

# NumPy

Jan Popko

Python Advanced

# NumPy - Operationen

Operator	Funktion	Bedeutung
+	<code>np.add()</code>	Addition
-	<code>np.subtract()</code>	Subtraktion
-	<code>np.negative()</code>	Negierung
*	<code>np.multiply()</code>	Multiplikation
/	<code>np.divide()</code>	Division
//	<code>np.floor_divide()</code>	Ganzzahlige Division
**	<code>np.power()</code>	Exponentieren
%	<code>np.mod()</code>	Modulo

## Möglich mit:

`np.Funktion.reduce()`

Array wird mit einer Funktion um eine Dimension reduziert

`np.Funktion.accumulate()`

Funktion wird auf alle vorherigen Elemente angewandt

Jan Popko

Python Advanced

# NumPy – NaN compatible Funktionen

Funktion	NaN-Compatible Funktion	Beschreibung
<code>np.sum</code>	<code>np.nansum</code>	Berechne Summe der Elemente
<code>np.prod</code>	<code>np.nanprod</code>	Berechne Produkt der Elemente
<code>np.mean</code>	<code>np.nanmean</code>	Berechne Median der Elemente
<code>np.std</code>	<code>np.nanstd</code>	Berechne Standardabweichung
<code>np.var</code>	<code>np.nanvar</code>	Berechne Varianz
<code>np.min</code>	<code>np.nanmin</code>	Finde Minimum
<code>np.max</code>	<code>np.nanmax</code>	Finde Maximum
<code>np.argmin</code>	<code>np.nanargmin</code>	Finde Index vom Minimum
<code>np.argmax</code>	<code>np.nanargmax</code>	Finde Index vom Maximum
<code>np.median</code>	<code>np.nanmedian</code>	Berechne Median der Elemente
<code>np.percentile</code>	<code>np.nanpercentile</code>	Berechne Perzentil
<code>np.any</code>	N/A	Gibt es irgend ein True
<code>np.all</code>	N/A	Sind alle Elemente True

Jan Popko

Python Advanced

# NumPy - Broadcasting

Broadcasting von Arrays:

Regel 1:

Wenn zwei Arrays sich in Ihren Dimensionen unterscheiden, werden sie mit den Elementen ihrer linken Seite ausgepolstert.

Regel 2: Wenn die Form der beiden Arrays in keiner Dimension passt, wird das Array mit einer Dimension gleich eins gestreckt um sich an die andere Form anzugleichen.

Regel 3: Wenn die Dimensionen nicht gleich sind und keine gleich eins ist, dann wird ein Fehler ausgegeben.

`np.arange(3)+5`

0	1	2
---	---	---

 + 

5	5	5
---	---	---

 = 

5	6	7
---	---	---

`np.ones((3, 3))+np.arange(3)`

1	1	1
1	1	1
1	1	1

 + 

0	1	2
0	1	2
0	1	2

 = 

1	2	3
1	2	3
1	2	3

`np.ones((3, 1))+np.arange(3)`

0	0	0
1	1	1
2	2	2

 + 

0	1	2
0	1	2
0	1	2

 = 

0	1	2
1	2	3
2	3	4

# NumPy – Elementenvergleich von Arrays

Operator	Funktion für Arrays	Bedeutung
<code>==</code>	<code>np.equal</code>	ist gleich
<code>!=</code>	<code>np.not_equal</code>	ist nicht gleich
<code>&lt;</code>	<code>np.less</code>	kleiner
<code>&lt;=</code>	<code>np.less_equal</code>	kleiner gleich
<code>&gt;</code>	<code>np.greater</code>	größer
<code>&gt;=</code>	<code>np.greater_equal</code>	größer gleich
<code>&amp;</code>	<code>np.bitwise_and</code>	bitweise und
<code> </code>	<code>np.bitwise_or</code>	bitweise oder
<code>^</code>	<code>np.bitwise_xor</code>	bitweise entweder, oder
<code>~</code>	<code>np.bitwise_not</code>	bitweise nicht (dreht True/False um)
	<code>np.isclose</code>	ist fast gleich

`np.isnan` ist ein Element in einem Array `np.nan`

In Verbindung mit `np.any()` oder `np.all()` möglich.

Jan Popko

Python Advanced

# NumPy – Indexed Arrays

```
name = ['Alice', 'Bob', 'Cathy', 'Doug']
age = [25, 45, 37, 19]
weight = [55.0, 85.5, 68.0, 61.5]

data = np.zeros(4, dtype={'names':('name', 'age', 'weight'),
'formats':('U10', 'i4', 'f8')})
print(data.dtype)
[('name', '<U10'), ('age', '<i4'), ('weight', '<f8')]

data['name'] = name
data['age'] = age
data['weight'] = weight
print(data)

recarray: kann mit Punktoperator zugegriffen werden
data_rec = data.view(np.recarray)
print(data_rec.age)
```

Jan Popko

Python Advanced

# NumPy - DatenTypen

*NumPy Datentypen:*

'b'	Byte	<code>np.dtype('b')</code>
'i'	Int mit Vorzeichen	<code>np.dtype('i4') == np.int32</code>
'u'	Int ohne Vorzeichen	<code>np.dtype('u1') == np.uint8</code>
'f'	Float	<code>np.dtype('f8') == np.float64</code>
'c'	Complex floating point	<code>np.dtype('c16') == np.complex128</code>
'S', 'a'	String	<code>np.dtype('S5') (Max Länge == 5)</code>
'U'	Unicode string	<code>np.dtype('U') == np.str_</code>
'V'	Raw data (void)	<code>np.dtype('V') == np.void</code>