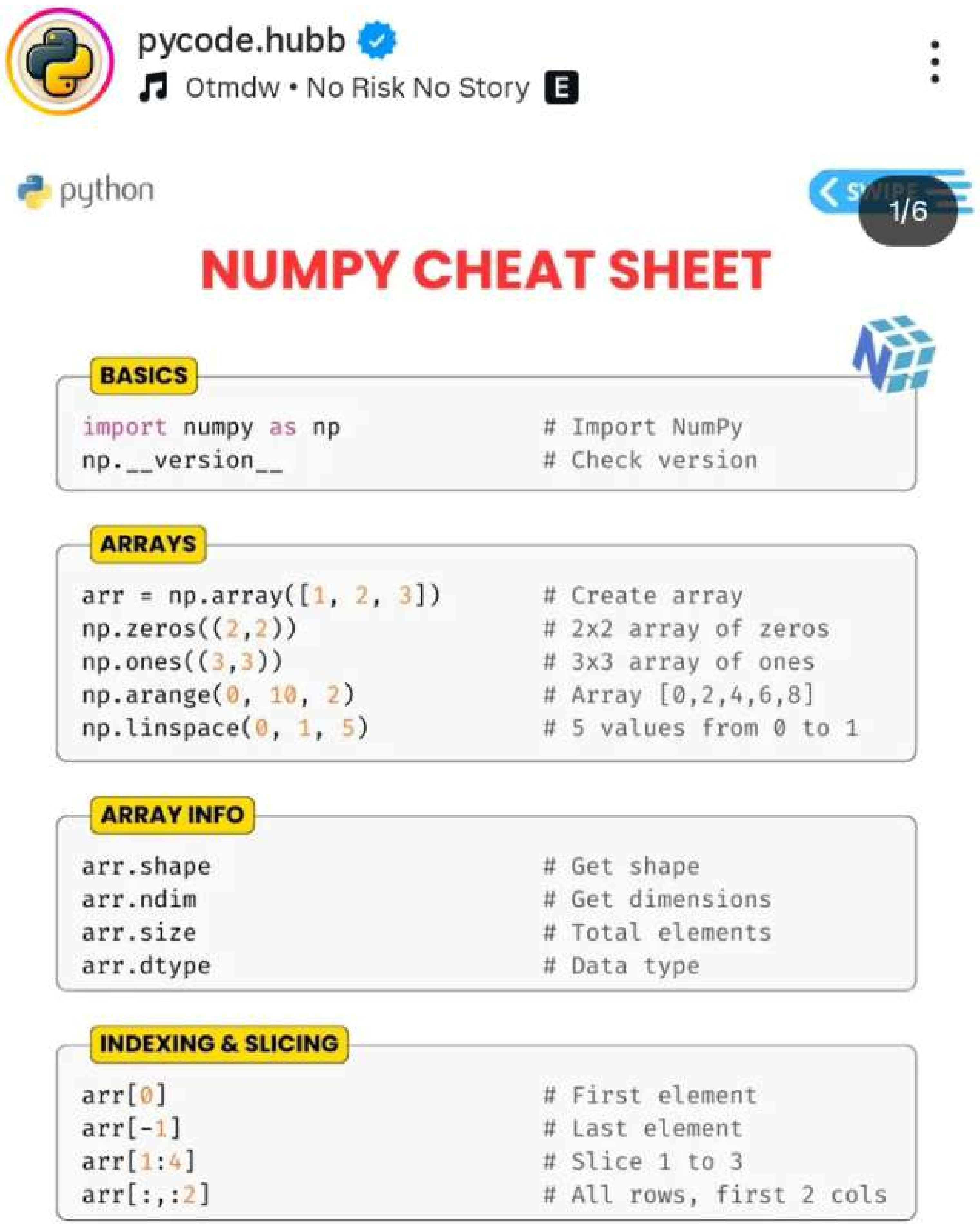


pycode.hubb

Otmdw • No Risk No Story E

NUMPY CHEAT SHEET

BASICS

```
import numpy as np                # Import NumPy
np.__version__                    # Check version
```

ARRAYS

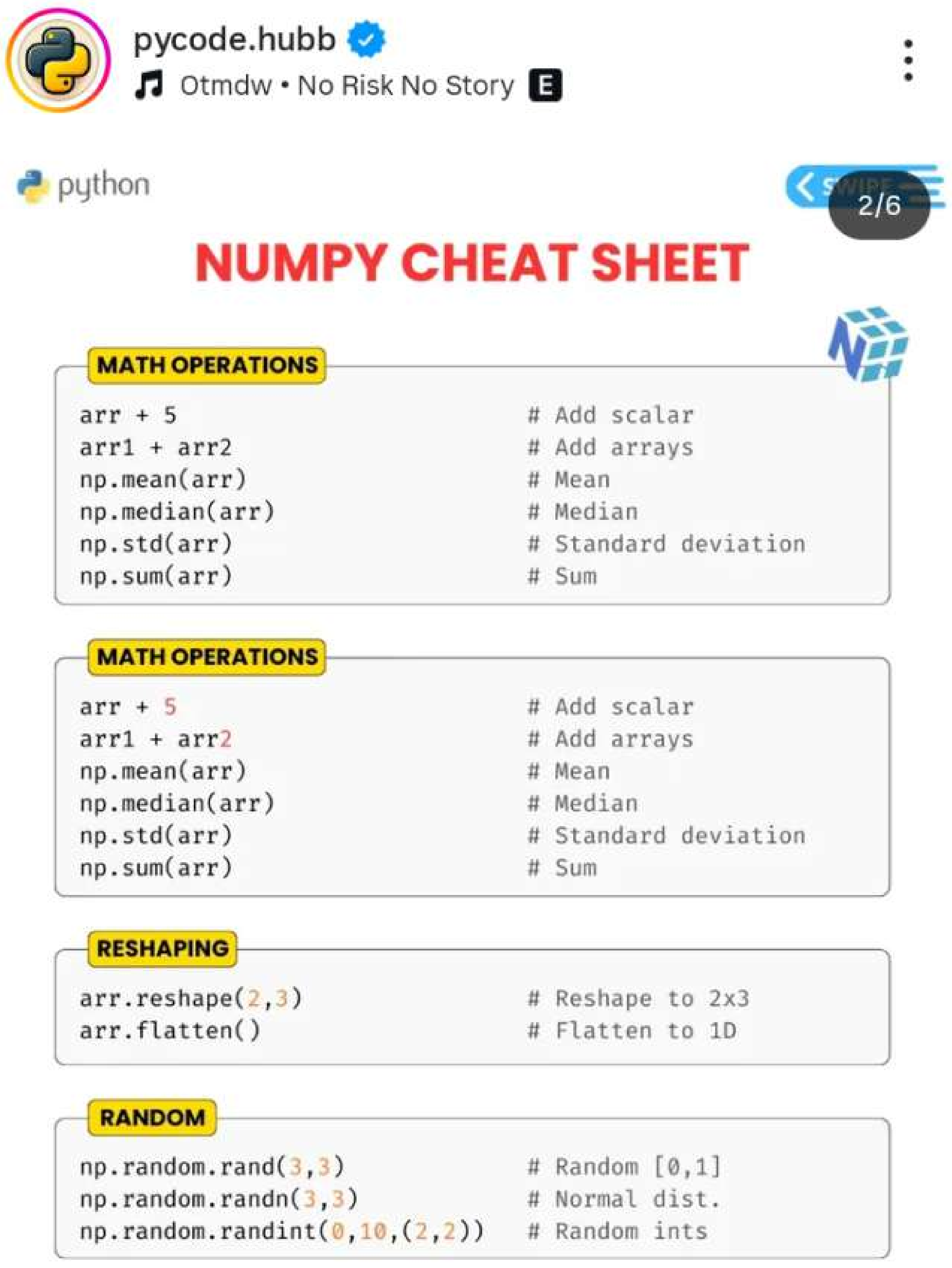
```
arr = np.array([1, 2, 3])         # Create array
np.zeros((2,2))                   # 2x2 array of zeros
np.ones((3,3))                    # 3x3 array of ones
np.arange(0, 10, 2)               # Array [0,2,4,6,8]
np.linspace(0, 1, 5)              # 5 values from 0 to 1
```

ARRAY INFO

```
arr.shape                         # Get shape
arr.ndim                          # Get dimensions
arr.size                          # Total elements
arr.dtype                         # Data type
```

INDEXING & SLICING

```
arr[0]                            # First element
arr[-1]                           # Last element
arr[1:4]                          # Slice 1 to 3
arr[:,:2]                         # All rows, first 2 cols
```



NUMPY CHEAT SHEET

MATH OPERATIONS

```
arr + 5                         # Add scalar
arr1 + arr2                     # Add arrays
np.mean(arr)                    # Mean
np.median(arr)                  # Median
np.std(arr)                     # Standard deviation
np.sum(arr)                     # Sum
```

MATH OPERATIONS

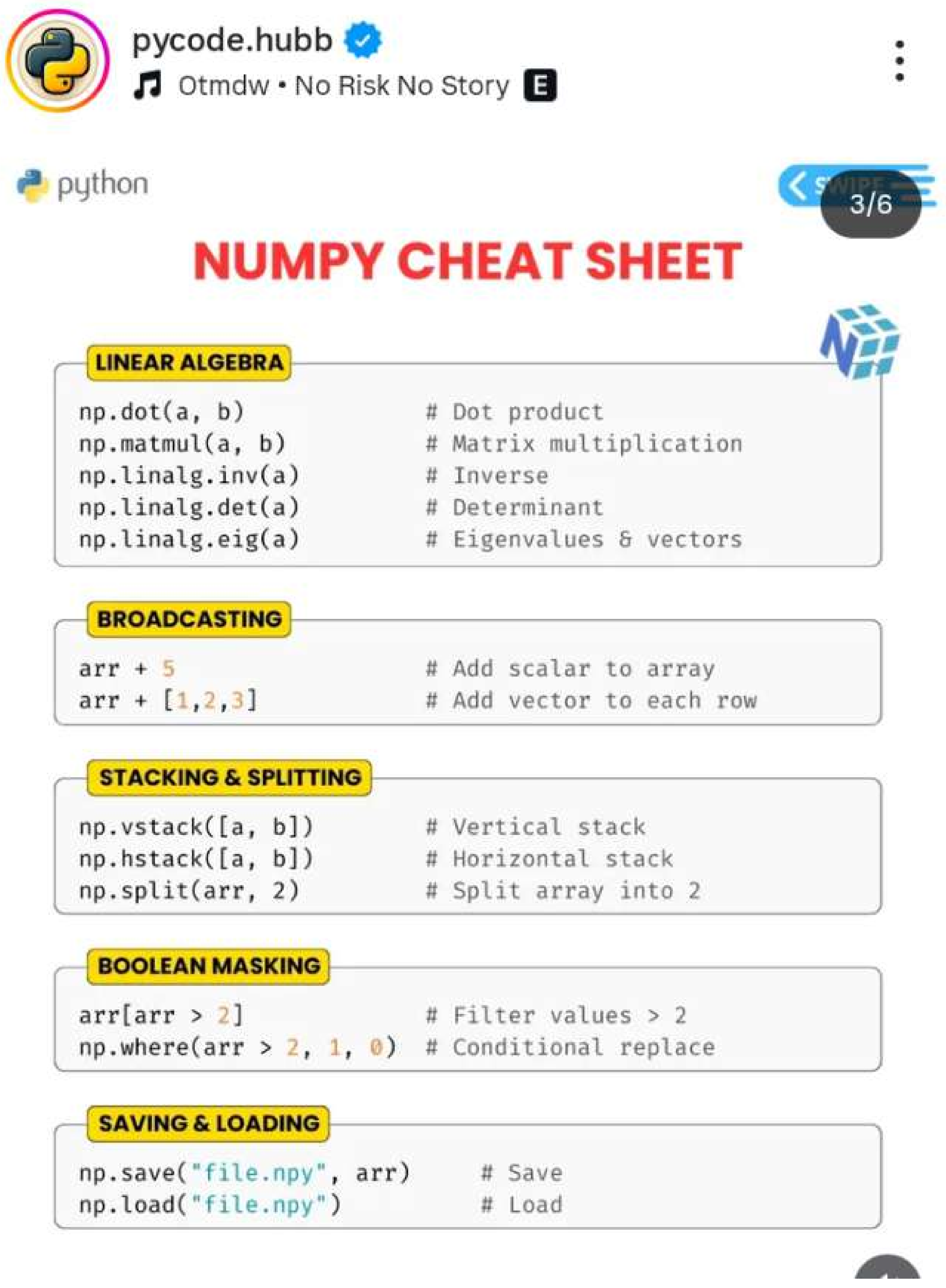
arr + 5   # Add scalar
arr1 + arr2   # Add arrays
np.mean(arr)   # Mean
np.median(arr)   # Median
np.std(arr)   # Standard deviation
np.sum(arr)   # Sum

RESHAPING

```
arr.reshape(2,3)                # Reshape to 2x3
arr.flatten()                   # Flatten to 1D
```

RANDOM

```
np.random.rand(3,3)             # Random [0,1]
np.random.randn(3,3)            # Normal dist.
np.random.randint(0,10,(2,2))   # Random ints
```



NUMPY CHEAT SHEET

LINEAR ALGEBRA

```
np.dot(a, b)             # Dot product
np.matmul(a, b)          # Matrix multiplication
np.linalg.inv(a)         # Inverse
np.linalg.det(a)         # Determinant
np.linalg.eig(a)         # Eigenvalues & vectors
```

BROADCASTING

```
arr + 5                  # Add scalar to array
arr + [1,2,3]            # Add vector to each row
```

STACKING & SPLITTING

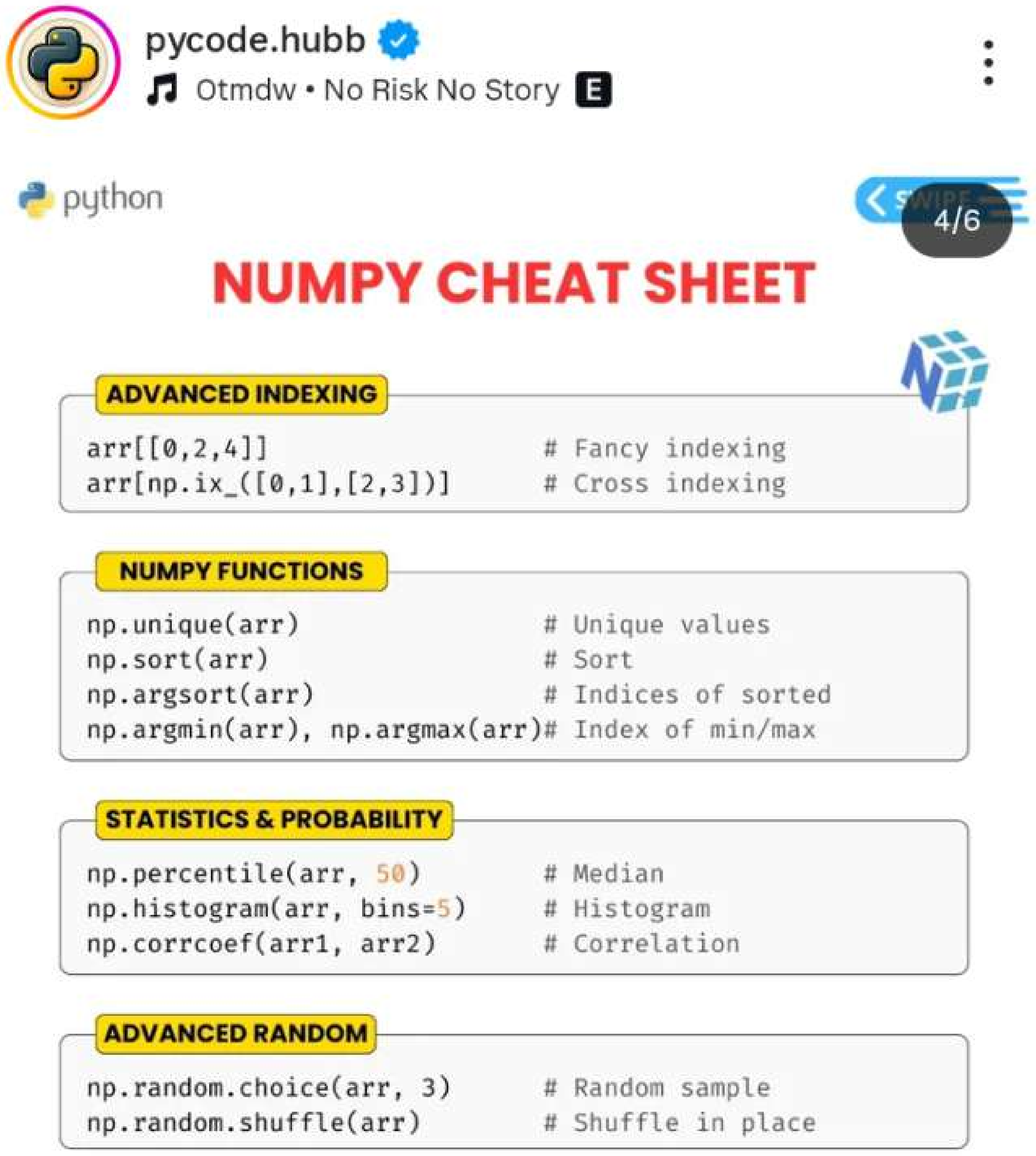
```
np.vstack([a, b])        # Vertical stack
np.hstack([a, b])        # Horizontal stack
np.split(arr, 2)         # Split array into 2
```

BOOLEAN MASKING

```
arr[arr > 2]             # Filter values > 2
np.where(arr > 2, 1, 0)  # Conditional replace
```

SAVING & LOADING

```
np.save("file.npy", arr)     # Save
np.load("file.npy")          # Load
```



pycode.hubb

Otmdw • No Risk No Story E

python

NUMPY CHEAT SHEET

ADVANCED INDEXING

```
arr[[0,2,4]]                  # Fancy indexing
arr[np.ix_([0,1],[2,3])]      # Cross indexing
```

NUMPY FUNCTIONS

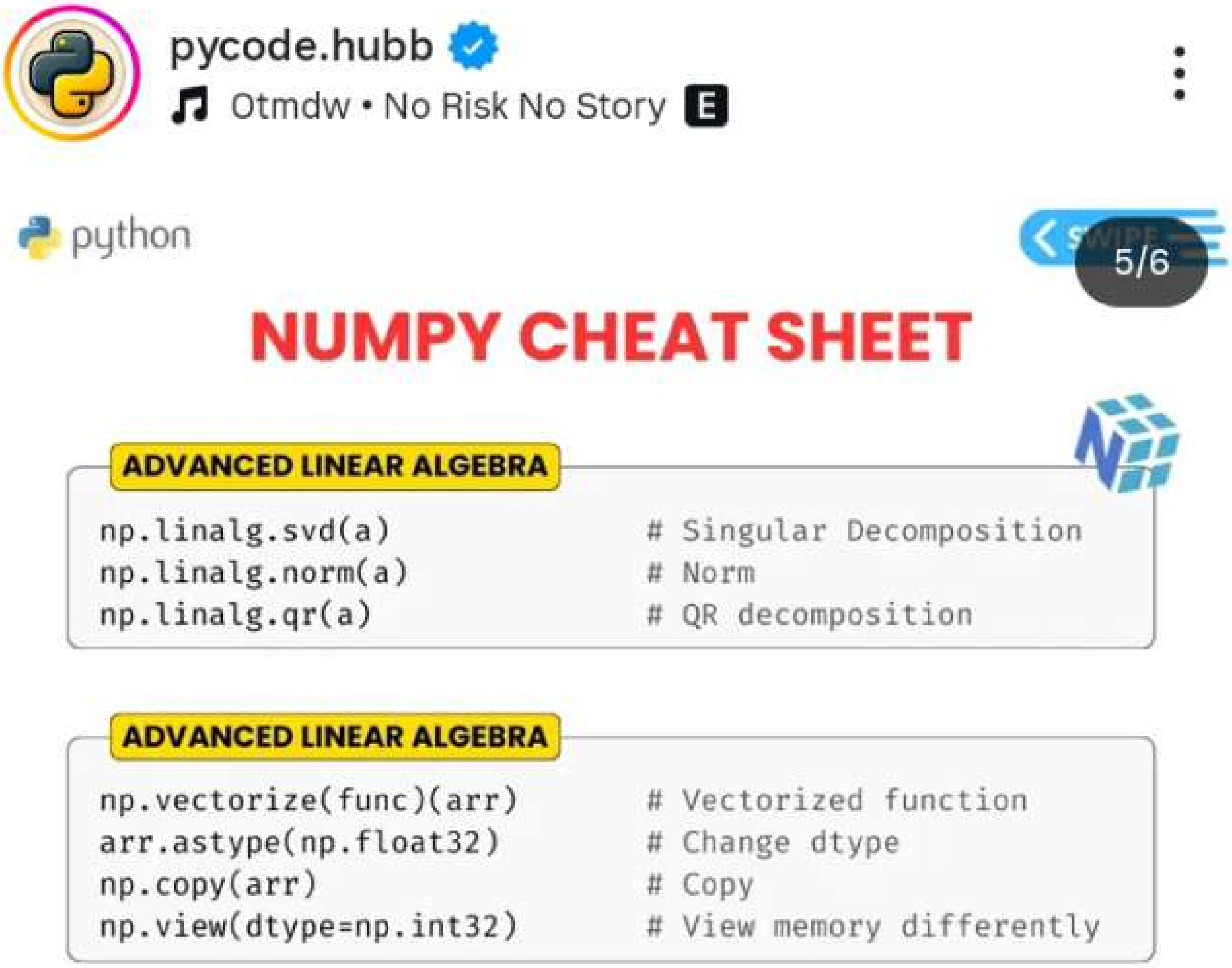
```
np.unique(arr)                # Unique values
np.sort(arr)                  # Sort
np.argsort(arr)               # Indices of sorted
np.argmin(arr), np.argmax(arr)# Index of min/max
```

STATISTICS & PROBABILITY

```
np.percentile(arr, 50)        # Median
np.histogram(arr, bins=5)     # Histogram
np.corrcoef(arr1, arr2)       # Correlation
```

ADVANCED RANDOM

```
np.random.choice(arr, 3)      # Random sample
np.random.shuffle(arr)        # Shuffle in place
```



NUMPY CHEAT SHEET

ADVANCED LINEAR ALGEBRA

```
np.linalg.svd(a)                 # Singular Decomposition
np.linalg.norm(a)                # Norm
np.linalg.qr(a)                  # QR decomposition
```

ADVANCED LINEAR ALGEBRA

```
np.vectorize(func)(arr)          # Vectorized function
arr.astype(np.float32)           # Change dtype
np.copy(arr)                     # Copy
np.view(dtype=np.int32)          # View memory differently
```