

Listy - struktury

Lista jest strukturą zawierającą dane w postaci łańcuchów znakowych. Istnieją trzy rodzaje list: lista zwykła - `list`, lista o strukturze wektorowej - `tlist`, lista o strukturze macierzowej `mlist`.

Lista zwykła `list()`

Lista zwykła jest zbiorem obiektów dostępnych w Scilabie. Podobnie jak w Matlabie obiekty nie muszą być tego samego rodzaju. Mogą więc być skalarami, wektorami, macierzami, łańcuchami znakowymi, funkcjami i wszystkimi możliwymi ich kombinacjami. Przykład listy pokazano poniżej. Poleceniem kreującym listę jest `list()`.

```
lista=list('test',[1,2;3 4],...
['To jest przykład'; 'różnych danych'])
lista =
      lista(1)
test
      lista(2)
!  1.    2. !
!  3.    4. !
      lista(3)
!To jest przykład !
!                   !
!roznych danych   !
```

Dostęp do poszczególnych elementów listy uzyskujemy poprzez indeksy.

```
lista(1)
ans=
test
```

Dostęp do dowolnego elementu macierzy na przykład z pierwszego wiersza i drugiej kolumny, która jest elementem tej listy, uzyskujemy bardzo prosto- wręcz intuicyjnie.

```
lista(1)(1,2)
ans=
2
```

Chcąc dopisać coś na początek listy, wykorzystujemy indeks zerowy. Oczywiście wszystkie obiekty przesuwać się o jedną pozycję do przodu.

```

lista(0)=%eps;
lista(2)
ans
test

```

Dopisanie można zrobić też w nieco inny sposób, zachowanie kolejności nowych elementów nie jest konieczne, brakujące pozostaną niezdefiniowane.

```

lista(8)='koniec'
lista =
    lista(1)
    2.220D-16
    lista(2)
test
    lista(3)
!  1.    2. !
!  3.    4. !
    lista(4)
!To jest przykład !
!                !
!roznych danych  !
    lista(5)
    Undefined
[More (y or n ) ?]
    lista(6)
    Undefined
    lista(7)
    Undefined
    lista(8)
koniec

```

Usuwanie z listy może odbywać się po jednym elemencie, wpisaniem na odpowiednią pozycję `null()`.

```

a=list(1,2,3,4,5)
a =
    a(1)
    1.
    a(2)

```

```

2.
  a(3)
3.
  a(4)
4.
  a(5)
5.

a(3)=null(); // usuniecie elementu 3
a           // wydruk nowej listy
a =
  a(1)
1.
  a(2)
2.
  a(3)
4.
  a(4)
5.

```

Jak widać brakuje 3, lista oczywiście przenieumerowała automatycznie indeksy. Stosunkowo łatwo jest wydobyć pewne elementy listy i przypisać je pod nowe zmienne.

```

[u,v]=a(2:3) v =
  4.
u =
  2.

```

Istnieją funkcje do pracy na listach, ich nazwy mówią dużo o przeznaczeniu: `size()`, `length()`.

Lista typów `tlist()`

Lista typów inicjowana poleceniem `tlist()` jest jej specjalnym rodzajem listy. Pierwszy elementem tej listy jest zawsze łańcuch lub wektor łańcuchów.

```

Tlista=tlist(['przyklad', 'pierwszy', 'drugi'], 1.23, [1,2], ' pozostałe')
Tlista =
  Tlista(1)

```

```
!przykład pierwszy drugi !
  Tlista(2)
  1.23
  Tlista(3)
! 1. 2. !
  Tlista(4)
  pozostałe
```

Dostęp do elementów listy uzyskujemy w zwykły sposób poprzez jej indeksy, ale jest również możliwy inny sposób, poprzez nazwy zawarte w pierwszym obiekcie listy (za wyjątkiem pierwszego łańcucha). Oczywiście dostęp będzie możliwy tylko do dwu pierwszych elementów listy.

```
Tlista.pierwszy
ans =
  1.23
Tlista.drugi
ans =
! 1. 2. !
Tlista.przykład
  |--error 4
```

Listy macierzowe

Ostatnim rodzajem listy jest lista macierzowa inicjowana poleceniem `mlist()`. Konstrukcja tej listy jest zbliżona do listy typów, ale nie jest tu możliwy dostęp poprzez indeksy, jedynym sposobem jest dostęp poprzez nazwę wymienioną w pierwszym łańcuchu tworzonej listy.

```
M=mlist(['V','name','value'],['a','b','c'],[1 2 3])
// odwołanie się elementu
M.name
ans =
!a b c !

//odwołanie się do elementu
M('value')
ans =
! 1. 2. 3. !
```