

# Kratak uvod u NumPy - Numerical Python biblioteku

```
In [1]: # "#" Ovo je znak za komentar.  
# Uvezli smo biblioteku numpy kao instancu np koristeći komandu import  
  
import numpy as np
```

```
In [2]: # np sada možemo koristiti u velikom broju primera.  
# ukoliko otkucamo np. i pritisnemo TAB dobićemo listu mogućih funkcija  
  
# kreiramo niz od 6 elemenata  
a = np.array([0, 1, 2, 3, 4, 5])
```

```
In [3]: # a * 3 u slučaju da je a numpy instanca, množi se svaki element sa 3  
a = a * 3
```

**# Ovo je važan koncept u NumPy-ju (sličan kao i u Matlabu). # Vectorization::**  
**možemo je procesirati nizove na brz način bez korišćenja loop petlji**

```
In [4]: print (a)  
  
[ 0  3  6  9 12 15]
```

```
In [5]: # ovim smo generisali numerički niz od 6 elemenata.  
# možemo proveriti dimenzije niza. U ovom slučaju rezultat 1 nam  
# govori da se radi o vektoru.  
a.ndim
```

Out[5]: 1

```
In [6]: # komandom reshape možemo promeniti oblik niza. Od 1D niza (1x6), možemo kreirati  
# matricu, 2D niz, dimenzija (2 x 3)  
a.reshape(2,3)
```

```
Out[6]: array([[ 0,  3,  6],  
               [ 9, 12, 15]])
```

```
In [7]: b = np.zeros((3,4))
```

```
In [8]: b
```

```
Out[8]: array([[ 0.,  0.,  0.,  0.],  
               [ 0.,  0.,  0.,  0.],  
               [ 0.,  0.,  0.,  0.]])
```

```
In [9]: x = np.array([[1., 2., 3.], [4., 5., 6.]])
```

```
In [10]: print(x)
```

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

```
In [11]: x * 3
```

```
Out[11]: array([[ 3.,  6.,  9.],  
               [12., 15., 18.]])
```

## Indeksiranje

```
In [12]: N = 10  
x = np.arange(N)
```

```
In [13]: # ovom naredbom type mozemo proveriti koji je tip podataka x. --> numpy.ndarray  
type(x)
```

```
Out[13]: numpy.ndarray
```

```
In [14]: print(x)
```

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

```
In [15]: # Indeksiranje pocinje od 0, do N-1  
x[0]
```

```
Out[15]: 0
```

```
In [16]: print(x[1], x[5], x[-1])
```

```
1 5 9
```

```
In [17]: print(x[2:5])
```

```
[2 3 4]
```

## # Funkcije za brzo procesiranje pojedinačnih elemenata niza

**# "Element-wise processing". U Matlabu se za ovo koristila tacka, npr. ".\*"**

```
In [18]: x = np.array(range(10))
```

```
In [19]: print (x)
```

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

```
In [20]: np.sqrt(x)
```

```
Out[20]: array([ 0.          ,  1.          ,  1.41421356,  1.73205081,  2.          ,
                2.23606798,  2.44948974,  2.64575131,  2.82842712,  3.          ])
```

```
In [21]: x
```

```
Out[21]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [22]: x = np.sqrt(x)
```

```
In [23]: t = np.arange(1000, dtype=float)
```

```
In [24]: t = t / 1000.0
```

```
In [25]: x = np.cos(2*np.pi*3*t )
```

```
In [26]: import matplotlib.pyplot as plt
```

```
In [27]: # komentar koji je potreban za iscrtavanje grafika  
%matplotlib inline
```

```
In [28]: plt.plot(t,x)
```

```
Out[28]: [<matplotlib.lines.Line2D at 0x9e618d0>]
```

