

#1)CREATE A 3X3 NUMPY ARRAY WITH RANDOM INTEGERS BETWEEN 1 AND 100 THEN,INTERCHANGE

```
import numpy as np
array=np.random.randint(1,101,size=(3,3))
print("original array:")
print(array)
transposed_array=array.T
print("\n transported array:")
print(transposed_array)
```

```
⇒ original array:
[[40 64  6]
 [93 12 54]
 [53 47 96]]

transported array:
[[40 93 53]
 [64 12 47]
 [ 6 54 96]]
```

#2:)generate a 1d numpy array with 10 elements reshape it into a 2x5 array, then int

```
import numpy as np
array_1d=np.arange(10)
print("original array:")
print(array_1d)
array_2x5=array_1d.reshape(2,5)
print(array_2x5)
array_5x2=array_1d.reshape(5,2)
print("\nreshaped to 5x2 array:")
print(array_5x2)
```

```
⇒ original array:
[0 1 2 3 4 5 6 7 8 9]
[[0 1 2 3 4]
 [5 6 7 8 9]]

reshaped to 5x2 array:
[[0 1]
 [2 3]
 [4 5]
 [6 7]
 [8 9]]
```

#create a 4x4 numpy array with random float values add a border of zeros around it result

```
import numpy as np
#xcreate a 4X4 array
original_array=np.random.rand(4,4)
#adding border
bordered_array=np.pad(original_array,pad_width=1,mode='constant',constant_values=0)

print("original 4x4 array:")
print(original_array)
print("\n6x6 array with a border of zeros:")
print(bordered_array)
```

```

➡ original 4x4 array:
[[0.58234617 0.55078973 0.56653237 0.23956233]
 [0.47441892 0.68025788 0.28959154 0.42574083]
 [0.36588924 0.19369737 0.6464166 0.64933434]
 [0.80634653 0.22786812 0.13779374 0.99258246]]

```

```

6x6 array with a border of zeros:
[[0. 0. 0. 0. 0. 0.]
 [0. 0.58234617 0.55078973 0.56653237 0.23956233 0.]
 [0. 0.47441892 0.68025788 0.28959154 0.42574083 0.]
 [0. 0.36588924 0.19369737 0.6464166 0.64933434 0.]
 [0. 0.80634653 0.22786812 0.13779374 0.99258246 0.]
 [0. 0. 0. 0. 0. 0.]]

```

#4) using numpy create an array of integers from 10 to 60 with a step of 5

```

import numpy as np
array=np.arange(10,65,5)
print(array)

```

```

➡ [10 15 20 25 30 35 40 45 50 55 60]

```

#5) create a numpy array of strings ['python','numpy','pandas'].apply different case tran

```

import numpy as np
#create a numpy array of strings
arr=np.array(['python','numpy','pandas'])
#apply different case transformations
uppercase=np.char.upper(arr)
lowercase=np.char.lower(arr)
titlecase=np.char.title(arr)
capitalize=np.char.capitalize(arr)
#print the result
print("original array:",arr)
print("uppercase:",uppercase)
print("lowercase:",lowercase)
print("titlecase:",titlecase)
print("capitalize:",capitalize)

```

```

➡ original array: ['python' 'numpy' 'pandas']
uppercase: ['PYTHON' 'NUMPY' 'PANDAS']
lowercase: ['python' 'numpy' 'pandas']
titlecase: ['Python' 'Numpy' 'Pandas']
capitalize: ['Python' 'Numpy' 'Pandas']

```

#6:)generate a numpy array of words insert a space between each character of every word i

```

import numpy as np
#create a numpy array words
words=np.array(['hello','world','python'])
#insert a space between each character of every word
spaced_words=np.char.join(' ',words)
#print the result
print("original array:",words)
print("words with spaces:",spaced_words)

```

```

→ original array: ['hello' 'world' 'python']
  words with spaces: ['h e l l o' 'w o r l d' 'p y t h o n']

```

```

#7:)create a 2d numpy arrays and perform elementwise addition subtraction multiplication
import numpy as np
#create two 2d numpy arrays
array1=np.array([[1,2,3],[4,5,6]])
array2=np.array([[7,8,9],[10,11,12]])
#perform elementwise addition
addition=array1+array2
#subtraction
subtraction=array1-array2
#multiplication
multiplication=array1*array2
#division
division=array1/array2
#print result
print("array 1:\n",array1)
print("array 2L:\n",array2)
print("\naddition:\n",addition)
print("\nsubtraction:\n",subtraction)
print("\nmultiplication:\n",multiplication)
print("\ndivision:\n",division)

```

```

→ array 1:
  [[1 2 3]
   [4 5 6]]
array 2L:
  [[ 7  8  9]
   [10 11 12]]

addition:
  [[ 8 10 12]
   [14 16 18]]

subtraction:
  [[-6 -6 -6]
   [-6 -6 -6]]

multiplication:
  [[ 7 16 27]
   [40 55 72]]

division:
  [[0.14285714 0.25      0.33333333]
   [0.4       0.45454545 0.5       ]]

```

```

#8)use numpy to create a 5x5 identity matrix then extract its diagonal elements
import numpy as np
#create a 5x5 matrix
identity_matrix=np.eye(5)
#extract the diagonal elements
diagonal_elements=np.diag(identity_matrix)
#print result

```

```
print("5x5 identity matrix:\n",identity_matrix)
print("\ndiagonal elements:",diagonal_elements)
```

```
⇒ 5x5 identity matrix:
[[1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]]
```

```
diagonal elements: [1. 1. 1. 1. 1.]
```

#9)generate a numpy array of 100 random .....

```
import numpy as np
random_integers=np.random.randint(0,1001,size=100)
#function to check if a number is prime
def is_prime(num):
    if num<=1:
        return False
    for i in range(2,int(num**0.5)+1):
        if num % i==0:
            return False
    return True
#find all prime numbers in the array
prime_numbers=[num for num in random_integers if is_prime(num)]
#print result
print("random integers array:\n",random_integers)
print("\nprime numbers in the array:,prime_numbers")
```

```
⇒ random integers array:
[908 857 268 234 93 707 436 538 395 438 896 299 340 652 414 931 864 59
 373 207 877 142 209 385 15 70 86 631 670 587 221 340 305 617 243 676
 157 652 870 22 667 276 696 900 972 755 870 526 374 733 243 956 241 713
 139 13 143 678 194 281 187 268 297 872 499 558 123 870 596 926 680 765
 527 509 834 360 748 996 219 238 302 109 161 687 214 526 849 714 633 425
 292 265 413 799 988 512 113 104 923 399]
```

```
prime numbers in the array:,prime_numbers
```

#10) create a numpy array representing daily temperatures.....

```
import numpy as np
#create a numpy array representing daily temp for 30 days
np.random.seed(0)
daily_temperatures=np.random.randint(0,101,size=30)
weekly_temperatures=daily_temperatures.reshape (-1,30)
weekly_averages=np.mean(weekly_temperatures,axis=1)
#print result
print("daily temperatures for a month:\n",daily_temperatures)
print("\nweekly averages:",weekly_averages)
```

```
⇒ daily temperatures for a month:
[44 47 64 67 67 9 83 21 36 87 70 88 88 12 58 65 39 87 46 88 81 37 25 77
 72 9 20 80 69 79]
```

```
weekly averages: [57.16666667]
```

Start coding or generate with AI.