

Agenda



Masking [Fancy Indexing]



Routines : np.wherec(), np.unique()



NPS Solution

Airbnb NPS



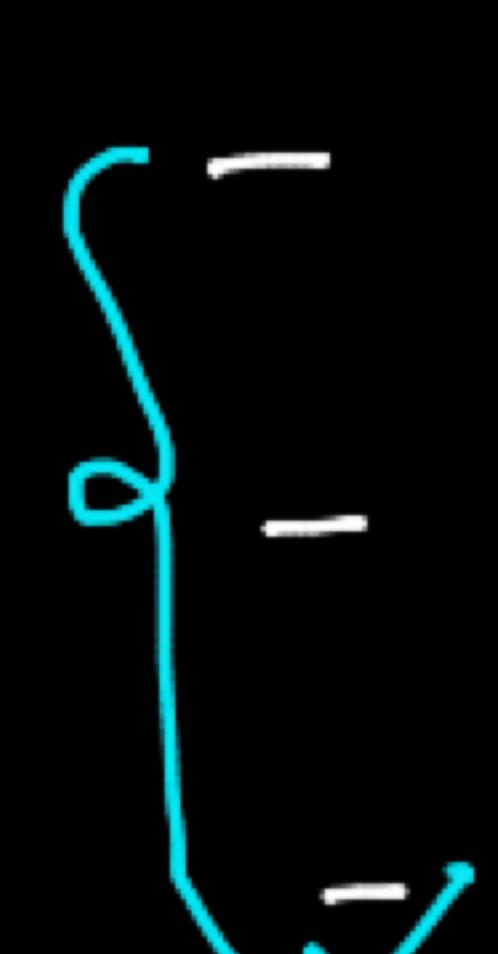
Introduction to Fitbit Usecase



Logical functions : np.any(), np.all()



2-D Arrays [Matrices]



- reshape()

- Transpose()

- Converting matrix back to vector -

1-D → vector TA Friends me

flatten()



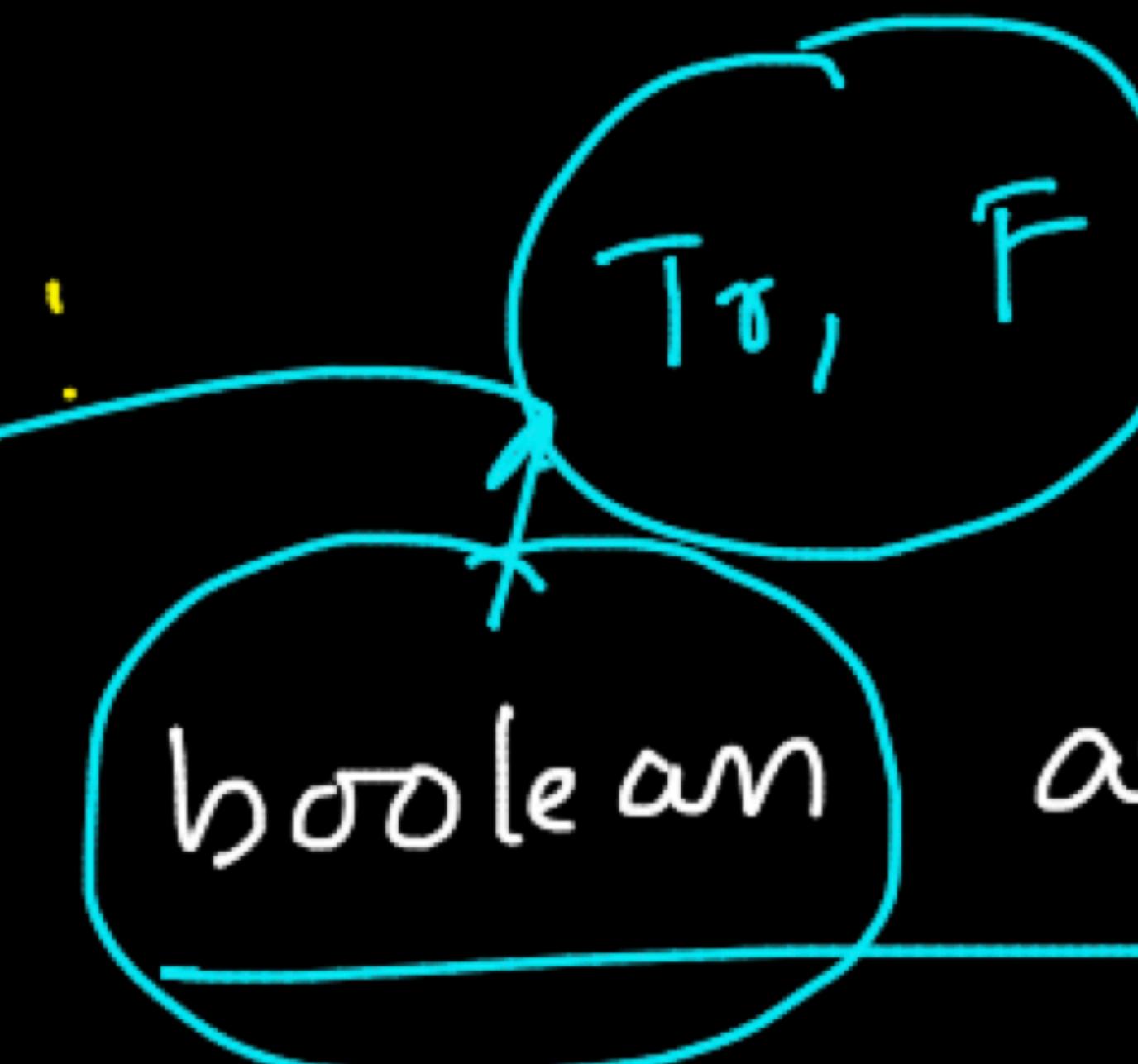
Indexing

* astype

Masking:

=

indexed with

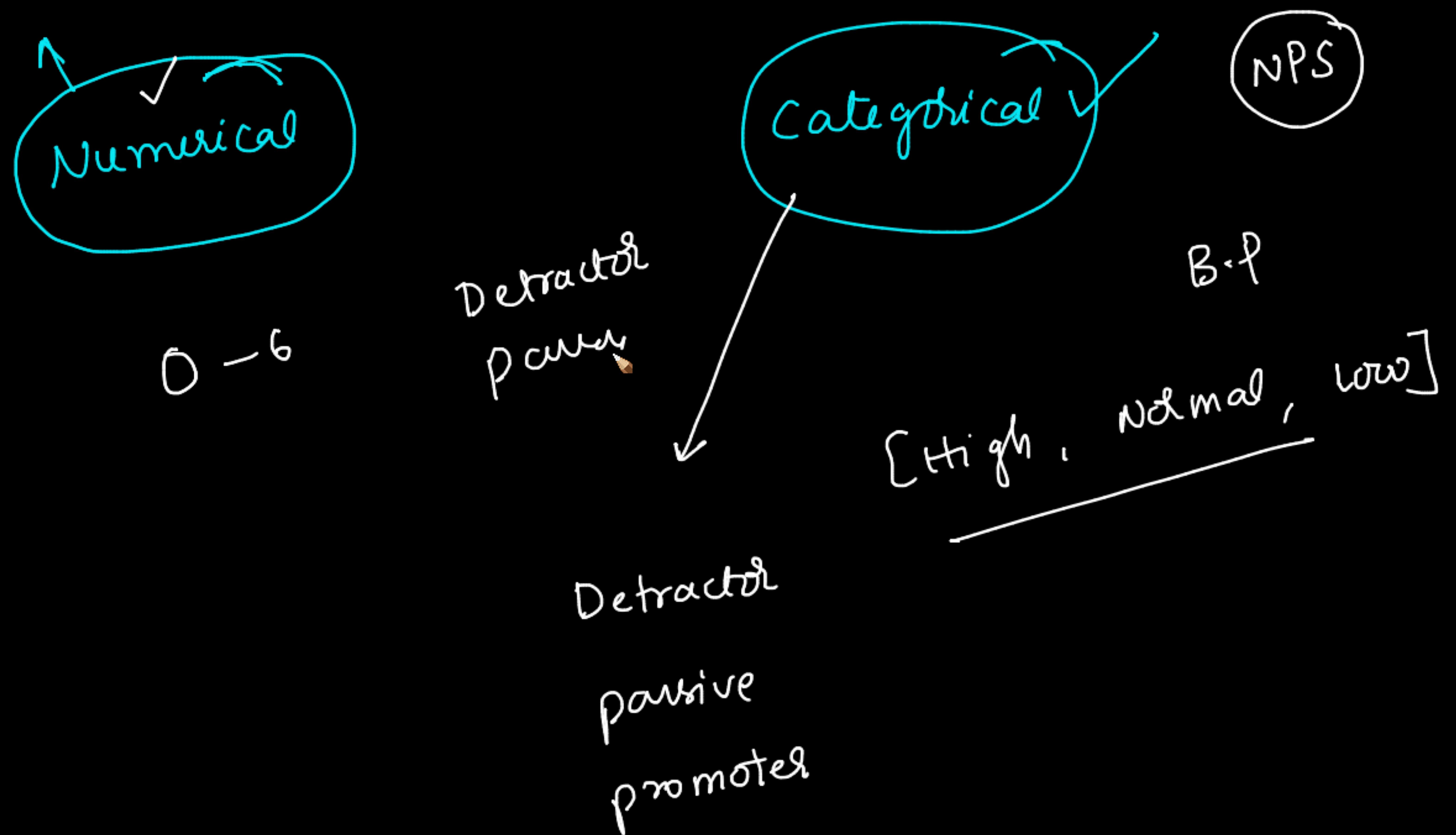


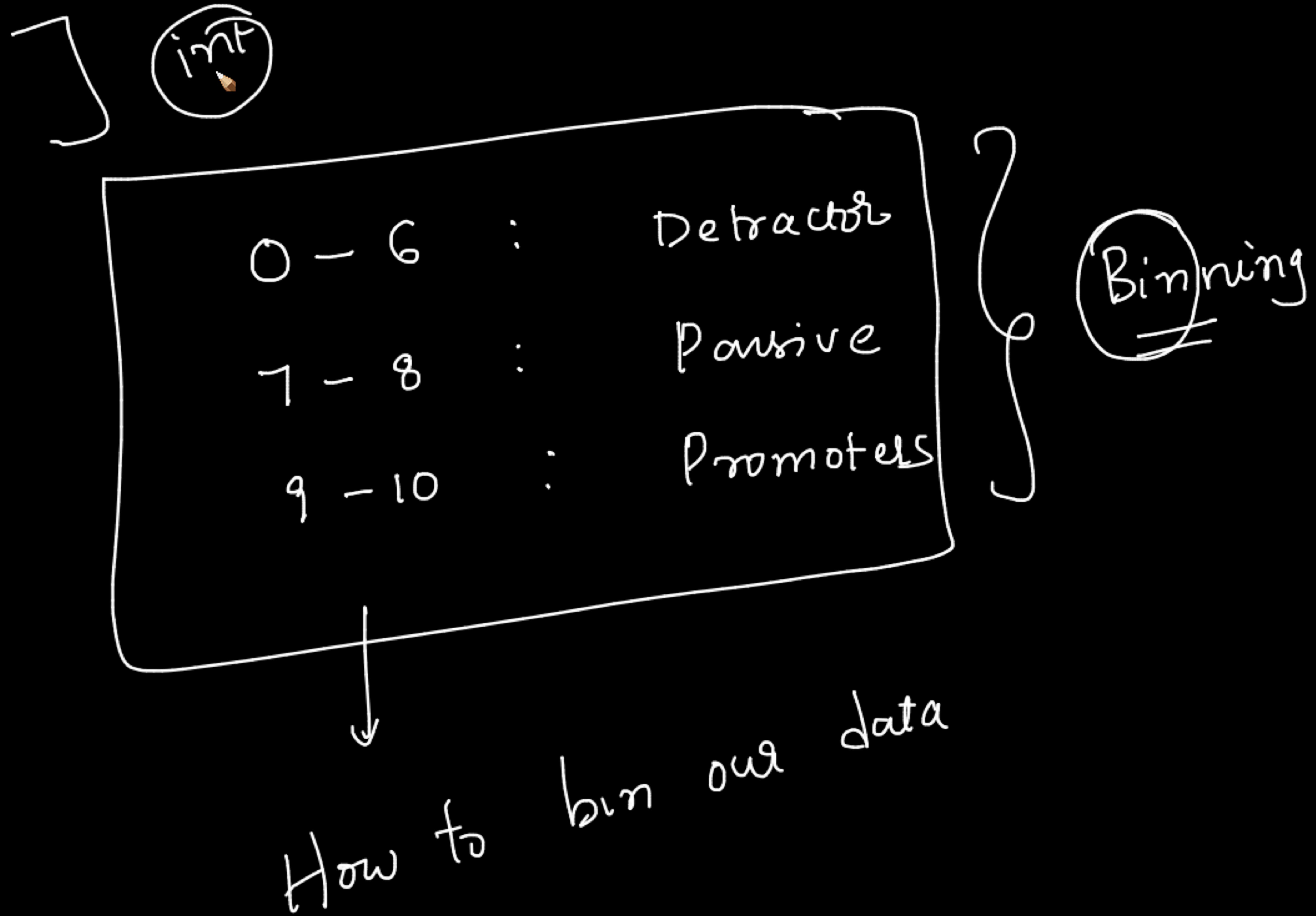
array.

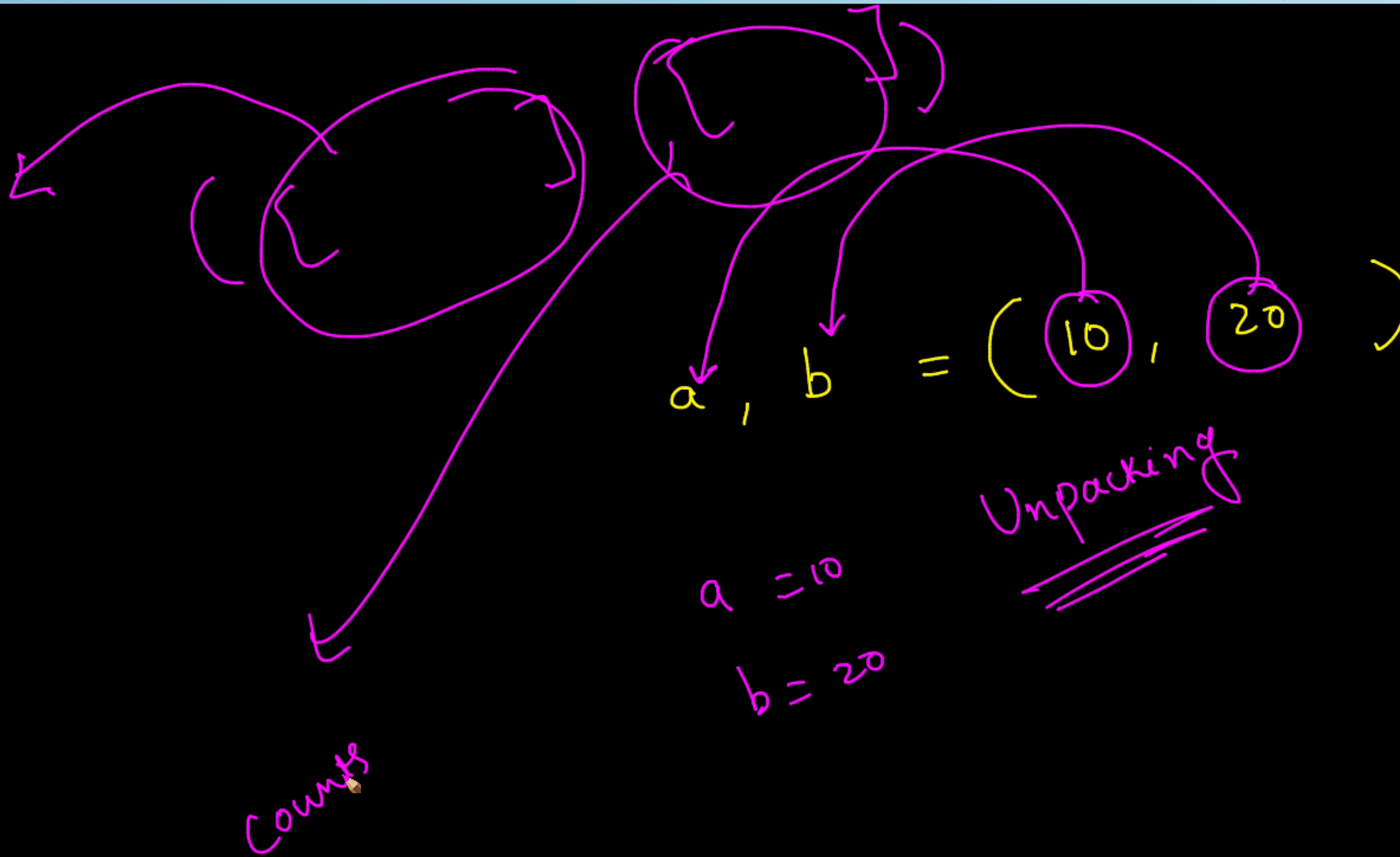
arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

arr < 6

Output: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}







Logical functions

any ()

all ()

Transpose

change rows into columns,

Transpose

(A^T)

$$A^T = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

A =

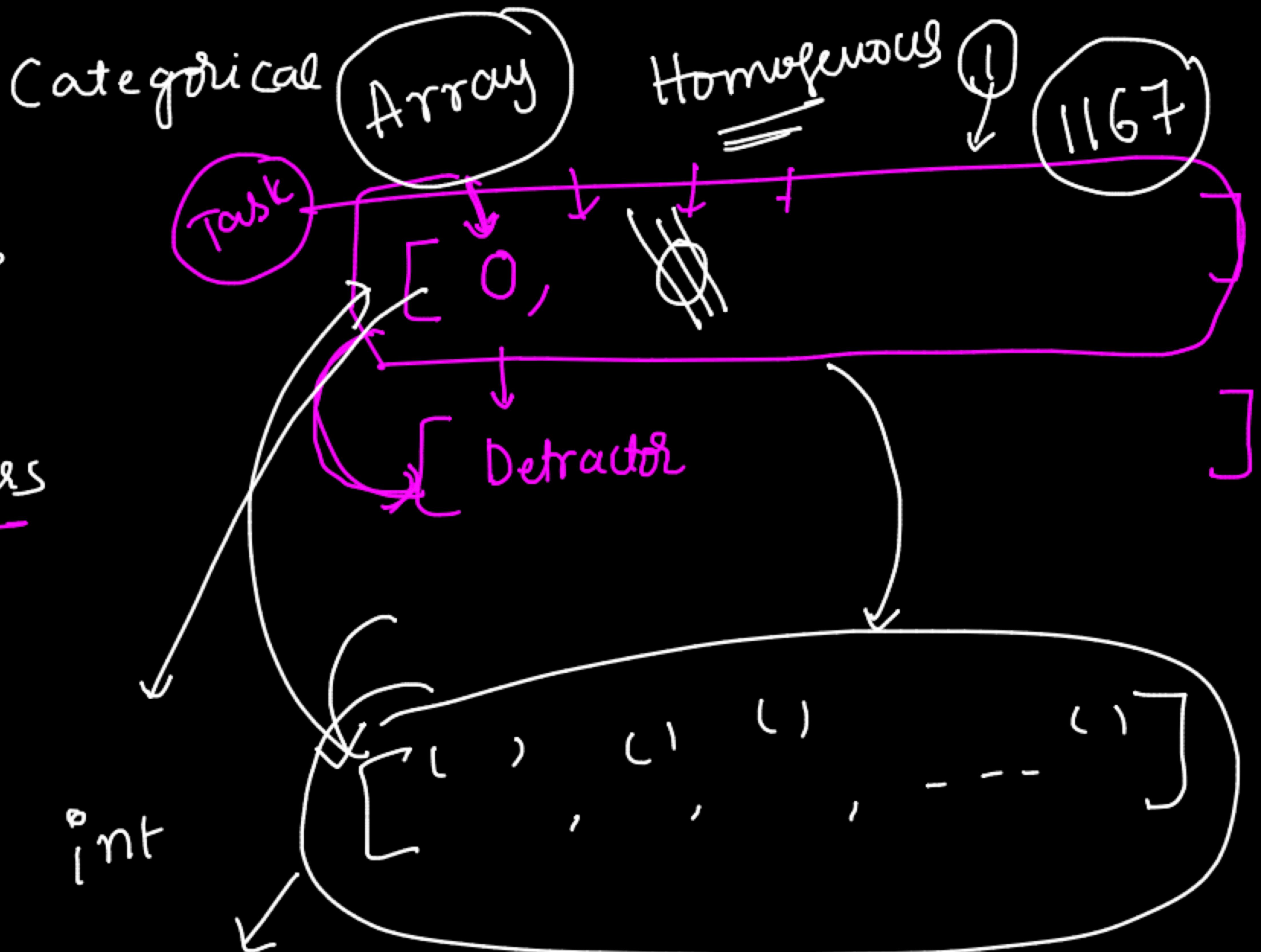
$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

3x3

0 = 6 : Detractors

7-8 : Pattine

q - 10 : Promoters



empty.

⇒ array ([]) ✓
array ([1, 2]) ✓

Cursor selected

