



Developer Student Clubs

Al-Azhar University

DSCOn Track

NumPy exercises



✓ what output of the following code: *

1/1

```
import numpy as np  
np.arange(1,10,2)
```

- ☒ return odd numbers in interval ✓
- ☐ return even numbers in interval
- ☐ return odd and even numbers in interval

✓ how many element in the following array: *

1/1

```
np.linspace(1,10)
```

- ☐ 10
- ☐ 20
- ☐ 49
- ☒ 50 ✓

✓ what data type of element in the following array: *

1/1

```
np.linspace(1,10)
```

☐ int

☒ float



☐ double

☐ long

✓ Which of the following can use to create a vector in specific interval?? * 1/1

☐ np.arange

☐ np.linspace

☐ np.random.randint

☒ All the above



✓ Which of the following have the same output? *

1/1

```
np.arange(1.,50.)
```

☐ A:

```
np.linspace(1,50)
```

☒ B:



```
np.arange(1.,51.)
```

☒ C:



```
np.random.randint(1.,51.)
```

☐ D:

✓ How we can create a vector with random integers with values ranging from 1 to 50 ? * 1/1

```
np.random.randint(1.,50.)
```

☐ A:

```
np.random.randint(1,50,(4,4))
```

☐ B:

```
np.random.randint(1.,51.,4)
```

☒ C:



```
np.random.randint(1.,51.)
```

☐ D:

✓ What dimension of the following array: *

1/1

```
np.random.randint(3,3,3)
```

☐ 1

☐ 2

☐ 3

☒ ValueError



✓ Consider an integer vector a , Which of the following can use to get "array([1, 2,3,4,5])" *

```
In [24]: a=np.arange(1,11)
a
```

```
Out[24]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

☐ a[0,1,2,3,4]

☒ a[:5]



☒ a[:-5]



☒ a[0:5]



- ✓ Consider an integer vector `a` , Which of the following can use to get numbers divisible of 3 ...? * 1/1

```
In [24]: a=np.arange(1,11)
a
```

```
Out[24]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

☐ `a[a%3=0]`

☒ `a[a%3==0]` ✓

☐ `a[i%3==0]`

☒ `a[2::3]` ✓

-
- ✓ Consider an integer Matrix `b` ,what output of the following code: * 1/1

```
b=np.random.randint(1,50,(5,10))
c=b[b>25]
c[c<40]
```

☐ return numbers greater than 25

☐ return numbers less than 40

☐ return numbers greater than 40

☒ return numbers greater than 25 and less than 40 ✓

- ✓ Consider an integer Matrix b , Which of the following can use to get The first three rows and The first five columns? * 1/1

```
In [59]: np.random.seed(1)
b=np.random.randint(1,50,(5,10))
b
```

```
Out[59]: array([[38, 44, 13,  9, 10, 12,  6, 16,  1, 17],
 [ 2, 13,  8, 46,  7, 26, 21, 38, 19, 21],
 [12, 43, 29, 30, 15,  5, 24, 24, 42, 31],
 [33, 23, 14, 42, 10,  8, 23,  2,  1, 18],
 [ 9, 25, 14, 48, 43,  9, 31,  8,  4,  7]])
```

☒ b[:-2,:-5] ✓

☐ b[-2,:5]

☒ b[:3,:-5] ✓

☒ b[[0,1,2],:5] ✓

- ✓ Consider an integer vector a , which of these expressions are legal? * 1/1

```
In [24]: a=np.arange(1,11)
a
```

```
Out[24]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

☐ a**a

☐ a//a

☐ a-np.log(a)

☒ All the above ✓

- ✓ Consider an integer Matrix A , Which of the following can use to swap the 1/1 first two columns? *

```
A = np.arange(12).reshape(3, 4)
A
```

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11]])
```

- ☒ A[:,[0,1]]=A[:,[1,0]] ✓
- ☐ A[[0,1]]=A[[1,0]]
- ☐ A[[0,1],:]=A[[1,0],:]
- ☐ A[0,1]=A[1,2]

- ✓ Which of the following can use to add value to array? * 1/1

- ☐ np.append(val)
- ☒ np.append(arr,val) ✓
- ☐ insert(arr,val)
- ☐ insert(val)

✓ what output of the following code: *

1/1

```
B=np.random.randint(0,12,(3,4))  
np.append(B,2)[-1]
```

☐ 0

☐ 12

☒ 2



☐ 4

✓ what output of the following code: *

1/1

```
B=np.random.randint(0,12,(3,4))  
np.insert(B,2,3)[2]
```

☒ 3



☐ 2

☐ 0

☐ 12

✓ what shape of output of the following code: *

1/1

```
B=np.random.randint(0,12,(3,4))  
np.append(B,2,axis=0)
```

☐ (4,4)

☐ (3,5)

☐ (3,4)

☒ Value Error



✓ Consider an integer Matrix B , Which of the following can use to get minimum value in B? *

1/1

```
np.random.seed(1)  
B=np.random.randint(0,12,(3,4))  
B
```

```
array([[ 5, 11,  8,  9],  
       [11,  5,  0,  0],  
       [ 1,  7,  6,  9]])
```

☒ B.min()



☒ np.min(B)



☐ min(B)

☐ None of the above

✓ Consider an integer Matrix B is the same in previous question ,what output of the following code: *

1/1

```
np.random.seed(1)
B=np.random.randint(0,12,(3,4))
np.max(B,0)
```

- ☐ 11
- ☒ array([11, 11, 8, 9])
- ☐ array([11, 11, 9])



✓ what output of the following code: *

1/1

```
B=np.arange(12).reshape(4,3)
np.mean(B,1)[0]
```

- ☐ array([1., 4., 7., 10.])
- ☐ array([4.5, 5.5, 6.5])
- ☒ 1.0
- ☐ 4.5



✓ what output of the following code: *

1/1

```
a=np.arange(12).reshape(4,3)
b=np.arange(8).reshape(4,2)
c=np.concatenate(a,b,axis=1)
c.shape
```

☐ (4, 5)

☐ (8, 5)

☐ (4, 4)

☒ Error



✓ what output of the following code: *

1/1

```
np.random.seed(1)
B=np.random.randint(0,12,(3,4))
B
array([[ 5, 11,  8,  9],
       [11,  5,  0,  0],
       [ 1,  7,  6,  9]])
```

```
np.sort(B,1)
```

```
array([[ 5,  8,  9, 11],
       [ 0,  0,  5, 11],
       [ 1,  6,  7,  9]])
```

☐ A:

```
array([[ 1,  5,  0,  0],
       [ 5,  7,  6,  9],
       [11, 11,  8,  9]])
```

☐ B:

```
array([[ 5,  8,  9, 11],
       [ 0,  0,  5, 11],
       [ 1,  6,  7,  9]])
```

☒ C:

