

Основне академске студије  
Информациони инжењеринг

Основи рачунарске интелигенције

# Рад с низовима у језику Python

(материјали за вежбе)

## Библиотека *NumPy* (*Numerical Python*)

посвећена раду над низовима вредности

хомогеност елемената низа

подршка за низове разних димензионалности

вектор (једнодимензионални низ)

матрица (дводимензионални низ)

...

подршка за разне операције над низовима

## Библиотека *NumPy*

### ОСНОВНИ ПОДАЦИ

покретач Травис Олифант

настанак 2005. године

слободна за употребу

актуелна верзија 1.22.0

Интернет сајт

<https://numpy.org/>

## Припреме за коришћење

### инсталација библиотеке

библиотека доступна у дистрибуцији *Anaconda*

могућа засебна инсталација

коришћење *pip*

коришћење *conda*

### учитавање библиотеке

уобичајени начин учитавања

```
import numpy as np
```

коришћење псеудонима **np**

## Основне радње у вези с низовима

формирање низа

испитивање особина низа

преуређење низа

приступање низу

измена низа

математичке и статистичке операције над низом

сортирање низа

претраживање низа

...

## Формирање низа

тип низа

класа **`np.ndarray`**

облик низа

облик је описан торком ненегативних целобројних вредности  
свака компонента описује величину по једној од димензија

тип елемената низа

тип описан инстанцом класе **`np.dtype`**

примери припремљених типова

**`np.byte`, `np.short`, `np.intc`, `np.longlong`, `np.int8`, `np.int16`, `np.int32`,  
`np.int64`, `np.float16`, `np.float32`, `np.float64`, `np.complex64`,  
`np.complex128`...**

очекивани тип

**`np.float64`**

## Формирање низа

константе

**pr.e**

Ојлерова константа

**pr.inf**

плус бесконачно

**pr.nan**

није број

**pr.pi**

број пи

...

## Формирање низа

функција **`np.array(...)`**

формирање на основу датих података

функција **`np.empty(...)`**

формирање без конкретне иницијализације садржаја

функција **`np.zeros(...)`**

попуњавање нулама

функција **`np.ones(...)`**

попуњавање јединицама

функција **`np.full(...)`**

попуњавање задатом вредношћу или задатим вредностима

функција **`np.arange(...)`**

попуњавање задатим распоном вредности

...

## Формирање низа

```
np.array([1, 3, 5, 7])
```

```
[1 3 5 7]
```

```
np.array([1, 3, 5, 7], np.float128)
```

```
[1. 3. 5. 7.]
```

```
np.array([[1, 3], [5, 7]])
```

```
[[1 3]
```

```
[5 7]]
```

## Формирање низа

```
np.empty((2, 2))
```

```
[[4.9e-324 1.5e-323]  
 [2.5e-323 3.5e-323]]
```

```
np.zeros((2, 2))
```

```
[[0. 0.]  
 [0. 0.]]
```

## Формирање низа

```
np.ones((3, 2))
```

```
[[1.  1.]  
 [1.  1.]  
 [1.  1.]
```

```
np.full((2, 3), 5)
```

```
[[5 5 5]  
 [5 5 5]]
```

```
np.full((1, 2, 3), 7)
```

```
[[[7 7 7]  
   [7 7 7]]]
```

## Формирање низа

**пр. `arange(2)`**

`[0 1]`

**пр. `arange(1, 5)`**

`[1 2 3 4]`

**пр. `arange(1, 10, 3)`**

`[1 4 7]`

## Испитивање особина низа

особина **shape**

облик низа (величина низа по димензијама)

особина **ndim**

број димензија код низа

особина **size**

број елемената у низу

...

Испитивање особина низа

```
np.array([[4, 1, 2], [9, 7, 3]]).shape  
(2, 3)
```

```
np.zeros((5, 5)).ndim  
2
```

```
np.ones((3, 2)).size  
6
```

## Преуређење низа

функција **`np.reshape(...)`**

промена облика низа уз задржавање постојећих елемената

функција **`np.resize(...)`**

успостављање новог облика

функција **`np.flip(...)`**

обртање редоследа елемената

функција **`np.ravel(...)`**

развој у једној димензији

...

## Преуређење низа

```
np.reshape(np.array([1, 3, 5, 7]), (2, 2))
```

```
[[1 3]  
 [5 7]]
```

```
np.reshape(np.full((2, 3), 5), (3, 2))
```

```
[[5 5]  
 [5 5]  
 [5 5]]
```

## Преуређење низа

```
np.resize(np.array([1, 3, 5, 7]), (2, 1))
```

```
[[1]  
 [3]]
```

```
np.resize(np.full((2, 3), 5), (3, 3))
```

```
[[5 5 5]  
 [5 5 5]  
 [5 5 5]]
```

## Преуређење низа

```
np.flip(np.array([[1, 3], [5, 7]]))
```

```
[[7 5]
```

```
[3 1]]
```

```
np.ravel(np.array([[1, 3], [5, 7]]))
```

```
[1 3 5 7]
```

## Приступање низу

функција **print(...)**

приказ низа или одабраних елемената

оператор **[ ]**

приступање по позицији

путем појединачних вредности

путем више вредности

приступање по димензијама

по једној димензији

по више димензија

приступање путем вредности одређеног типа

путем бројчаних вредности

путем логичких вредности

...

## Пристапање низу

```
a = np.array([[1, 3, 5], [7, 9, 11]])
```

```
[[ 1  3  5]  
 [ 7  9 11]]
```

```
a[1]
```

```
[ 7  9 11]
```

```
a[:, 1]
```

```
[3 9]
```

```
a[1, 1]
```

```
9
```

## Пристапање низу

```
a = np.array([[1, 3, 5], [2, 4, 8], [6, 7, 9]])
```

```
[[1 3 5]  
 [2 4 8]  
 [6 7 9]]
```

```
a[1:3]
```

```
[[2 4 8]  
 [6 7 9]]
```

```
a[:2, 1:3]
```

```
[[3 5]  
 [4 8]]
```

## Пристапање низу

```
a = np.array([1, 4, 2, 8, 6])
```

```
[1 4 2 8 6]
```

```
a[np.array([2, 1, 3])] 
```

```
[2 4 8]
```

```
a[np.array([4, 2, 0, 2, 4])] 
```

```
[6 2 1 2 6]
```

## Пристапање низу

```
a = np.array([[7, 8, 9], [4, 6, 3], [2, 5, 1]])
```

```
[[7 8 9]  
 [4 6 3]  
 [2 5 1]]
```

```
a < 4.5
```

```
[[False False False]  
 [ True False  True]  
 [ True False  True]]
```

```
a[a < 4.5]
```

```
[4 3 2 1]
```

## Измена низа

оператор `[]` с леве стране оператора `=`  
приступање елементима низа и њихова измена

функција `np.insert(...)`

уметање елемената, без измене изворног низа

функција `np.append(...)`

додавање елемената на крај, без измене изворног низа

функција `np.delete(...)`

уклањање елемената, без измене изворног низа

...

## Измена низа

```
a = np.array([[7, 8, 2, 9], [5, 4, 6, 3]])
```

```
[[7 8 2 9]  
 [5 4 6 3]]
```

```
a[0, 2] = 0
```

```
[[7 8 0 9]  
 [5 4 6 3]]
```

```
a[1:, :2] = -1
```

```
[[ 7  8  0  9]  
 [-1 -1  6  3]]
```

## Измена низа

```
a = np.array([[7, 8, 9], [4, 6, 3], [2, 5, 1]])
```

```
[[7 8 9]  
 [4 6 3]  
 [2 5 1]]
```

```
np.insert(a, 2, -3)
```

```
[ 7  8 -3  9  4  6  3  2  5  1]
```

```
np.insert(a, 2, 0, axis=0)
```

```
[[7 8 9]  
 [4 6 3]  
 [0 0 0]  
 [2 5 1]]
```

## Измена низа

```
a = np.array([[7, 8, 9], [4, 6, 3], [2, 5, 1]])
```

```
[[7 8 9]  
 [4 6 3]  
 [2 5 1]]
```

```
np.append(a, [[10, 11, 12]], axis=0)
```

```
[[ 7  8  9]  
 [ 4  6  3]  
 [ 2  5  1]  
 [10 11 12]]
```

## Измена низа

```
a = np.array([[7, 8, 9], [4, 6, 3], [2, 5, 1]])
```

```
[[7 8 9]  
 [4 6 3]  
 [2 5 1]]
```

```
np.append(a, [[10], [11], [12]], axis=1)
```

```
[[ 7  8  9 10]  
 [ 4  6  3 11]  
 [ 2  5  1 12]]
```

## Измена низа

```
a = np.array([[7, 8, 9], [4, 6, 3], [2, 5, 1]])
```

```
[[7 8 9]  
 [4 6 3]  
 [2 5 1]]
```

```
np.delete(a, 1, axis=1)
```

```
[[7 9]  
 [4 3]  
 [2 1]]
```

```
np.delete(a, 1, axis=0)
```

```
[[7 8 9]  
 [2 5 1]]
```

## Математичке и статистичке операције над низом

оператори **+** **-** **\*** **/** **@**

функција **sum(...)**

функција **cumsum(...)**

функција **min(...)**

функција **max(...)**

функција **mean(...)**

функција **median(...)**

функција **np.sqrt(...)**

функција **np.sin(...)**

функција **np.cos(...)**

...

Математичке и статистичке операције над низом

```
a = np.array([[4, 1, 3], [7, 5, 9]])
```

```
[[4 1 3]  
 [7 5 9]]
```

```
a + a
```

```
[[ 8  2  6]  
 [14 10 18]]
```

```
a * a
```

```
[[16  1  9]  
 [49 25 81]]
```

Математичке и статистичке операције над низом

```
a = np.array([[4, 1, 3], [7, 5, 9]])
```

```
[[4 1 3]  
 [7 5 9]]
```

```
a.sum()
```

```
29
```

```
a.cumsum()
```

```
[ 4  5  8 15 20 29]
```

```
a.mean()
```

```
4.833333333333333
```

Математичке и статистичке операције над низом

```
a = np.array([[4, 1, 3], [7, 5, 9]])
```

```
[[4 1 3]  
 [7 5 9]]
```

```
np.sqrt(a)
```

```
[[2.          1.          1.73205081]  
 [2.64575131  2.23606798  3.          ]]
```

```
np.cos(a)
```

```
[[ -0.65364362  0.54030231 -0.9899925  ]  
 [ 0.75390225  0.28366219 -0.91113026]]
```

## Сортирање низа

функција **`np.sort(...)`**

основно сортирање низа

функција **`np.msort(...)`**

сортирање низа по првој димензији

функција **`np.argsort(...)`**

утврђивање позиција за потребе сортирања низа

...

## Сортирање низа

```
a = np.array([[2, 5, 6], [3, 1, 4]])
```

```
[[2 5 6]  
 [3 1 4]]
```

```
np.sort(a)
```

```
[[2 5 6]  
 [1 3 4]]
```

```
np.sort(a, axis=0)
```

```
[[2 1 4]  
 [3 5 6]]
```

## Претраживање низа

функција **`np.argmax(...)`**

утврђивање позиције најмањег елемента низа

функција **`np.argmin(...)`**

утврђивање позиције највећег елемента низа

функција **`np.argwhere(...)`**

утврђивање позиција за елементе низа који нису нула

функција **`np.extract(...)`**

проналажење елемената који задовољавају постављени критеријум

...

## Претраживање низа

```
a = np.array([[3, -2, 0], [1, 5, -7]])
```

```
[[ 3 -2  0]  
 [ 1  5 -7]]
```

```
np.argmax(a)
```

```
4
```

```
np.argmax(a, axis=0)
```

```
[0 1 0]
```

```
np.argmax(a, axis=1)
```

```
[0 1]
```

## Задатак 1.

Формирати произвољну квадратну матрицу целобројних вредности с ознаком *matr* и израчунати збир елемената на главној дијагонали те матрице!

## Задатак 2.

Формирати две матрице целобројних вредности димензија 3x4 коришћењем генератора псеудослучајних вредности! Матрице означити као *matrA* и *matrB*! Посебно иницијализовати генератор псеудослучајних вредности!

## Задатак 3.

За матрице  $matrA$  и  $matrB$  засебно израчунати аритметичку средину на следеће начине: по свим елементима заједно, по колонама и по врстама! Израчунате вредности заокружити на два децимална места!

## Задатак 4.

За матрицу  $matrA$  израчунати медијану! Одредити места појављивања медијане у матрици  $matrA$  и приказати одговарајуће координате!

## Задатак 5.

Од матрице  $matrB$  направити сортирану матрицу димензија  $4 \times 3$  на два начина: ређањем елемената по врстама и ређањем елемената по колонама!

## Задатак 6.

Написати функцију која ће за дату матрицу све садржане вредности мање од дефинисане доње граничне вредности поставити на ту граничну вредност! Испробати написану функцију за матрицу  $matrA$ !

## Задатак 7.

Написати функцију за издвајање позиција елемената дате матрице према растућем поретку тих елемената!  
Испробати написану функцију за матрицу *matrB*!

## Задатак 8.

Написати функцију која за дату доњу граничну вредност и дату горњу граничну вредност израчунава аритметичку средину за елементе дате матрице из опсега дефинисаног граничним вредностима! Израчунавање треба извести према задатом броју децималних места! Испробати написану функцију за матрицу *matrA*!

## Основна литература

NumPy. NumPy. Internet: <https://numpy.org/>

NumPy. NumPy documentation. Internet:  
<https://numpy.org/doc/stable/index.html>