NumPy

```
>>> import numpy
>>> cube = numpy.zeros((3,4,4), dtype=int)
>>> cube[0, :, 0::3] = 1
>>> cube[0, numpy.eye(4, dtype=bool)] = 1
>>> cube
```

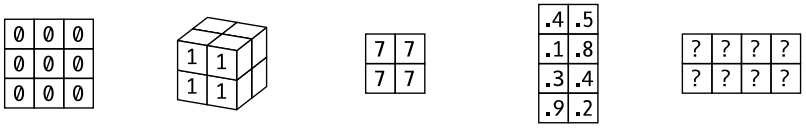
Tvorba polí

Daty

```
numpy.array([1, 2, 3])                  1D ze seznamu
numpy.array([[1, 2, 3], [4, 5, 6]])     2D ze seznamu seznamů
numpy.array([[[1, 2], [3, 4]], ...])    3D ze seznamu seznamů seznamů
numpy.array([[[[...]]]])                a tak dále...
```

Tvarem

```
numpy.zeros((3, 3))             3×3 z nul
numpy.ones((2, 2, 2))           2×2×2 z jedniček
numpy.full((2, 2), 7)           2×2 z konkrétní hodnoty
numpy.random.random((4, 2))     4×2 náhodné hodnoty (0, 1)
numpy.empty((2, 4))             2×4 neinicializované hodnoty jako v C
```

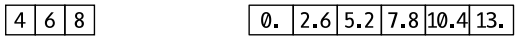


Čtvercové matice

```
numpy.eye(4)                    jednotková
numpy.diag([1, 2, 3, 4])        diagonální
```

Číslené řady

```
numpy.arange(4, 10, 2)          jako range()
numpy.linspace(0, 13, num=6)    pro floaty
```



Datové typy

```
numpy.zeros((3, 3), dtype=int)    parametr dtype určuje typ
int, float, bool, nejhůře object  datové typy z Pythonu
stringem 'int8', 'uint64' apod.   celá čísla různých velikostí
řetězce znaků ('U', 8)            délka max 8
řetězce bytů ('a', 3)             délka max 3
a.astype(float)                   vrací pole daného typu
```

Atributy

```
a.shape     tvar (velikost)
a.size      počet prvků
a.dtype     typ
a.ndim      dimenze
```

Matematické operace

```
+ - * / // ...      po prvcích
+= -= *= /= ...     modifikuje původní
> >= < <= == ...    vrací pravdivostní tabulku
a @ b               maticové násobení
a.T                 transponovaná matice
```

Indexování

Na jedné dimenzi (jako seznam)

```
matrix[0]           první „řádek“
matrix[0:-1:2]      řezání jako v seznamech
matrix[0][1]        jako se seznamem seznamů (pomalé)
```

n-ticí

```
matrix[0, 1]        prvek na souřadnici 0, 1
matrix[0:-1, 1:]    řezání podle více dimenzí
matrix[:, 1]        kompletní interval sežere dimenzi
cube[:, :, 0]       jde nahradit cube[..., 0]
```

Pravdivostní tabulkou

```
array[array > 4]                    vrátí vektor hodnot
array[(array > 4) & (array < 8)]    skládání pomocí bitových operátorů
```

Je pole pravdivé?

```
if matrix:                  ValueError
if matrix.any():            alespoň jedna pravdivá hodnota
if matrix.all():            všechny pravdivé hodnoty
```

Detailní povídání na: http://naucse.python.cz/course/mi-pyt/intro/numpy/