Kiru0310 / Python-for-machine-learning

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main Python-for-machine-learning / numpy 2.ipynb Go to file t ···

Kiru0310 Add files via upload 980214b · 6 minutes ago

211 lines (211 loc) · 6.26 KB

Preview Code Blame Raw

In [17]:

```
import numpy as np
arry=np.array([[-1,2,0,4],[4,-0.5,6,0],[2.6,0,7,8,],[3,-7,4,2.0]])
print("original array:\n",arry)
```

```
original array:
 [[-1.   2.   0.   4. ]
 [ 4.  -0.5  6.   0. ]
 [ 2.6  0.   7.   8. ]
 [ 3.  -7.   4.   2. ]]
```

In [9]:

```
print("every other rows:\n",arr[0:3:2])
```

```
every other rows:
 [[-1.   2.   0.   4. ]
 [ 2.6  0.   7.   8. ]]
```

In [16]:

```
arr=np.array([1,2,3,4,5,6,7])
print("original array:",arr)
print("\n return every other element in the array:arr[::2]:",arr[::2])
```

```
original array: [1 2 3 4 5 6 7]

 return every other element in the array:arr[::2]: [1 3 5 7]
```

In [18]:

```
#slicing array
temp=arry[:2,:3]
print("\n Array with first 2 rows and 3 columns:\n",temp)
```

```
 Array with first 2 rows and 3 columns:
 [[-1.   2.   0. ]
 [ 4.  -0.5  6. ]]
```

In [19]:

```
import numpy as np
arr = np.array([[-1,2,0,4],[4,-0.5,6,0],[2.6,0,7,8],[3,-7,4,2.0]])
temp=arr[[0,1,2,3],[3,2,1,0]]
print("\n elements at indices(0,3),(1,2),(2,1),""(3,0):\n",temp)
```

```
 elements at indices(0,3),(1,2),(2,1),(3,0):
 [4. 6. 0. 3.]
```

In [20]:

```
cond=arr>2
temp=arr[cond]
print("\n elements greater than 2:\n",temp)
```

```
 elements greater than 2:
 [4.  4.  6.  2.6 7.  8.  3.  4. ]
```

In [25]:

```
arr1=np.array([1,2,3])
arr2=np.array([4,5,6])
arr=np.concatenate((arr1,arr2))
print("\n original arrays:\n",arr1,arr2)
print("\n Joined array:\n",arr)
```

```
 original arrays:
 [1 2 3] [4 5 6]

 Joined array:
 [1 2 3 4 5 6]
```

In [26]:

```
arr=np.hstack((arr1,arr2))
print("\nhorizontal joining:\n",arr)
```

```
horizontal joining:
 [1 2 3 4 5 6]
```

In [30]:

```
arr=np.vstack((arr1,arr2))
print("\n vertical joining:\n",arr)
```

```
 vertical joining:
 [[1 2 3]
 [4 5 6]]
 [4 5 6]]
```

In [31]:

```python
arr=np.dstack((arr1,arr2))
print("\ndepth joining:\n",arr)
```

```
depth joining:
 [[[1 4]
  [2 5]
  [3 6]]]
```

In [32]:

```python
arr=np.array([1,2,3,4,5,6])
newarr=np.array_split(arr,3)
print("\noriginal array:\n",arr)
print("\nsplitted array:\n",newarr)
print("\n splitted array in another form:\n")
print(newarr[0])
print(newarr[1])
print(newarr[2])
```

```
original array:
 [1 2 3 4 5 6]

splitted array:
 [array([1, 2]), array([3, 4]), array([5, 6])]

 splitted array in another form:

[1 2]
[3 4]
[5 6]
```

In [ ]:

```python

```