

NumPy Grundlagen

Die Python-Bibliothek NumPy ist ein wichtiges Modul für Scientific Computing mit Python. Wesentlich hierbei die NumPy-eigene hochdimensionale Datenstruktur **array** mit ihren Werkzeugen.

Import

Üblich ist der Import von NumPy mit **import numpy as np**

Arrays erzeugen

Die Dimensionalität eines Arrays wird mit aufeinanderfolgenden Tupeln oder Listen gleicher Länge bestimmt:

```
a = np.array([1,2,3])
b = np.array([(1.5,2,3), (4,5,6)], dtype = float)
c = np.array([[(1.5,2,3), (4,5,6)], [(3,2,1), (4,5,6)]])
```

Platzhalter

Arrays können bei der Erzeugung mit Werten befüllt werden:

```
np.zeros((3,4))
Array von Nullen
np.ones((2,3,4))
Array von Einsen
d = np.arange(10,25,5))
Array-Range mit Schrittgrösse 5
np.linspace(0,2,9)
9 gleichverteilte Werte im Bereich 0..2
e = np.full((2,2),7)
Array von Konstanten
f = np.eye(2)
2x2 Identitätsmatrix
np.empty((3,2))
Leeres Array
```

Speichern und Einlesen

```
Textdatei:
np.loadtxt("meinedatei.txt")
np.genfromtxt("meinedatei.csv", delimiter=',')
np.savetxt("meinarray.txt", a, delimiter=)
```

NumPy-Datenformat:

```
np.save('mein_array', a)
np.savez('array.npz', a, b) # Mehrere Arrays
np.load('mein_array.npy') # Oder .npz ...
```

Arbeiten mit Arrays

Array Eigenschaften

Befehl	Beschreibung
a.shape	Dimensionierung
len(a)	Länge
b.ndim	Anzahl Dimensionen
e.size	Anzahl Elemente
b.dtype	Datentyp der Elemente
b.dtype.name	Name des Datentyps
b.astype(int)	Typkonvertierung
np.info(np.ndarray)	Hilfe

Array-Mathematik

```
g = a - b # Subtraktion
array([[ -0.5,  0. ,  0. ], [ -3. , -3. , -3. ]])
b + a # Addition
array([[2.5, 4. , 6. ], [5. , 7. , 9. ]])
a / b # Division
array([[0.66666667, 1. , 1. ], [0.25 , 0.4 , 0.5 ]])
a * b # Multiplikation
array([[ 1.5, 4. , 9. ], [ 4. , 10. , 18. ]])
np.exp(b) # Potenzierung
np.sqrt(b) # Quadratwurzel
np.sin(a) # Sinus
np.cos(b) # Cosinus
np.log(a) # Logarithmus
e.dot(f) # Punktprodukt
array([[7., 7.], [7., 7.]])
```

Vergleichsoperatoren

```
a == b # Elementweiser Vergleich
array([[False, True, True], [False, False, False]])
a < 2 # Elementweiser Vergleich
array([ True, False, False])
np.array_equal(a, b) # Array-Vergleich
```

Aggregierungsfunktionen

Funktion	Bedeutung
a.sum()	Summierung des Arrays
a.min()	Minimum des Arrays
b.max(axis=0)	Maximum einer Zeile
b.cumsum(axis=1)	Kumulative Summe der Elemente
a.mean()	Mittelwert
b.median()	Median
a.corrcoef()	Korrelationskoeffizient
np.std(b)	Standardabweichung

Fortgeschrittene Arrayoperationen

Slicing

a[2] 3	Wähle das Element am Index 2
b[1,2] 6.0	Element mit Spaltenindex 0 und Zeilenindex 1 (entspricht b[1][2])
a[0:2] array([1, 2])	Wähle die Elemente am Index 0 und 1
b[0:2,1] [0:2,1]	Wähle die Elemente in Zeile 0 und 1 in der Spalte mit dem Index 1
b[:1] array([[1.5, 2. , 3.]])	Wähle alle Elemente in der Zeile mit dem Index 0
c[1,...] array([[3., 2., 1.], [4., 5., 6.]])	Entspricht c[1,:,:]
a[: :-1] array([3, 2, 1])	Invertiere a
a[a<2] array([1])	Boolsche Indizierung: wähle alle Elemente aus a kleiner 2
a[: :-1] array([3, 2, 1])	Invertiere a

Array-Manipulation

i = np.transpose(b) b.ravel() g.reshape(3,-2) h.resize((2,6)) np.append(h,g) np.insert(a, 1, 5) np.delete(a,[1]) np.concatenate((a,d),axis=0) array([1, 2, 3, 10, 15, 20]) np.vstack((a,b)) array([[1. , 2. , 3.], [1.5, 2. , 3.], [4. , 5. , 6.]]) np.hstack((e,f)) array([[7., 7., 1., 0.], [7., 7., 0., 1.]]) np.hsplit(a,3) [array([1]), array([2]), array([3])] np.vsplit(c,2) [array([[1.5, 2. , 3.],[4. , 5. , 6.]]), array([[3., 2., 1.], [4., 5., 6.]])]	Transponieren Array Verflachen Form ändern mit Originaldaten Dimensionalität auf 2,6 setzen Elemente an dan Array anhängen Elemente einfügen Elemente aus dem Array löschen Arrays zusammenfügen Arrays vertikal zusammenfügen Arrays horizontal zusammenfügen Horizontal am 3. Index spalten Vertikal am 2. Index spalten
---	--

Arrays sortieren

a.sort() c.sort(axis=0)	Sortiere das Array Sortiere die Elemente der spezifizierten Dimension
----------------------------	--

Arrays kopieren

h = a.view() np.copy(a) h = a.copy()	Erzeuge eine Ansicht des Arrays mit den gleichen Daten Erzeuge eine Kopie des Arrays Erzeuge eine tiefe Kopie des Arrays
--	--