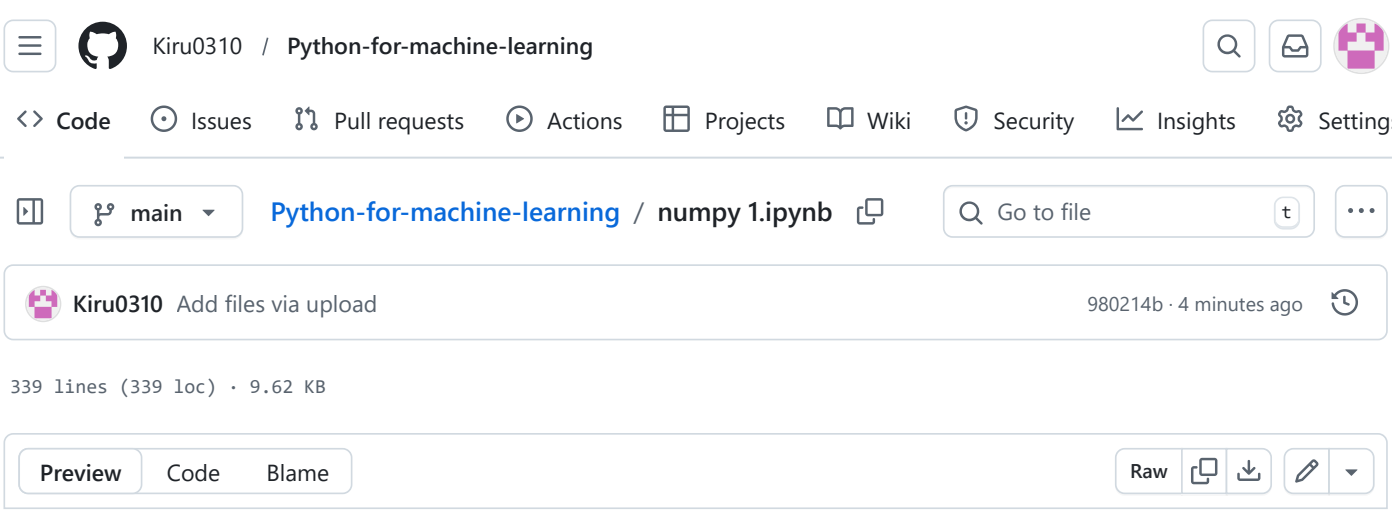
Kiru0310 / Python-for-machine-learning
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
main
Python-for-machine-learning / numpy 1.ipynb
Go to file
t
Kiru0310 Add files via upload
980214b · 4 minutes ago
339 lines (339 loc) · 9.62 KB
Preview
Code
Blame
Raw

In [1]:
```
import numpy as np
a = np.array([[1, 2, 4], [5, 8, 7]])
print ("Array created using passed list:\n", a)
```

```
Array created using passed list:
 [[1 2 4]
 [5 8 7]]
```

In [2]:
```
b = np.zeros((3, 4))
print ("\nAn array initialized with all zeros:\n", b)
```

```
An array initialized with all zeros:
 [[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]
```

In [3]:
```
c = np.full((3, 3), 6)
print ("\nAn array initialized with all 6s.\n", c)
```

```
An array initialized with all 6s.
 [[6 6 6]
 [6 6 6]
 [6 6 6]]
```

In [4]:
```
d = np.random.random((2, 2))
print ("\nA random array:\n", d)
```

```
A random array:
 [[0.31867621 0.30424336]
 [0.37980133 0.34074526]]
```

In [5]:
```
e = np.arange(0, 30, 5)
print ("\n A sequential array with steps of 5:\n", e)
```

```
 A sequential array with steps of 5:
 [ 0  5 10 15 20 25]
```

In [7]:
```
arr = np.array([[1, 2, 3, 4], [5, 2, 4, 2],[1, 2, 0, 1]])
newarr = arr.reshape(4, 3)
print ("\nOriginal array:\n", arr)
print ("Reshaped array[4,3]:\n", newarr)
```

```
Original array:
 [[1 2 3 4]
 [5 2 4 2]
 [1 2 0 1]]
Reshaped array[4,3]:
 [[1 2 3]
 [4 5 2]
 [4 2 1]
 [2 0 1]]
```

In [8]:
```
flarr= arr.flatten()
print ("\nOriginal array:\n", arr)
print ("Fattened array:\n", flarr)
```

```
Original array:
 [[1 2 3 4]
 [5 2 4 2]
 [1 2 0 1]]
Fattened array:
 [1 2 3 4 5 2 4 2 1 2 0 1]
```

In [10]:
```
# Printing array dimensions (axes)
print("\nNo. of dimensions: ", arr.ndim)
```

```
No. of dimensions:  2
```

In [11]:
```
# Printing shape of array
print("\nShape of array: ", arr.shape)
```

```
Shape of array:  (3, 4)
```

In [12]:
```
# Printing size (total number of elements) of array
print("\nSize of array: ", arr.size)
```

```
Size of array:  12
```

In [13]:
```
# Printing type of elements in array
print("\nArray stores elements of type: ", arr.dtype)
```

```
Array stores elements of type:  int32
```

In [16]:
```
import numpy as np

# original integer array
arr = np.array([1, 2, 3, 4, 5])

# converting datatype from integer to float
newtype = arr.astype(float)

print("\nConverted array elements:\n", newtype)
print("Converted array type:", newtype.dtype)
```

```
Converted array elements:
 [1. 2. 3. 4. 5.]
Converted array type: float64
```

In [17]:
```
import numpy as np
p=np.array([[1,2,3],[4,5,6],[7,8,9],[10,11,12],[13,14,15]])
print(p[3:0:-1])
```

```
[[10 11 12]
 [ 7  8  9]
 [ 4  5  6]]
```

In [18]:
```
import numpy as np
p=np.array([[1,2,3],[4,5,6],[7,8,9],[10,11,12],[13,14,15]])
print(p[2,0:2])
```

```
[7 8]
```

In [19]:
```
import numpy as np
p=np.array([[1,2,3],[4,5,6],[7,8,9],[10,11,12],[13,14,15]])
print(p[2:,2:])
```

```
[[ 9]
 [12]
 [15]]
```

In [20]:
```
import numpy as np
p=np.array([[1,2,3],[4,5,6],[7,8,9],[10,11,12],[13,14,15]])
print(p[3:,3:])
```

```
[]
```

In [21]:
```
import numpy as np
p=np.array([[1,2,3],[4,5,6],[7,8,9],[10,11,12],[13,14,15]])
print(p[:,1])
```

```
[ 2  5  8 11 14]
```

In [22]:
```
import numpy as np
p=np.array([[1,2,3],[4,5,6],[7,8,9],[10,11,12],[13,14,15]])
c=p.astype('f')
print(c)
```

```
[[ 1.  2.  3.]
 [ 4.  5.  6.]
 [ 7.  8.  9.]
 [10. 11. 12.]
 [13. 14. 15.]]
```

In [23]:
```
import numpy as np
p=np.array([[1,2,3],[4,5,6],[7,8,9],[10,11,12],[13,14,15]])
c=p.astype('i')
print(c)
```

```
[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]
 [13 14 15]]
```

In [ ]:
```

```