



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...
```

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

1	2	3
4	5	6

1	2
3	4

...

Tvarem

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

0	0	0
0	0	0
1	0	0

1	1
1	1

7	7
7	7

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

?	?	?	?
?	?	?	?

Čtvercové matice

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

jednotková
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', 'unit64' apod.
řetězcem znaků ('U', 8)
řetězcem bytů ('a', 3)
.astype(float)
```

parametr dtype určuje typ
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]          první prvek
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://nauce.python.cz/course/mi-pyt/intro/numpy/>

© 2017 Miro Hrončok. Ke stažení na <https://github.com/pyvec/cheatsheets>
Šířeno pod licencí CC BY-SA: <https://creativecommons.org/licenses/by-sa/4.0/>

Ukázky kódu šířeny pod licencí CC0: <https://creativecommons.org/publicdomain/zero/1.0/>

Logo NumPy © 2005–2017 NumPy Developers, BSD 3-clause License: <https://github.com/numpy/numpy/blob/master/LICENSE.txt>

