cell**)**
- **Štruktúry (**struct**)**
- **Tabuľky (**table**)**
- **Rukoväť funkcie (**function handles**)**



Údajové typy – string (reťazec)

- Rovnako ako v programovacom jazyku C aj v Matlabe máme údajový typ, pomocou ktorého vieme pracovať s reťazcami.
- **Reťazec sa definuje pomocou apostrofou alebo úvodzoviek.**

```
>> a = 'ahoj'
a =
ahoj

>> whos a
  Name        Size    Bytes      Class    Attributes
  a           1x4     8          char
```

- Môžeme si všimnúť, že premenná **a** je uložená ako vektor znakov (**char**) 1x4. Dĺžku reťazca **a** môžeme zistiť pomocou príkazu **length(a)**. **Pozor! V Matlabe neuvažujeme o terminátoroch. Nepracujeme s pamäťou ako v C-čku.**
- V prípade, že potrebujeme mať viacero reťazcov uložených ako riadky matice, je to možné, ale jednotlivé reťazce musia mať pri definovaní rovnakú dĺžku, inak nastane chyba.

Údajové typy – string (reťazec)

- V prípade, že potrebujeme mať viacero reťazcov uložených ako riadky matice, je to možné, ale jednotlivé reťazce musia mať pri definovaní rovnakú dĺžku, inak nastane chyba.
- Ak chceme pridať ďalší reťazec ako nový riadok matice, musíme zadefinovať aký dlhý reťazec je a o koľko sa musí matica rozšíriť.

```
>> slova = ['ahoj'; 'jan']
Dimensions of matrices being
concatenated are not
consistent.

>> slova = ['ahoj'; 'jano']
slova =
ahoj
jano

>> whos slova
Name     Size   Bytes   Class
slova    2x4    16      char
```

```
>> slova(3,:) = 'mlato'
Subscripted assignment dimension
mismatch.

>> slova(3,1:5) = 'mlato'
slova =
ahoj
jano
mlato

>> whos slova
  Name     Size   Bytes   Class
  slova    3x5    30      char
```

- Z výpisov premennej **slova** si môžeme všimnúť, že reťazce boli pôvodne uložené v matici 2x4 a po pridaní ďalšieho reťazca sa celá matica predĺžila o jeden stĺpec na rozmer 3x5. Takáto práca s reťazcami programátora značne obmedzuje.
- Pre prácu s väčším množstvom reťazcov je niekedy vhodnejšie použiť údajový typ **cell.**

Údajové typy – string (reťazec) – základné operácie

- Hoci Matlab nie je primárne určený na prácu s reťazcami, poskytuje možnosť s nimi efektívne pracovať. V Tab. sú uvedené niektoré základné funkcie pre prácu s reťazcami.

Funkcia	Popis funkcie
strcat	Spojiť viacero reťazcov do jedného
strcmp	Porovnať dva reťazce
strcmpi	Porovnať dva reťazce (citlivé na veľkosť)
sprintf	Formátovať dáta do reťazca
cell2str	Konverzia typu cell na typ string
num2str	Konverzia číselného typu na typ string
str2double	Konverzia reťazca na číslo
strfind	Vyhľadanie reťazca v inom reťazci
strrep	Nahradenie reťazca novým reťazcom, v rámci reťazca, ktorý obsahuje pôvodný reťazec.
strsplit	Rozdelenie reťazca na viacero reťazcov.
eval	Vykoná príkaz uložený ako reťazec

Údajové typy – string (reťazec) – základné operácie

- **strcat(reťazec1, reťazec2, reťazec3)**
  - Spojenie reťazcov do jedného
- **strrep(reťazec, ‘nahrádzaný’, ‘náhrada’)**
  - Nahradenie reťazca iným
- **strsplit(reťazec, ‘vzor’)**
  - Rozdeli reťazec do viacerých reťazcov podľa vzoru reps. znaku
- **strfind(reťazec, ‘hľadaný_výraz’)**
  - Nájde pozíciu kde sa v reťazci začína hľadaný výraz
- **sprintf(‘text + premenné’, premenné)**
  - Vytvorí formátovaný text aj s premennými
- **eval(reťazec)**
  - Vykoná reťazec ako príkaz

```
>> a = 'ahoj';
>> b = ',';
>> c = 'Jano';

>> abc = strcat(a,b,c)
abc =
ahoj,Jano

>> strrep(abc,'Jano','Michal')
ans =
ahoj,Michal

>> strsplit(abc,',')
ans =
    'ahoj'    'Jano'

>> strfind(abc, 'Jano')
ans =
     6

s = sprintf('Ahoj, %s. prave bolo %d hodin',c,17)
s =
Ahoj, Jano. prave bolo 17 hodin

>> fc = 'res = 1 + 2';

>> eval(fc)
res =
     3
```

Údajové typy - Polia buniek - cells

- Údajový typ **cell** umožňuje v jednej premennej ukladať viacero údajových typov súčasne.

- Napr. v údajovom type **double** je možné uložiť len číslo, v údajovom type **string** len znaky a reťazce.

- Jednotlivé položky sa do poľa buniek pridávajú podobne ako v prípade matice, ale namiesto hranatých zátvoriek **sa používajú zložené zátvorky {}.** Indexácia v poli buniek je potom rovnaká, ako to bolo pri numerickom type.

```
>> bunky = {'ahoj, jano',105,NaN,inf,[1 2; 2 1] }

bunky =
    'ahoj, jano'    [105]    [NaN]    [Inf]    [2x2 double]

>> whos bunky
  Name          Size            Bytes  Class     Attributes
  bunky         1x5               636  cell
```

Pozn.
NaN – nedefinovaný stav
Inf - nekonečno

```
>> bunky{5}/bunky{2}
ans =
    0.0095    0.0190
    0.0190    0.0095
```

Údajové typy – štruktúry - struct

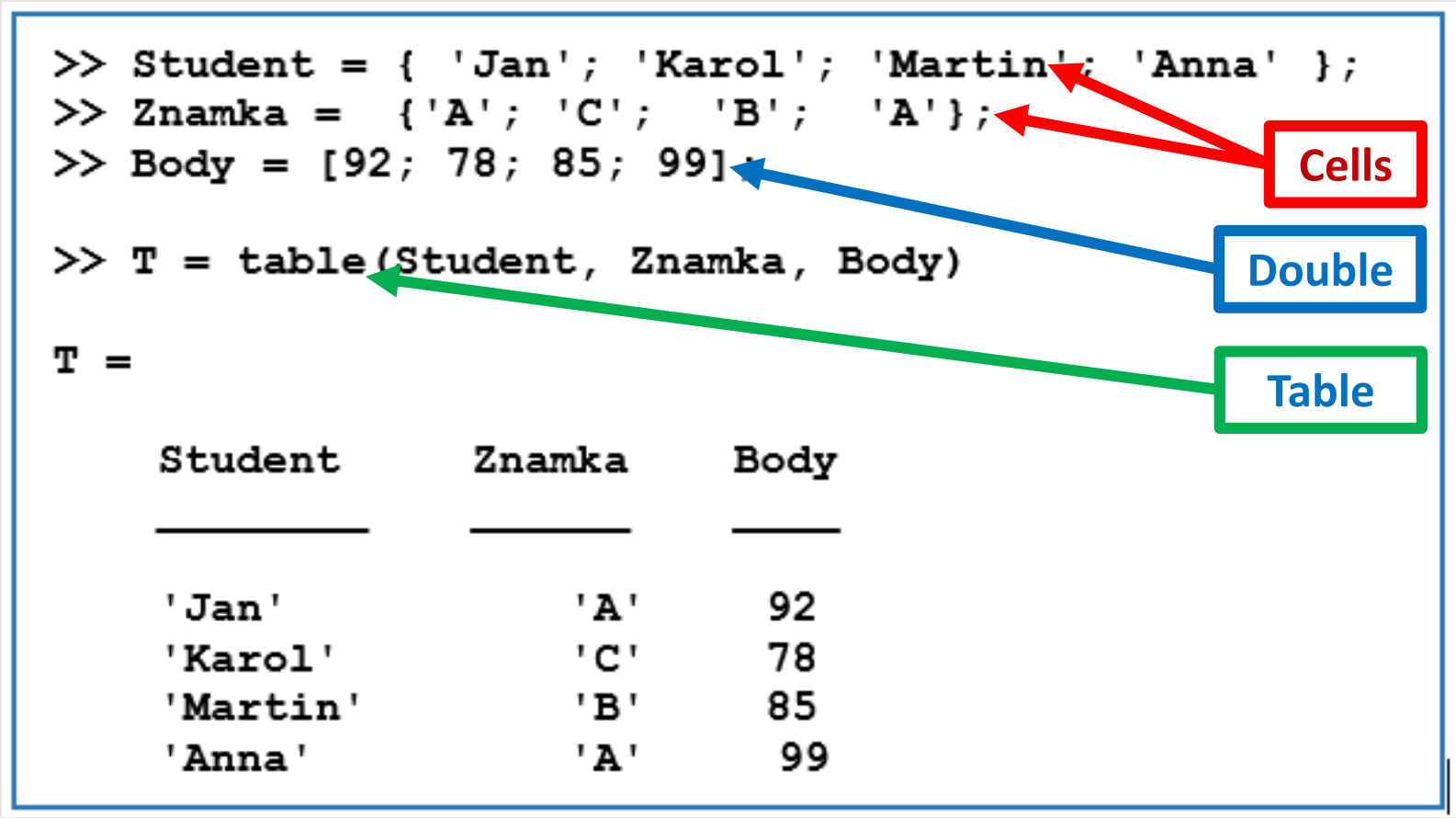
- Údajový typ štruktúra je veľmi podobná tomu čo poznáme z programovacieho jazyka C.
- **K údajom pristupujeme cez „ . “**
- **Štruktúru zvyčajne vytvárame vtedy, ak chceme vytvoriť nejakú entitu, ktorá obsahuje viacero parametrov.**
- Štruktúra je veľmi výhodná pre návratové hodnoty funkcii, ktoré vracajú veľmi veľa rôznych údajov.

Command Window

```
>> osoba.ID = 001;
>> osoba.meno = 'Jan Hrach';
>> osoba.zamestnanie = 'vodič';
>> osoba.bydlisko = 'Slničková 10, 012 45'

osoba =

  struct with fields:

            ID: 1
          meno: 'Jan Hrach'
    zamestnanie: 'vodič'
       bydlisko: 'Slničková 10, 012 45'
```

Údajové typy – Tabuľky - Tables

- Tento údajový typ, takisto ako pole buniek, umožňuje uchovávať rozličné typy údajov.
- Výhodou oproti poľu buniek je to, že tabuľka umožňuje efektívne zobraziť v nej uložené údaje.



Fakulta elektrotechniky
a informatiky

Úvod do Matlabu

Prednáška č. 3

- Údajové typy (string, struct, cell, table)
- **Príkazový riadok – základné príkazy**
- Práca s priečinkom a súborom
- Prvý skript
- Debugging

Ing. Ondrej Kováč, PhD.

Príkazový riadok – základné príkazy

V príkazovom riadku resp. v skripte či funkcii sa najčastejšie môžeme stretnúť s nasledovnými príkazmi. Pre potreby tohto predmetu si nazveme **príkazy príkazového riadku**.

Príkaz	Funkcia príkazu
;	Potlačenie zobrazenia výstupu do príkazového riadku, oddelenie príkazov v jednom riadku
%	Komentár
clc	Vymazanie príkazového okna
...	Rozdelenie dlhého príkazu na viac riadkov
ans	Posledná odpoveď, posledný výsledok

Príkazový riadok – základné príkazy – pre workspace

Príkaz	Funkcia príkazu
clear var1, var2 ...	Vymazanie konkrétnej premennej z workspace
who	Zobrazenie názvov premenných vo workspace
whos	Zobrazenie detailov o premenných vo workspace
exist	Overenie existencie premennej vo workspace
disp	Zobrazenie obsahu premennej
Var = input('Promt')	Zadanie hodnoty do premennej Var počas behu programu
save	Uloženie premenných z workspace na pevný disk
load	Načítanie uložených premenných do workspace

Fakulta elektrotechniky
a informatiky

Úvod do Matlabu

Prednáška č. 3

- Údajové typy (string, struct, cell, table)
- Príkazový riadok – základné príkazy
- **Práca s priečinkom a súborom**
- Prvý skript
- Debugging

Ing. Ondrej Kováč, PhD.

Práca s priečinkom a súborom – základné príkazy

- V tabuľke sú uvedené príkazy, ktoré poznáme z operačného systému LINUX.
- Pribudli len príkazy **addpath** a **rmpath,** ktoré sú výlučne záležitosťou Matlabu. Tieto príkazy sú určené na určenie cesty k funkciám, ktoré nie sú v jadre a toolboxoch Matlabu a zároveň sa nenachádzajú ani v aktuálnom priečinku.

Príkaz	Funkcia príkazu
pwd	Zobrazenie cesty do aktuálneho priečinku
cd	Zmena priečinku
dir/ls	Zobrazenie obsahu priečinku
mkdir	Vytvorenie nového priečinku
rmdir	Vymazanie priečinku
delete	Vymazanie súboru
addpath	Pridanie cesty
rmpath	Odstránenie cesty

Fakulta elektrotechniky
a informatiky

Matlab v elektronike

Prednáška č. 3

- Údajové typy (string, struct, cell, table)
- Príkazový riadok – základné príkazy
- Práca s priečinkom a súborom
- **Prvý skript**
- Debugging

Ing. Ondrej Kováč, PhD.

Prvý skript

- Až doposiaľ sme sa zameriavali na základné funkcie a údajové typy Matlabu. Pritom sme si v podstate vystačili len s príkazovým riadkom.

- Opakované písanie príkazov je značne nepraktické.

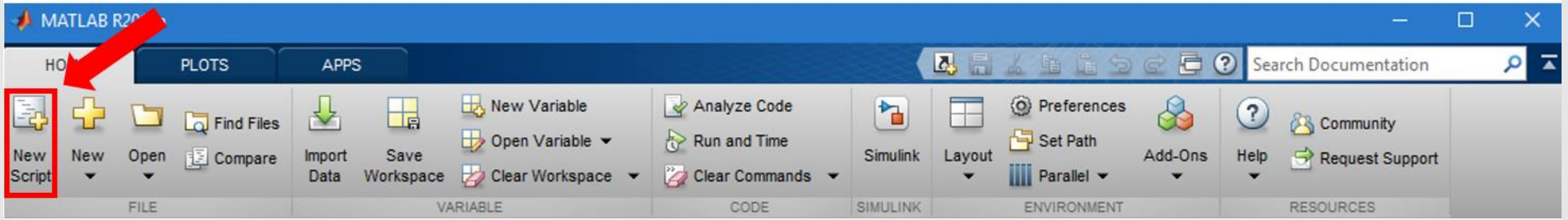
- Skripty sú bloky kódu uložené v súbore, ktorý má **príponu .m**. V literatúre, ale aj v hovorovej reči Matlab-komunity sa stretávame s pojmom **m-file** resp. **m-súbor**.

- Matlab má implementovaný vlastný editor s automatickou kontrolou syntaxi, zobrazovaním tipov a nápovedi k funkciám.

Prvý skript - Editor

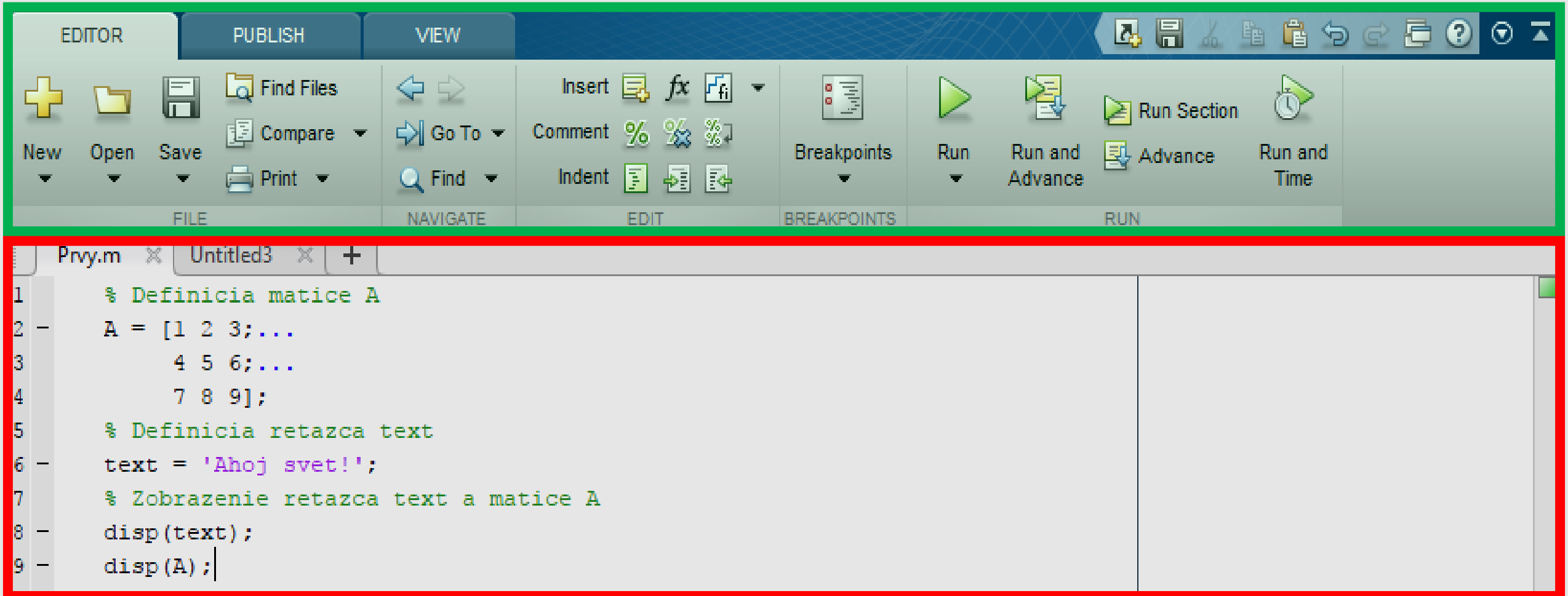
- Editor vyvoláme stlačením tlačidla **New Script** na hlavnej lište Matlabu alebo volaním príkazu **edit priamo z príkazového riadka**.

- Eventuálne **dvojklikom na m-súbor**, v aktuálnom priečinku alebo **napísaním príkazu** **open** alebo **edit** nasledovaným názvom súboru.



Prvý skript - Editor

- Okno editora pozostáva z 3 záložiek. Zatiaľ je pre nás podstatná záložka editor, ktorá je prvá v poradí.
- Okno editora je možne rozdeliť na dve časti – **panel nástrojov** a **textový editor**

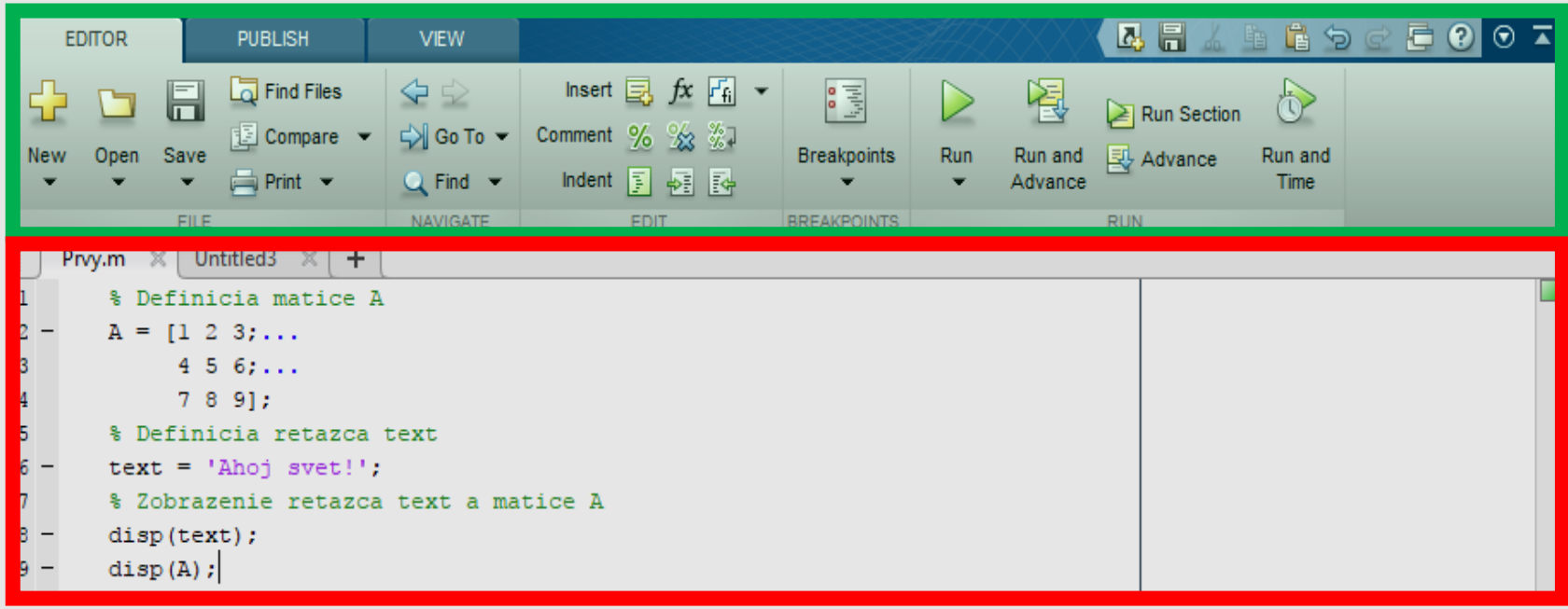


Prvý skript - Editor

Lišta nástrojov obsahuje nástroje pre:

- Prácu so súborom
- Editáciu
- Breakpointy
- Debugging

Okno textového editora je určené na samotné písanie kódu.



Prvý skript

Skript z pravidla pozostáva zo:

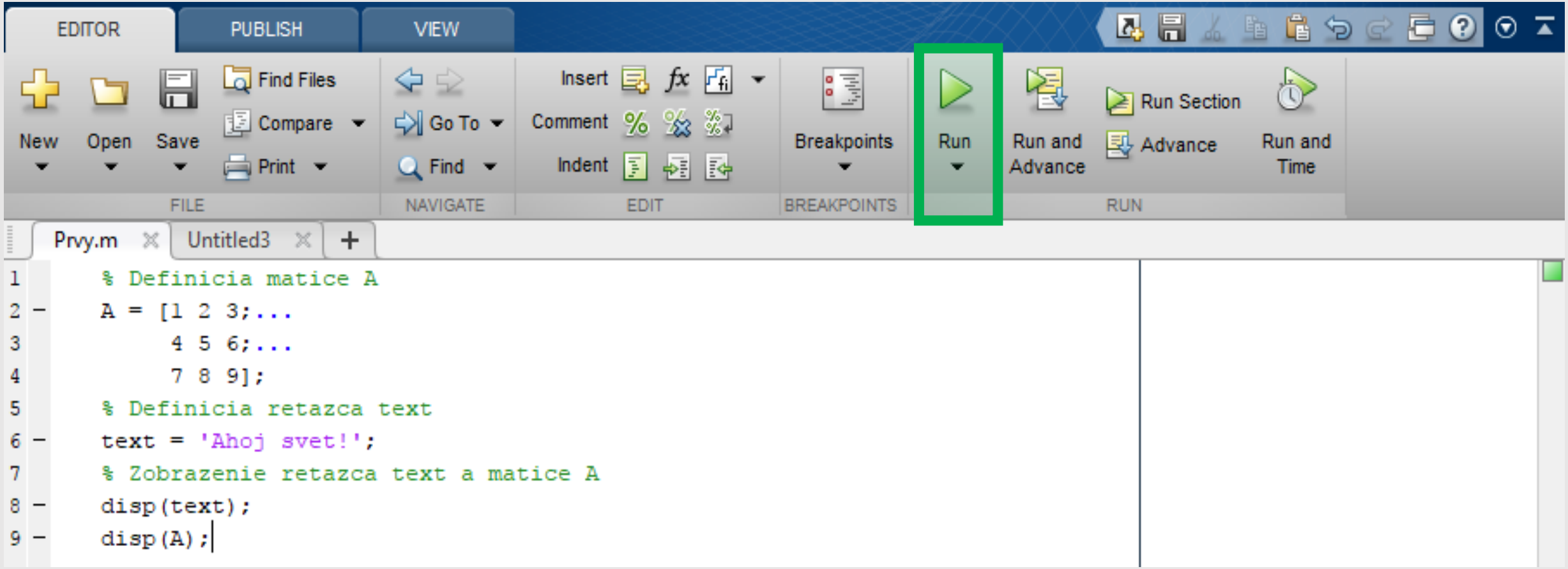
- ***Sekcií***
  - Sekcia začína takto: **%%** a predstavuje ucelenú časť kódu. **Sekcie sprehľadňujú kód.**
- ***Komentárov***
  - Komentár začína takto: **%**
- **Definícii premenných**
  - Premenné je vhodné definovať na začiatku skriptu.
- **Volaní funkcii resp. operácii s premennými**
- **Zobrazenia výsledku**
  - Výsledok sa zobrazuje podľa potreby. Nie vždy je potrebné výsledky zobrazovať.

Prvy.m

```
1     %% Vymazanie workspace a prikazoveho
2 -   clear;
3 -   clc;
4
5     %% Definicie premennych
6     % Definicia premennej c
7 -   c = 10;
8     % Definicia matice A
9 -   A = [1 2 3;...
10          4 5 6;...
11          7 8 9];
12    % Definicia retazca text
13 -  text = 'vysledok delenia je';
14    %% Hlavny vypocet
15 -  A = A/c;
16
17    %% Zobrazenie premennych
18    % Zobrazenie retazca text a matice A
19 -  disp(text);
20 -  disp(A);
```

Prvý skript

Skript sa spúšťa pomocou **zelenej šípky** v panely nástrojov:



Fakulta elektrotechniky
a informatiky

Úvod do Matlabu

Prednáška č. 3

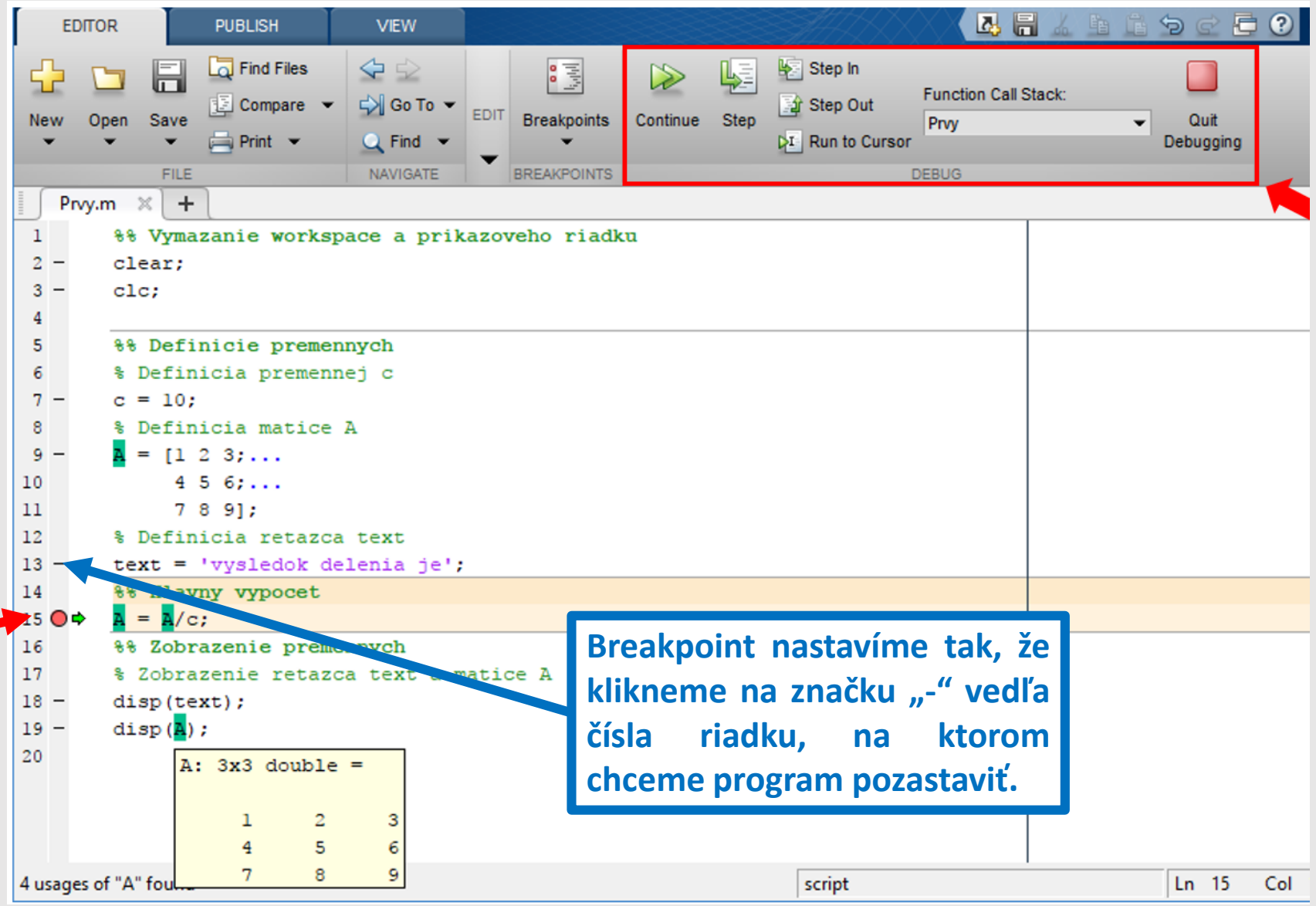
- Údajové typy (string, struct, cell, table)
- Príkazový riadok – základné príkazy
- Práca s priečinkom a súborom
- Prvý skript
- **Debugging**

Ing. Ondrej Kováč, PhD.

Základné okno - Debugging

Jednou zo základných činnosti programátora je okrem písania kódu aj jeho ladenie teda vyhľadávanie chýb. V angličtine sa stretávame s pojmom **debugging**, ktoré voľne možno preložiť ako **„odchrobačnenie“**.

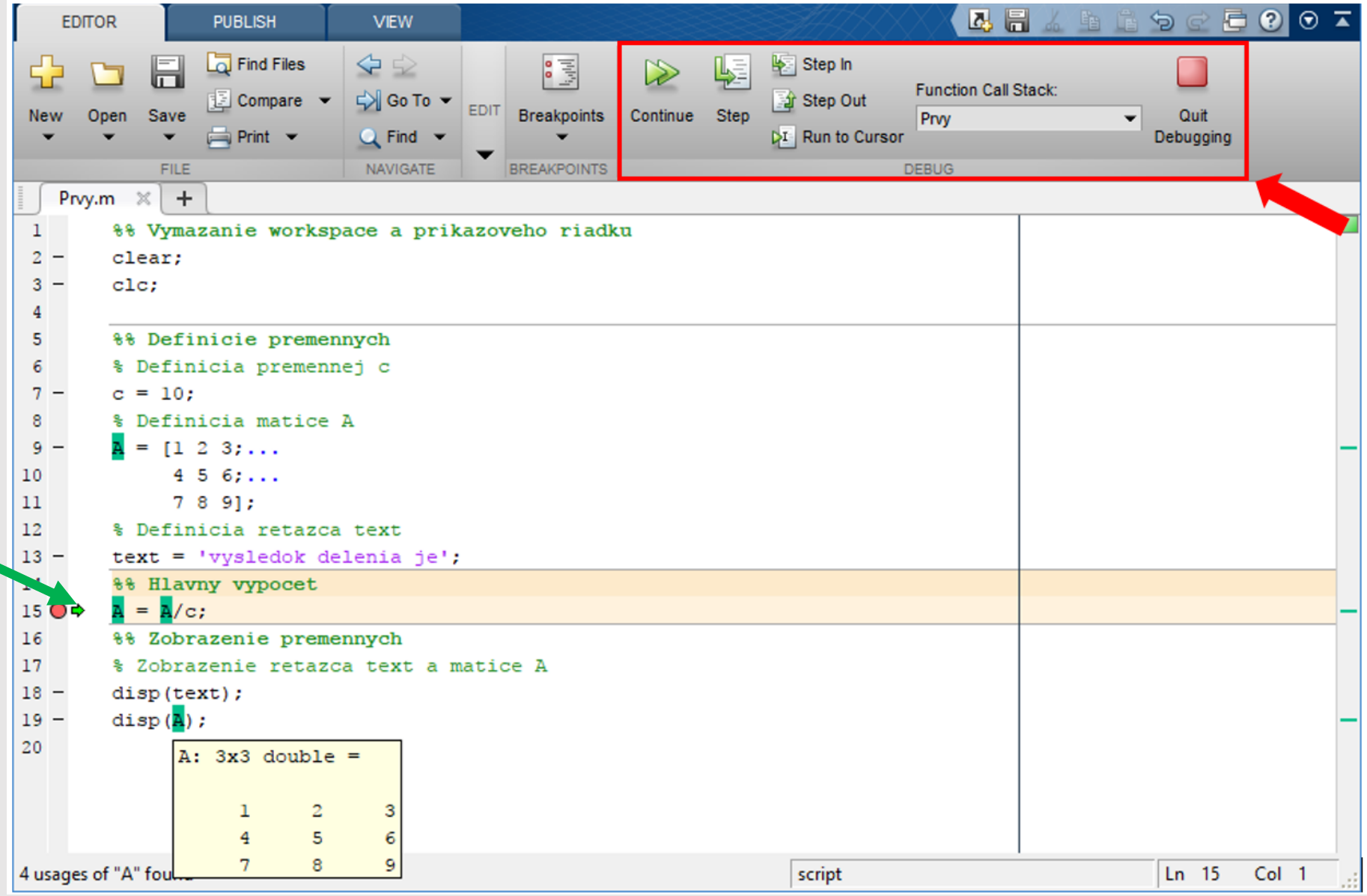
Základom ladenia je umiestnenie bodov pozastavenia (breakpoints).



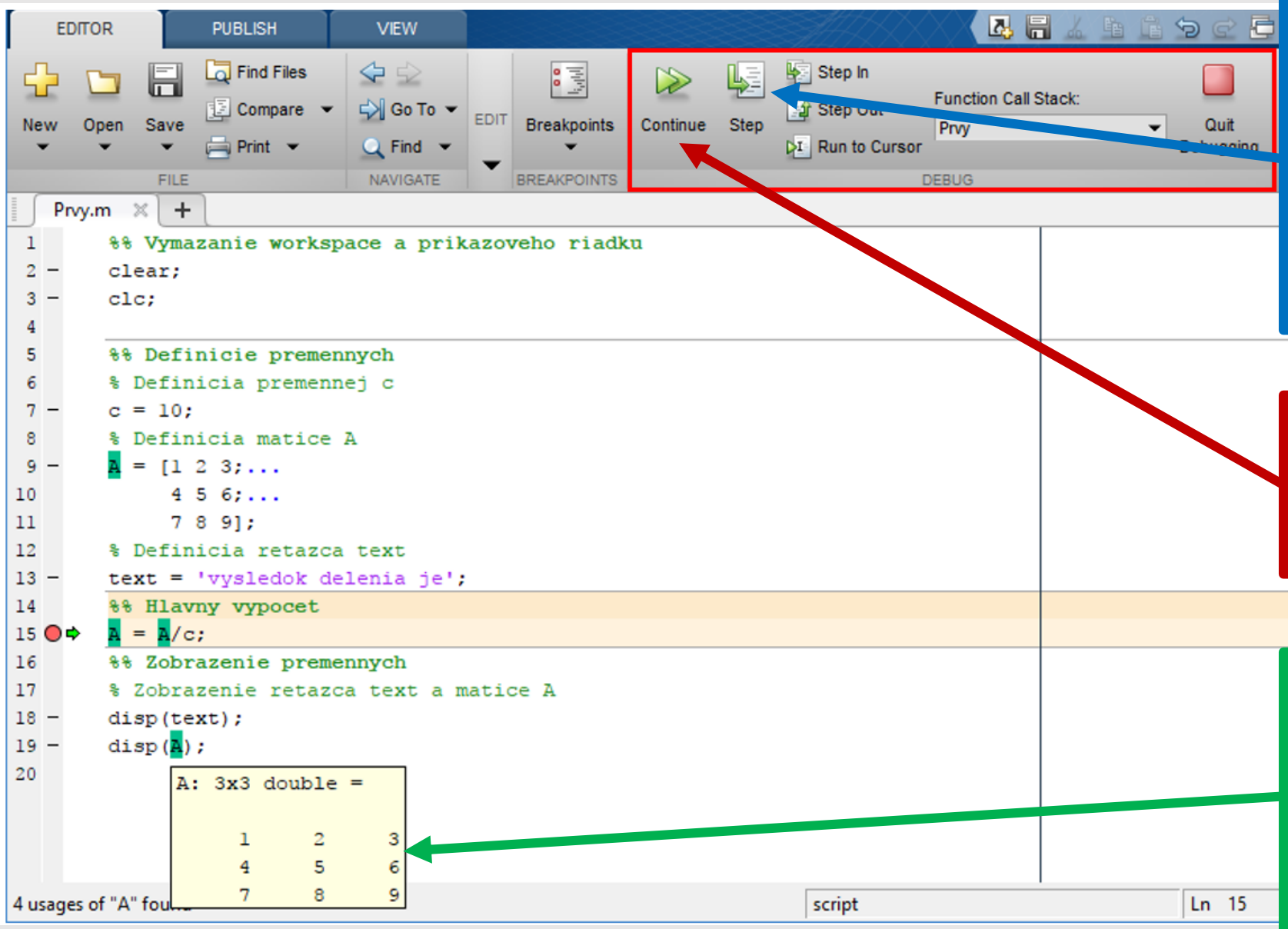
Breakpoint nastavíme tak, že klikneme na značku „-“ vedľa čísla riadku, na ktorom chceme program pozastaviť.

Základné okno - Debugging

Po spustení programu resp. skriptu sa jeho vykonávanie pozastaví v na danom mieste.
Toto je indikované zelenou šípkou



Základné okno - Debugging

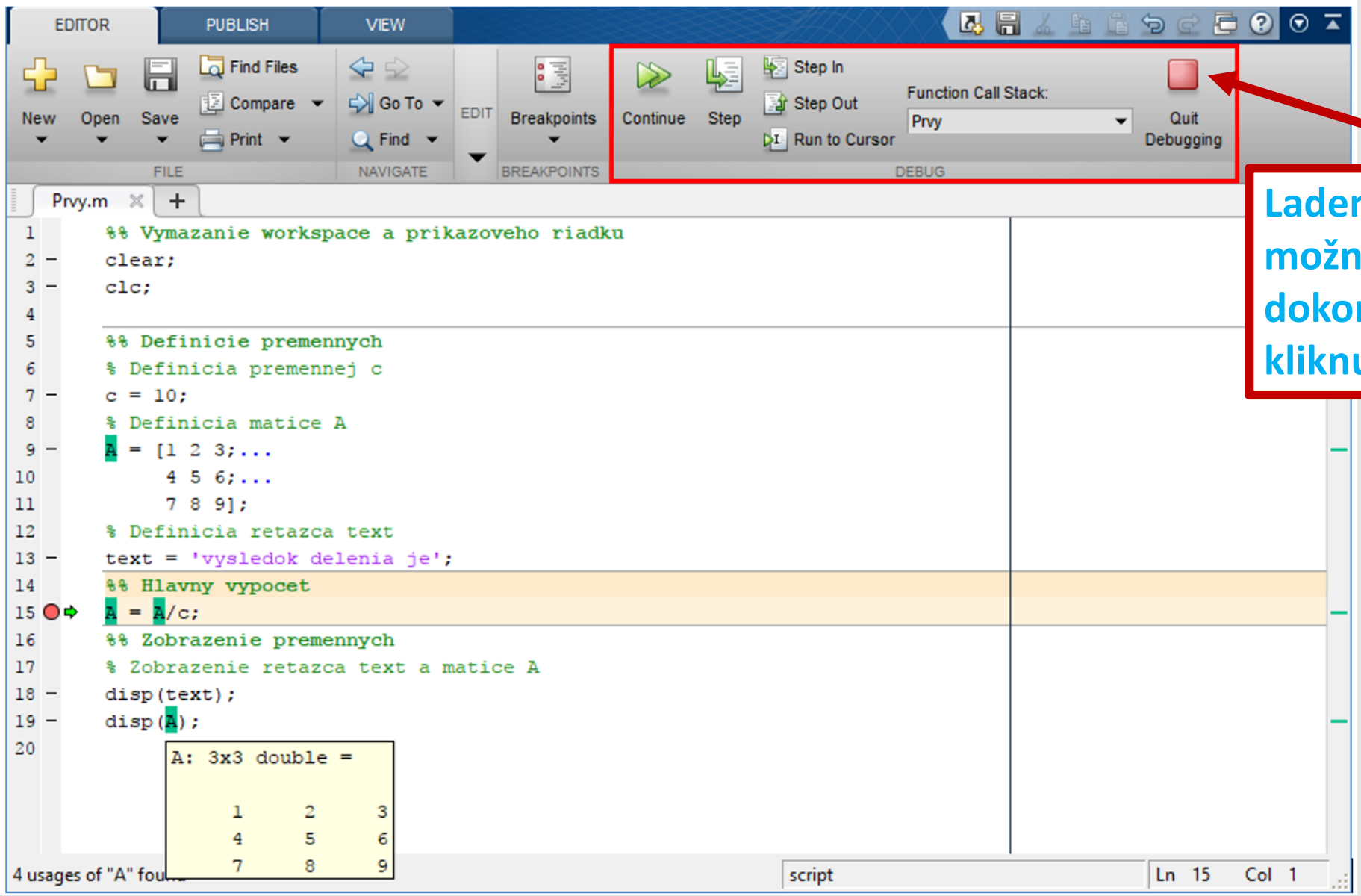


Pozastavený program je možné „krokovať“, teda postupne prechádzať riadky programu. K tomuto účelu slúži ikona step.

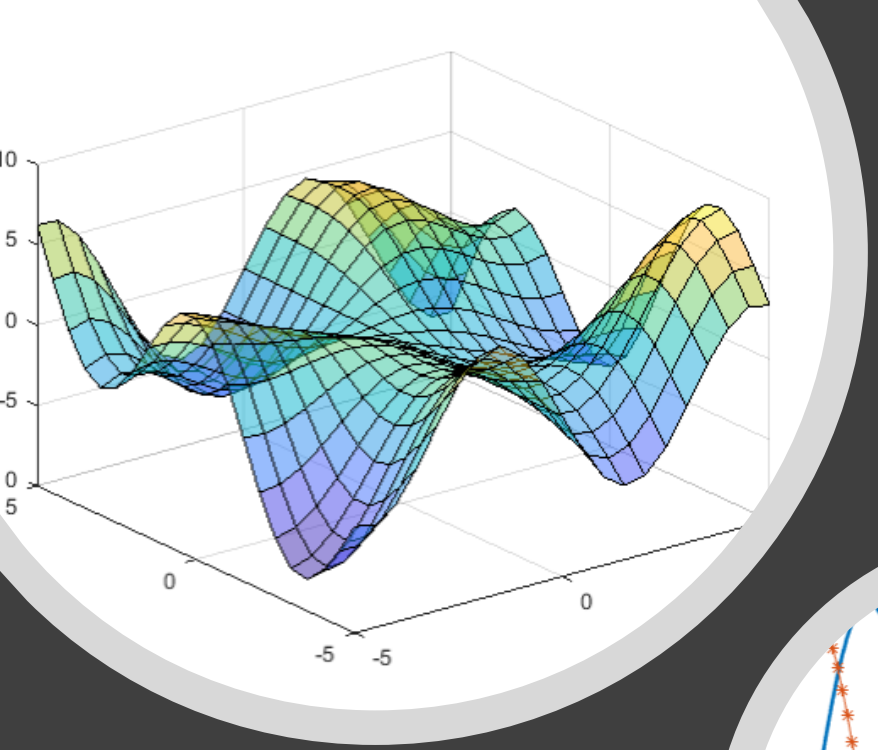
Beh programu možno obnoviť kliknutím na ikonu Continue

Zároveň je možné podržaním kurzoru myši na premennej zobraziť jej obsah. (pokiaľ nie je veľmi rozsiahla)

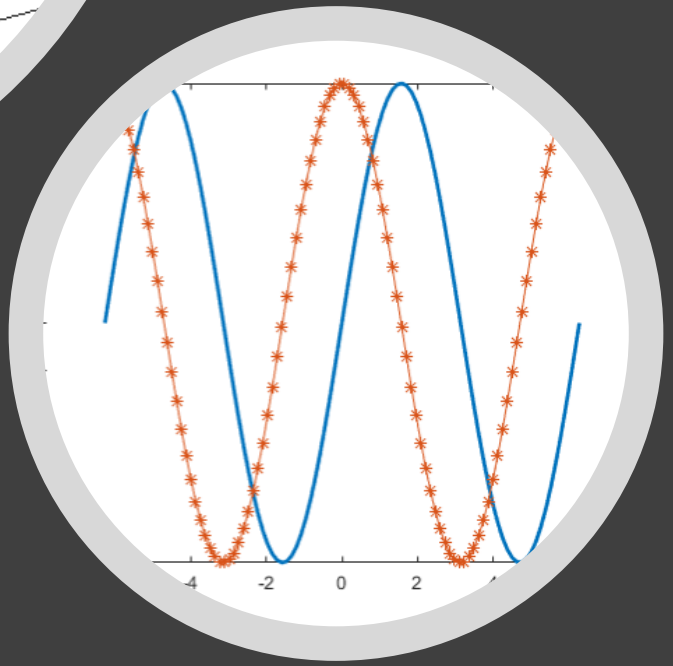
Základné okno - Debugging

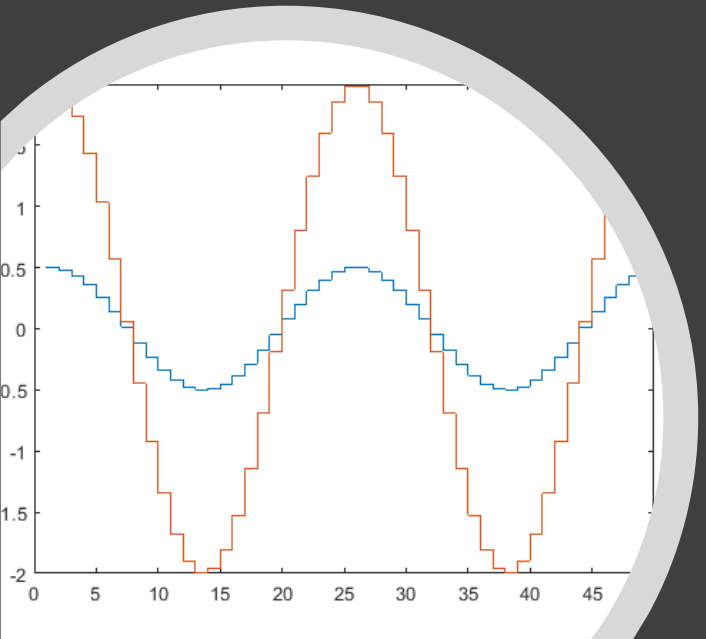


Ladenie programu je možné zrušiť (bez dokončenia programu) kliknutím na ikonu **Quit**



Nabudúce





- *Vizualizácia dát*
  - Okno „Figure“
- *Vizualizácia 1D a 2D dát*
  - Plot
  - Stem
  - Stairs
  - ...
- *Vizualizácia 3D dát*
  - Surf
  - Mesh
  - ...
- *Vizualizácia obrazových dát*
  - Imshow