



```
>>> 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...
```

1	2	3
---	---	---

1	2	3
4	5	6

1	2	3
3	4	

...

Tvarem

```
numpy.zeros((3, 3))          3x3 z nul
numpy.ones((2, 2, 2))        2x2x2 z jedniček
numpy.full((2, 2), 7)        2x2 z konkrétní hodnoty
numpy.random.random((4, 2))  4x2 náhodné hodnoty (0,1)
numpy.empty((2, 4))          2x4 neinicializované hodnoty jako v C
```

0	0	0
0	0	0
0	0	0

1	1
1	1

7	7
7	7

.4	.5
.1	.8
.3	.4
.9	.2

?	?	?	?
?	?	?	?

Čtvercové matice

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

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

1	0	0	0
0	2	0	0
0	0	3	0
0	0	0	4

Číselné řady

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

4	6	8
---	---	---

0.	2.6	5.2	7.8	10.4	13.
----	-----	-----	-----	------	-----

Datové typy

```
numpy.zeros((3, 3), dtype=int)
int, float, bool, nejhůře object
stringem 'int8', 'uint64' apod.
řetězce znaků ('U', 8)
řetězce bytů ('a', 3)
a.astype(float)
```

parametr dtype určuje typ
datové typy z Pythonu
celá čísla různých velikostí
délka max 8
délka max 3
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
> >= < <= == ... vrátí pravdivostní tabulku
a @ b          maticové násobení
a.T            transponovaná matice
```

Indexování

Na jedné dimenzi (jako seznam)

```
matrix[0]
matrix[0:-1:2]
matrix[0][1]
```

první „řádek“
řezání jako v seznamech
jako se seznamem seznamů (pomale)

n-ticí

```
matrix[0, 1]
matrix[0:-1, 1:]
matrix[:, 1]
cube[:, :, 0]
```

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

Pravdivostní tabulkou

```
array[array > 4]
array[(array > 4) & (array < 8)]
```

vrátí vektor hodnot
skládání pomocí bitových operátorů

Je pole pravdivé?

```
if matrix:
if matrix.any():
if matrix.all():
```

ValueError
alespoň jedna pravdivá hodnota
všechny pravdivé hodnoty

Detailní povídání na: <http://nauce.python.cz/course/mi-pyt/intro/numpy/>