
Listy (seznamy)

Nadefinujme následující listy (seznamy):

```
basaPiv = {Gambrinus, Plzen, Staropramen}
```

```
basaPiv2 = {Bernard, Svijany, Rohozec}
```

```
hromadaPiv = {basaPiv, basaPiv, basaPiv2, Starobrnno, Radegast}
```

```
skladPiv = {hromadaPiv, basaPiv2}
```

```
{Gambrinus, Plzen, Staropramen}
```

```
{Bernard, Svijany, Rohozec}
```

```
{ {Gambrinus, Plzen, Staropramen}, {Gambrinus, Plzen, Staropramen}, {Bernard, Svijany, Rohozec}, Starobrnno, Radegast}
```

```
{ { {Gambrinus, Plzen, Staropramen}, {Gambrinus, Plzen, Staropramen}, {Bernard, Svijany, Rohozec}, Starobrnno, Radegast},  
  {Bernard, Svijany, Rohozec}}
```

◀ | ▶

Přístup k prvkům seznamu (indexace)

K jednotlivým položkám listu lze přistupovat pomocí [[]] (po programátorsku: indexy polí se píšou do dvojitéch hranatých závorek)

jednotlivým K listu lze přistupovat po položkám pomocí (programátorsku : do dvojitéch hranatých indexy píšou polí se závorek)

```
basaPiv[[2]]
```

```
Plzen
```

```
hromadaPiv[[3]]
```

```
{Bernard, Svijany, Rohozec}
```

```
hromadaPiv[[3, 2]]
```

```
Svijany
```

```
skladPiv[[1]]
```

```
{ {Gambrinus, Plzen, Staropramen}, {Gambrinus, Plzen, Staropramen}, {Bernard, Svijany, Rohozec}, Starobrnno, Radegast }
```

```
skladPiv[[1, 2, 3]]
```

```
Staropramen
```

Funkce operující nad listy

? Flatten

Flatten[*list*] flattens out nested lists.

Flatten[*list*, *n*] flattens to level *n*.

Flatten[*list*, *n*, *h*] flattens subexpressions with head *h*.

Flatten[*list*, {{*s*₁₁, *s*₁₂, ...}, {*s*₂₁, *s*₂₂, ...}, ...}] flattens *list* by combining all levels *s*_{*ij*} to make each level *i* in the result. >>

Flatten[skladPiv]

```
{Gambrinus, Plzen, Staropramen, Gambrinus, Plzen, Staropramen, Bernard, Svijany, Rohozec, Starobrnno, Radegast, Bernard, Svijany, Rohozec}
```

Funkce generující listy

? Range

`Range[i_{max}]` generates the list $\{1, 2, \dots, i_{max}\}$.

`Range[i_{min}, i_{max}]` generates the list $\{i_{min}, \dots, i_{max}\}$.

`Range[i_{min}, i_{max}, di]` uses step di . >>

```
Range[3, 30]
```

```
Range[3, 30, 6]
```

```
{3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30}
```

```
{3, 9, 15, 21, 27}
```

? Table

`Table[$expr, \{i_{max}\}$]` generates a list of i_{max} copies of $expr$.

`Table[$expr, \{i, i_{max}\}$]` generates a list of the values of $expr$ when i runs from 1 to i_{max} .

`Table[$expr, \{i, i_{min}, i_{max}\}$]` starts with $i = i_{min}$.

`Table[$expr, \{i, i_{min}, i_{max}, di\}$]` uses steps di .

`Table[$expr, \{i, \{i_1, i_2, \dots\}\}$]` uses the successive values i_1, i_2, \dots .

`Table[$expr, \{i, i_{min}, i_{max}\}, \{j, j_{min}, j_{max}\}, \dots]$` gives a nested list. The list associated with i is outermost. >>

```
Table[x2, {x, 0, 10}]
```

```
{0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100}
```

Proměnné - rozšířené možnosti

Hodnotou proměnné může být nejenom číslo, ale prakticky jakýkoliv výraz či řetězec. Například:

```
x = 5
budova = NTK
5
NTK
```

Proměnné lze proto využít například pro pojmenování složitých výrazů, například rovnic. Soustavu lineárních rovnic

$$\begin{aligned} 2x_1 + 3x_2 + 4.2x_3 &= 5 \\ 3x_1 + 15.2x_2 + 8.7x_3 &= 80.1 \\ 4x_1 + 8.9x_2 + 17.2x_3 &= 11 \end{aligned}$$

lze proto řešit buď takto:

```
Solve[{2 x1 + 3 x2 + 4.2 x3 == 5, 3 x1 + 15.2 x2 + 8.7 x3 == 80.1, 4 x1 + 8.9 x2 + 17.2 x3 == 11}]
```

$$\{\{5_1 \rightarrow -3.63596, 5_2 \rightarrow 7.29908, 5_3 \rightarrow -2.29174\}\}$$

nebo takto :

```
rce1 = 2 x1 + 3 x2 + 4.2 x3 == 5;
rce2 = 3 x1 + 15.2 x2 + 8.7 x3 == 80.1;
rce3 = 4 x1 + 8.9 x2 + 17.2 x3 == 11;
Solve[{rce1, rce2, rce3}]
```

$$\{\{5_1 \rightarrow -3.63596, 5_2 \rightarrow 7.29908, 5_3 \rightarrow -2.29174\}\}$$

Pravidla

```
ClearAll[x, y, z]

Pravidla
def = {x → 2, y → 1, z → 4}
moje = {x → 5}
{x, y, z} /. moje
({x, y, z} /. moje) /. def

Pravidla

{x → 2, y → 1, z → 4}

{x → 5}

{5, y, z}

{5, 1, 4}
```

"→" se napíše tak, že zadáte - následované > a pak pokračujete v psaní

Aplikace pravidel pomocí /.

```
{x, y, z} /. moje
({x, y, z} /. moje) /. def

{5, y, z}

{5, 1, 4}
```

Uživatelem definované funkce

Pro definování se používá dvojznak `:=`

```
mojeFce[x_] := x3 + 2 x2 + 5 x - 27
```

Formální parametry jsou na levé straně označeny podtržítkem (`x_`, `xx_`, `y_`), na pravé straně podtržítka nemají. Uživatelem definovanou funkci lze následně použít jako jakoukoliv jinou funkci.

Například:

```
Plot[mojeFce[r], {r, -3, 3}]
```

