

Complete Numpy Documentation

Array Creation Functions

```
In [1]: import numpy as np
```

```
In [2]: a=np.array([1,2,3])  
print("Array a:",a)
```

Array a: [1 2 3]

```
In [3]: b=np.arange(0, 10, 2)  
print("Array b:",b)
```

Array b: [0 2 4 6 8]

```
In [4]: d=np.zeros((2,3))  
print("Array d:\n",d)
```

Array d:
[[0. 0. 0.]
 [0. 0. 0.]]

```
In [5]: e=np.ones((3,2))  
print("Array e:\n",e)
```

Array e:
[[1. 1.]
 [1. 1.]
 [1. 1.]]

```
In [6]: e
```

```
Out[6]: array([[1., 1.],  
              [1., 1.],  
              [1., 1.]])
```

```
In [7]: f=np.eye(4)  
print("Identity matrix f:\n",f)
```

Identity matrix f:
[[1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 0. 0. 1.]]

```
In [8]: f=np.eye(3)  
print("Identity matrix f:\n",f)
```

Identity matrix f:
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]

2.Array Manipulation Functions

```
In [9]: a1=np.array([1,2,3])
        reshaped=np.reshape(a1,(1,3))
        print("Reshaped array:",reshaped)
```

Reshaped array: `[[1 2 3]]`

```
In [10]: np.arange(10,30).reshape(2,3)
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In[10], line 1
----> 1 np.arange(10,30).reshape(2,3)

ValueError: cannot reshape array of size 20 into shape (2,3)
```

```
In [ ]: np.reshape(10,30,(3,2))
```

```
In [ ]: a=np.array([1,2,3])
        a
```

```
In [ ]: a
```

```
In [ ]: a=np.reshape(3,3)
```

```
In [13]: f1=np.array([[1,2],[3,4]])
        flattened=np.ravel(f1)
        print("Flattened array:",flattened)
```

Flattened array: `[1 2 3 4]`

```
In [ ]: e1=np.array([[1,2],[3,4]])
        transposed=np.transpose(e1)
        print("Transposed array:\n",transposed)
```

```
In [ ]: a2=np.array([1,2])
        b2=np.array([3,4])
        stacked=np.vstack([a2,b2])
        print("Stacked arrays:\n",stacked)
```

```
In [ ]: g=np.array([1,2,3,4])
        added=np.add(g,2)
        print("Added 2 to g:",added)
```

```
In [ ]: squared=np.power(g,2)
        print("Squared g:",squared)
```

```
In [ ]: sqrt_val=np.sqrt(g)
        print("square root of g:",sqrt_val)
```

```
In [ ]: print(a1)
```

```
In [ ]: print(g)
```

```
In [ ]: a2=np.array([1,2,3])
        dot_product=np.dot(a2, g)
        print("Dot product of a and g:",dot_product)
```

```
In [ ]: print(a)
        print(a1)
```

```
In [ ]: a3=np.array([1, 2, 3])
        dot_product=np.dot(a1, a)
        print("Dot product of a1 and a:",dot_product)
```

4. Statistical Functions

```
In [ ]: s=np.array([1, 2, 3, 4])
        mean=np.mean(s)
        print("Mean of s:",mean)
```

```
In [ ]: m=np.arange(1,5)
```

```
In [ ]: m
```

```
In [ ]: np.mean(m)
```

```
In [ ]: std_dev = np.std(s)
        print("Standard deviation of s:",std_dev)
```

```
In [ ]: minimum = np.min(s)
        print("Min of s:",minimum)
```

```
In [ ]: s
```

```
In [ ]: maximum=np.max(s)
        print("Max of s:",maximum)
```

```
In [ ]: np.median(s)
```

5.Linear Algebra Functions

```
In [11]: matrix=np.array([[1,2],[3,4]])
```

```
In [12]: matrix
```

```
Out[12]: array([[1, 2],
                [3, 4]])
```

6.Random Sampling Functions

```
In [ ]: random_vals=np.random.rand(3)
        print("Random vlaues:",random_vals)
```

```
In [ ]: np.random.seed(0)
        random_vals=np.random.rand(3)
        print("Random values:",random_vals)
```

```
In [ ]: np.random.seed(0)
```

```
In [ ]: rand_ints=np.random.randint(0,10, size=5)
```

```
In [ ]: rand_ints
```

```
In [ ]: rand_ints=np.random.randint(0,10,5)
```

```
In [ ]: rand_ints
```

```
In [ ]: rand_ints=np.random.randint(0,10,2)
```

```
In [ ]: rand_ints
```

```
In [ ]: np.random.seed(0)
rand_ints=np.random.randint(0,10,size=5)
print("Random integers:",rand_ints)
```

7.Boolean & Logical Functions

```
In [ ]: logical_test=np.array([True,False,True])
all_true=np.all(logical_test)
print("All elements True:",all_true)
```

```
In [ ]: logical_test=np.array([True,False,True])
all_true=np.all(logical_test)
print("All elements True:",all_true)
```

```
In [ ]: logical_test=np.array([False,False,False])
all_true=np.all(logical_test)
print("All elements True:",all_true)
```

```
In [ ]: any_true=np.any(logical_test)
print("Any elements True:",any_true)
```

8.Set Operations

```
In [ ]: set_a=np.array([1,2,3,4])
set_b=np.array([3,4,5,6])
intersection=np.intersect1d(set_a, set_b)
print("Intersection of a and b:",intersection)
```

```
In [ ]: union=np.union1d(set_a,set_b)
print("Union of a and b:",union)
```

9.Array Attribute Functions

```
In [ ]: a=np.array([1, 2, 3])
shape=a.shape
```

```
size=a.size
dimensions=a.ndim
dtype=a.dtype

print("Shape of a:",shape)
print("Size of a:",size)
print("Number of dimensions of a:",dimensions)
print("Data type of a:",dtype)
```

10.Other Functions

```
In [ ]: a=np.array([1,2,3])
        copied_array=np.copy(a)
        print("Copied array:",copied_array)
```

```
In [ ]: array_size_in_bytes=a.nbytes
        print("Size of a in bytes:",array_size_in_bytes)
```

```
In [ ]: shared=np.shares_memory(a,copied_array)
        print("Do a and copied_array share memory?",shared)
```

```
In [ ]:
```