

Seminář 2

Práce se skripty a live skripty

- více příkazů v jednom souboru
- může obsahovat komentáře

```
% komentar
```

- lze ho dělit na buňky a ty vyhodnocovat (spouštět) zvlášť (Run section, ctrl + enter)

```
%%
```

- v live skriptu hned vidíme výsledky, lze ho exportovat do pdf (různé layouty)
- v live skriptu je možné psát i text

Live skripty

Vytvořte live skript `seminar2.mlx`. V něm vytvořte proměnné `a` s hodnotou 5, `b` s hodnotou 3 a `c`, která má hodnotu součtu proměnných `a` a `b`.

```
a = 5
```

```
a = 5
```

```
b = 3
```

```
b = 3
```

```
c = a + b
```

```
c = 8
```

Vyzkoušejte si práci s editorem - různé layouty, psaní textu a kódu, dělení na sekce.

Kliknutím na výsledek se označí část kódu a naopak.

UI prvky

V live skriptech je možné přidat různé UI prvky - Controls.

Vytvořte celočíselnou proměnnou `a`, která může nabývat hodnot od -10 do 10 a její hodnota je nastavována pomocí slideru. Vytvořte proměnnou `b`, která je absolutní hodnotou `a`.

```
a = -8
```

```
a = -8
```

```
b = abs(a)
```

```
b = 8
```

Vyzkoušejte si i další UI prvky.

Chyby

V live skriptu se hned zobrazují chyby a varování - červené a oranžové pruhy u části, kde se chyba nachází.

```
%5a = 3  
a = 5  
a = 3
```

Práce s řetězci

Text je možné v matlabu uchovávat v poli znaků (character array) nebo řetězci (string).

Pole znaků - chová se a dá se s ním pracovat jako s vektory. Uvozuje se v ' '.

```
pole = 'ahoj'
```

```
pole =  
'ahoj'
```

```
pole(1)
```

```
ans =  
'a'
```

Řetězec - obsahuje celý text. Pro práci s řetězci existují vestavěné funkce. Zapisuje se do " ".

```
text = "ahoj"
```

```
text =  
"ahoj"
```

```
text(1)
```

```
ans =  
"ahoj"
```

Pozor na rozdílnou práci s řetězci a poli znaků!

```
pole2 = ['ahoj' ' ' 'svete']
```

```
pole2 =  
'ahoj svete'
```

```
text2 = ["ahoj" " " "svete"]
```

```
text2 = 1x3 string  
"ahoj" " " "svete"
```

```
text3 = append("ahoj ", "svete")
```

```
text3 =  
"ahoj svete"
```

```
text4 = "ahoj " + "svete"
```

```
text4 =  
"ahoj svete"
```

Tyto dva přístupy se dají použít i pro pole znaků.

Další zajímavé funkce pro práci s řetězci:

`strcat()`, `char()`, `num2str()`, `sprintf()`, `lower()`, `upper()`, `strcmp()`, `findstr()`, `strvcat()`, `ischar()`, ...

Pro více informací o těchto funkcích použijte nápovědu.

Práce s tabulkami

Tabulky jsou speciální datový typ. Jednotlivé sloupce mají název a mohou mít jiný datový typ.

```
T = table([1;2],["a";"b"], 'VariableNames', {'col1' 'col2'})
```

```
T = 2x2 table
```

	col1	col2
1	1	"a"
2	2	"b"

Přístup k jednotlivým hodnotám přes `.` operátor.

```
col1 = T.col1
```

```
col1 = 2x1  
1  
2
```

Přidání nového sloupce

```
T.col3 = [0; 0]
```

```
T = 2x3 table
```

	col1	col2	col3
1	1	"a"	0
2	2	"b"	0

Řazení hodnot v tabulce podle sloupce

```
T = sortrows(T, 'col1', 'descend')
```

Přímo v náhledu je nápověda.

Filtrování hodnot v tabulce

```
T = T(T.col2 == "a",:)
```

Nápověda je v náhledu.

Počet řádků a sloupců tabulky je možné získat pomocí `height()` a `width()`

```
height(T)
```

```
ans = 2
```

```
width(T)
```

```
ans = 3
```

Další zajímavé funkce pro práci s tabulkou

`summary()`, `head()`, `tail()`, ...

Kombinace více tabulek

`join()`, `innerjoin()`, `outerjoin()`, `union()`, `intersection()`, ...

Načtení ze souboru (.csv)

```
T = readtable('data.csv')
```

T = 45x5 table

	Name	LastName	Age	Height	Weight
1	'John'	'Doe'	30	180	75
2	'Jane'	'Doe'	28	167	60
3	'Bob'	'Smith'	35	190	85
4	'Emma'	'Johnson'	25	162	55
5	'Michael'	'Brown'	40	185	80
6	'Emily'	'Davis'	27	170	65
7	'William'	'Miller'	32	178	70
8	'Ava'	'Wilson'	29	165	58
9	'James'	'Anderson'	36	193	90
10	'Olivia'	'Thomas'	26	161	53
11	'Christopher'	'Jackson'	33	187	78
12	'Sophia'	'White'	24	160	50
13	'David'	'Harris'	37	195	92
14	'Charlotte'	'Martin'	31	163	57

⋮

Případně 2x poklepat na název souboru (otevře se import tool) nebo tlačítko import.

Uložení tabulky

```
writetable(T, 'nazev.csv')
```

.txt, .dat, .csv - data uložena oddělené čárkou

.xls, .xlsm, .xlsx - Excel

.xml - XML soubor

Další funkce pro práci s vektory (maticemi), které je dobré znát.

sum(), unique()

Co vrátí následující příkaz?

```
unique(T.Age)
```

```
ans = 17x1
    23
    24
    25
    26
    27
    28
    29
    30
    31
    32
     ⋮
```