

Numpy continue

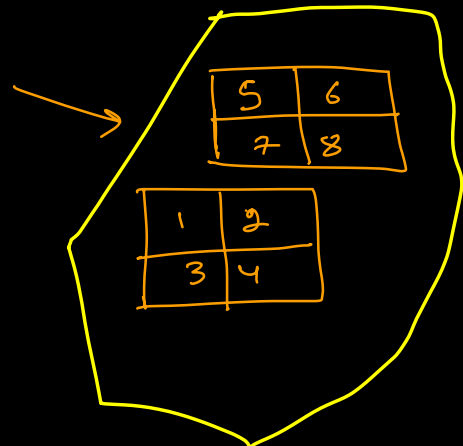
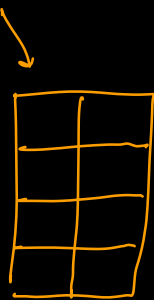
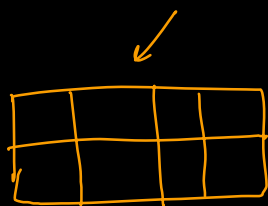
Reshape

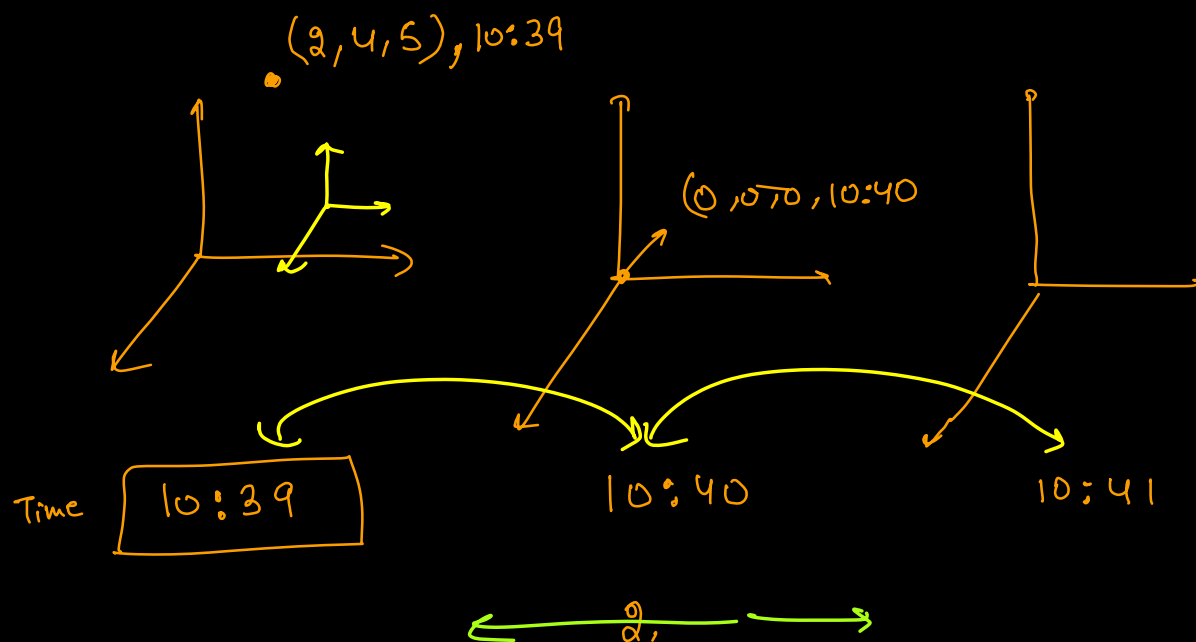
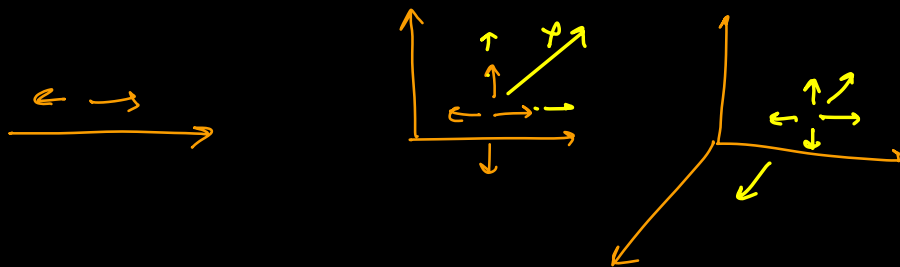
$$[1, 2, 3, 4]_{1 \times 4} \rightarrow \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}_{2 \times 2}$$

$$\rightarrow \begin{bmatrix} - & - & - \\ - & - & - \\ - & - & - \end{bmatrix}_{3 \times 3}$$

$$\underline{\text{no. of elements}} = \underline{a_1 \cdot a_2 \cdot a_3 \cdot a_4}$$

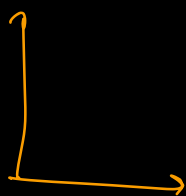
$$[1, 2, 3, 4, 5, 6, 7, 8]$$



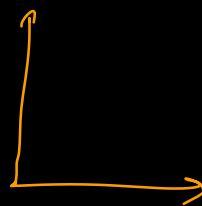




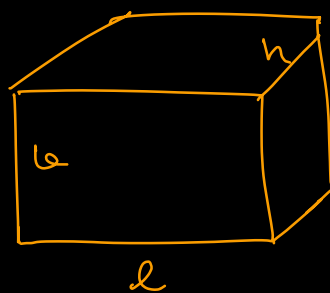
10:40



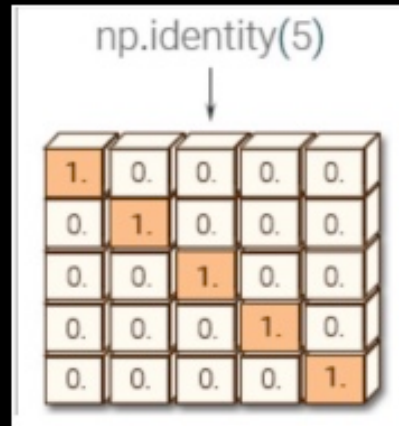
10:41



10:42



Np.zeros & Np.ones



Np.linspace



6 numbers equally spaced between
[0,10] both inclusive.

```
np.linspace (start,  
             stop,  
             num =
```

0	2	4	6	8	10	
1	2	3	4	5	6	6

Array Attributes

1. Shape
2. ndim
3. size

1 bit 0, 1
8 bit \Rightarrow 1 byte

2 bits



8 bits \Rightarrow 8 bytes

100 MBs \rightarrow 100MB
100/8

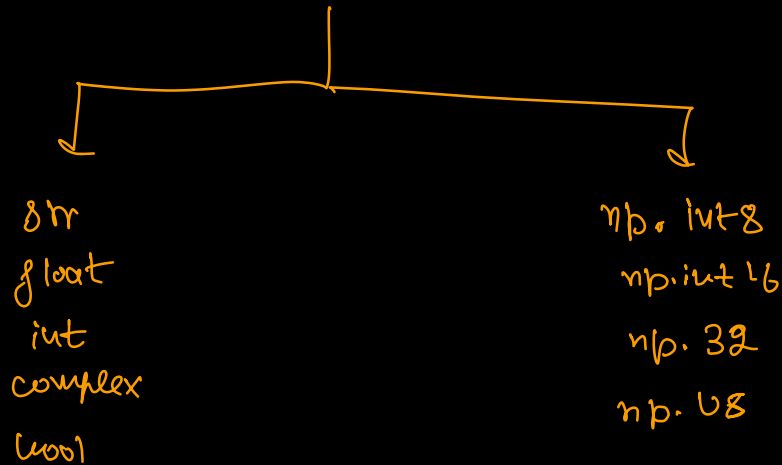
00	\rightarrow	0
01	\rightarrow	1
10	\rightarrow	2
11	\rightarrow	3

Changing Data Type

• (array).astype

Table 4-2. NumPy data types

Type	Type Code	Description
int8, uint8	i1, u1	Signed and unsigned 8-bit (1 byte) integer types
int16, uint16	i2, u2	Signed and unsigned 16-bit integer types
int32, uint32	i4, u4	Signed and unsigned 32-bit integer types
int64, uint64	i8, u8	Signed and unsigned 64-bit integer types
float16	f2	Half-precision floating point
float32	f4 or f	Standard single-precision floating point. Compatible with C float
float64, float128	f8 or d	Standard double-precision floating point. Compatible with C double and Python float object



Array Operations

Numpy Operation

$$Y = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$
$$2Y = \begin{bmatrix} 2 \times 2 & 2 \times 1 \\ 2 \times 1 & 2 \times 2 \end{bmatrix} = \begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix}$$