foo

February 10, 2021

:

1. numpy 8 ,

	(Data Type)	(Detail Bits)
1		
2		
3		
4		
5		
6		
7		
8		

2.

```
lst2 = [[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]]
arr2 = np.array(lst2)
print(len(arr2))
print(np.size(arr2))
```

3.

```
lst3 = [[1, 2], [3, 4], [5, 6]]
arr3 = np.asfarray(lst3)
print(arr3)
```

4. 4 ,

	( )	( )	(Description)
1			
2			
3			
4			

5.

```
print(np.arange(5))
print(np.arange(3, 10))
print(np.arange(2, 10, 2))
```

6.

```
arr8 = np.arange(12)
print(arr8.reshape(3, -1))
print(arr8.reshape(4, -1))
```

7.

```
arr2 = np.arange(7)
arr3 = arr2[1:5]
print(arr2)
print(arr3)
arr3[2] = 10
print(arr2)
print(arr3)
```

8.

```
subject = np.array(['math', 'math', 'lang', 'chem', 'math', 'phys', 'chem'])
data = np.arange(28).reshape(7, 4)
print(data[(subject=='math') | (subject=='chem'), :])
```

9. 1

(1) 1 10

(2) 0~10 2 4x3 12

10.

```
x = np.arange(7)
print(np.random.permutation(x))
print(np.random.shuffle(x))
```

11.

```
z = np.array([-1, 3, -5])
print(np.abs(z))
print(np.fabs(z))
print(np.sign(z))
print(z+2)
```

12.

```
a = np.array([1, 2, 3, 4])
b = np.array([[1, 2], [3, 4]])
print(np.prod(a))
print(np.prod(b, axis=0))
print(np.prod(b, axis=1))
```

13.

```
a = np.array([3.14, -5.21, 9.69, -1.87, 0])
print(np.ceil(a))
print(np.floor(a))
print(np.rint(a))
```

```
print(np.round_(a, -1))
print(np.fix(a))
print(np.trunc(a))
```

14.

```
a = np.array([1, 2, 3, 4, 5, 6, 7])
print((a>=3).any())
print((a>=3).all())
```

15.

```
a = np.array([[0, 8], [7, 7]])
b = np.array([[0, 1], [4, 2]])
print(a*b)
print(np.multiply(a, b))
print(np.dot(a, b))
```

16.

```
y = np.array([[1, 2], [3, 4]])
print(np.linalg.det(y))
a = np.array([[1, 2], [3, 4]])
b = np.array([9, 8])
x = np.linalg.solve(a, b)
print(x)
```

17.

```
a = np.array([1, 1, 3, 3, 2, 1, 2, 3, 1, 2, 3])
print(np.unique(a))
```

18.

(1)

```
a = np.arange(12).reshape((4, 3))
b = np.arange(3)
print(a+b)
```

(2)

```
a = np.arange(12).reshape((4, 3))
c = np.arange(4)
print(a+c)
```

(3)

```
a = np.arange(12).reshape((4, 3))
d = np.arange(4).reshape((4, 1))
print(a+d)
```

[ ]: