

# Rozšíření o pole a tabulky

Tato rozšíření přidají do NetLoga dvě nové datové struktury – pole a hashovací tabulky.

## Použití

Na cokoliv, co děláte s poli a tabulkami, můžete v podstatě použít i seznamy. Tabulky a pole však umožňují zvýšit rychlost. Všechny tři datové struktury (seznam, pole a tabulka) mají co do výkonu rozdílné vlastnosti, takže pokud zvolíte vhodnou datovou strukturu, poběží model rychleji.

Pole jsou vhodná v případě, že potřebujete soubor hodnot, který má pevně danou velikost. Znáte-li pozici položky, můžete si ji velmi rychle přečíst nebo změnit.

Tabulky se hodí, když potřebujete přiřadit hodnoty k jiným hodnotám. Chcete například vytvořit tabulku slov s jejich definicemi. Potom můžete vyhledat definici každého slova. Slova tu fungují jako „klíče“. Velmi jednoduše tak můžete vyvolat hodnotu jakéhokoliv klíče v tabulce, ale ne obráceně.

## Způsob použití

Obě rozšíření jsou již předinstalovaná.

Rozšíření o pole do modelu přidáte tak, že do horního řádku v panelu **Procedures** napíšete:

```
extensions [array]
```

U rozšíření o hashovací tabulky na stejné místo napíšete:

```
extensions [table]
```

Můžete použít také obě rozšíření zároveň, a to následujícím způsobem:

```
extensions [array table]
```

Pokud model používá nějaká jiná rozšíření, obsahuje už řádek extensions, takže stačí do seznamu přidat `array` a/nebo `table`.

Více informací o použití rozšíření v NetLogu najdete v kapitole [Průvodce rozšířeními](#).

## Omezení u klíčů v tabulce

Klíče v tabulce smějí být pouze řetězce, čísla, booleovské hodnoty či seznamy. (Seznamy mohou obsahovat vnořené seznamy, pokud i všechny jejich položky jsou řetězce, čísla či booleovské hodnoty.)

## Ukázka pole

```
let a array:from-list n-values 5 [0]
print a
=> {{array: 0 0 0 0 0}}
print array:length a
=> 5
foreach n-values 5 [?] [ array:set a ? ? * ? ]
print a

=> {{array: 0 1 4 9 16}}
print array:item a 0
=> 0
print array:item a 3
=> 9
array:set a 3 50
print a
=> {{array: 0 1 4 50 16}}
```

## Ukázka tabulky

```
let dict table:make
table:put dict "turtle" "cute"
table:put dict "bunny" "cutest"
print dict
=> {{table: "turtle" -> "cute", "bunny" -> "cutest" }}
print table:length dict
=> 2
print table:get dict "turtle"
=> "cute"
print table:get dict "leopard"
=> 0
print table:keys dict
=> ["turtle" "bunny"]
```

## Známé problémy

Když exportujete svět NetLoga (pomocí příkazu export-world nebo přes položku menu Export World), exportují se pole a tabulky „podle hodnoty“. To znamená, že pokud jste měli stejné pole nebo stejnou tabulku na více místech a toto pole či tuto tabulku jste pak exportovali a znovu importovali, budou se na místech, kde dříve bylo původní pole či původní tabulka, nacházet rozdílné verze. Tyto duplikáty sice budou zpočátku obsahovat stejné hodnoty, ale pokud změníte něco v jednom duplikátu, v ostatních se změna neprojeví.

## Primitiva polí

array:from-list array:item array:set array:length array:to-list

### **array:from-list**

**array:from-list** *list*

Vrací nové pole obsahující stejné položky ve stejném pořadí jako vstupní seznam.

### **array:item**

**array:item** *array index*

Vrací položku v daném poli s daným indexem (v rozmezí od nuly do délky pole minus jedna).

### **array:set**

**array:set** *array index value*

Nastaví položku v daném poli s daným indexem (v rozmezí od nuly do délky pole minus jedna) na danou hodnotu.

Na rozdíl od primitiva replace-item, určeného pro seznamy, není vytvořeno nové pole, ale pouze se změní to vybrané.

**array:length****array:length array**

Vrací délku daného pole, tj. počet položek v poli.

**array:to-list****array:to-list array**

Vrací nový seznam obsahující stejné položky ve stejném pořadí jako v daném poli.

**Primitiva tabulek**

table:clear   table:from-list   table:get   table:has-key?   table:keys  
table:length   table:make   table:put   table:remove   table:to-list

**table:clear****table:clear table**

Odstraní z tabulky všechny dvojice klíč-hodnota.

**table:from-list****table:from-list list**

Vrací novou tabulku s obsahem seznamu. Seznam musí obsahovat dva seznamy prvků či dvojice. První člen dvojice je klíč, druhý člen je hodnota.

**table:get****table:get table key**

Vrací hodnotu, která je v tabulce přiřazena ke klíči. Způsobí chybu v případě, že pro klíč není v tabulce nalezen vstup.

## **table:has-key?**

**table:has-key? *table key***

Vrací pravdivou hodnotu, pokud je pro klíč v tabulce nalezen vstup.

## **table:keys**

**table:keys *table***

Vrací seznam všech klíčů v tabulce.

## **table:length**

**table:length *table***

Vrací počet vstupů v tabulce.

## **table:make**

**table:make**

Vrací novou, prázdnou tabulku.

## **table:put**

**table:put *table key value***

Přiřadí klíč k hodnotě. Pokud už v tabulce k danému klíči existuje nějaký vstup, je nahrazen.

## **table:remove**

**table:remove *table key***

Odstraní přiřazení ke klíči.

## **table:to-list**

### **table:to-list** *table*

Vrací seznam s obsahem tabulky. Seznam budou tvořit seznamy prvků – dvojic. První člen dvojice je klíč, druhý člen je hodnota.

Copyright 1999-2009 by Uri Wilensky.  
Všechna práva vyhrazena.

Aplikace NetLogo, modely i dokumentace jsou šířeny veřejnosti zdarma pro účel tvorby a studia modelů. Software, modely a dokumentaci je možné pro studijní a výzkumné účely používat a měnit, a to za podmínky, že je výsledný produkt nabízen bezplatně a s uvedením informace o autorských právech a jménem původce na všech kopiích a související dokumentaci.

Pro jiné využití - než jsou výše zmíněné nekomerční způsoby - celku i jednotlivých částí (a to jak v původní, nebo změněné podobě) je třeba předem požádat o svolení od Uri Wilensky. Software, modely ani dokumentace nesmějí být užívány, přepisovány, ani upravovány jako součást komerčního softwaru nebo hardwaru bez předchozího získání licence od Uri Wilensky. Nezaručujeme kompatibilitu tohoto systému s jakýmkoliv jiným systémem a neposkytujeme žádné záruky.

Pro účely citování v akademických publikacích používejte tento odkaz:  
Wilensky, U. (1999). NetLogo. <http://ccl.northwestern.edu/netlogo>. Center for Connected Learning and Computer-Based Modeling. Northwestern University, Evanston, IL.