NumPy Array Indexing and Slicing

Basic Indexing - 1D and 2D

```
import numpy as np

# 1D array indexing
arr_1d = np.array([10, 20, 30, 40, 50])
print("Element at index 2:", arr_1d[2])
print("Last element:", arr_1d[-1])
print("Second to last:", arr_1d[-2])

# 2D array indexing
arr_2d = np.array([[1, 2, 3, 4],
                   [5, 6, 7, 8],
                   [9, 10, 11, 12]])

print("\nElement at [1, 2]:", arr_2d[1, 2])  # Row 1, Column 2
print("First row:", arr_2d[0])
print("Last column:", arr_2d[:, -1])
```

Slicing Arrays

```
# 1D slicing
arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
print("Elements 2-5:", arr[2:6])
print("Every other element:", arr[::2])
print("Reverse array:", arr[::-1])

# 2D slicing
arr_2d = np.array([[1, 2, 3, 4],
                   [5, 6, 7, 8],
                   [9, 10, 11, 12]])

print("\nFirst 2 rows, columns 1-3:")
print(arr_2d[:2, 1:3])

print("\nAll rows, first 2 columns:")
print(arr_2d[:, :2])

print("\nLast 2 rows, last 2 columns:")
print(arr_2d[-2:, -2:])
```

Boolean Indexing

```
arr = np.array([[1, 2, 3, 4],
                [5, 6, 7, 8],
                [9, 10, 11, 12]])

# Boolean mask
mask = arr > 6
print("Mask (elements > 6):\n", mask)
print("\nElements > 6:", arr[mask])

# Multiple conditions
print("\nElements > 5 AND < 10:")
print(arr[(arr > 5) & (arr < 10)])

# Modify using boolean indexing
arr_copy = arr.copy()
```

```
arr_copy[arr_copy > 8] = 0
print("\nSet elements > 8 to 0:\n", arr_copy)
```

Fancy Indexing

```
arr = np.array([[1, 2, 3, 4],
                [5, 6, 7, 8],
                [9, 10, 11, 12]])

# Index with lists
print("Rows [0, 2]:")
print(arr[[0, 2]])

print("\nRows [0, 2], columns [1, 3]:")
print(arr[[0, 2]][:, [1, 3]])

# More complex indexing
rows = np.array([0, 1, 2])
cols = np.array([1, 2, 3])
print("\nDiagonal-like selection:")
print(arr[rows, cols])  # Elements [0,1], [1,2], [2,3]
```

Advanced Indexing Techniques

```
# Using np.where
arr = np.array([1, 6, 3, 8, 2, 9, 4])
indices = np.where(arr > 5)
print("Indices where arr > 5:", indices[0])
print("Values:", arr[indices])

# Using np.argmax and np.argmin
print("\nIndex of maximum:", np.argmax(arr))
print("Index of minimum:", np.argmin(arr))

# Indexing with conditions
arr_2d = np.array([[1, 2, 3],
                   [4, 5, 6],
                   [7, 8, 9]])

# Get row and column indices
rows, cols = np.where(arr_2d > 5)
print("\nPositions where arr_2d > 5:")
for r, c in zip(rows, cols):
    print(f"  [{r}, {c}] = {arr_2d[r, c]}")
```