NumPy Arrays Assignment

Mustafa AbdulRazek

Task 1: Creating and Inspecting Arrays

Objective: Learn how to create NumPy arrays and inspect their attributes.

Task Description

NumPy arrays are powerful tools for data manipulation in Python. In this task, you will create one-dimensional, two-dimensional, and three-dimensional arrays and explore their attributes such as dimensions, shape, size, and data type.

Task

Write a Python script to:

- Create a one-dimensional array with 6 random integers.
- Create a two-dimensional array with random integers of size 3x4.
- Create a three-dimensional array with random integers of size 3x4x5.
- Print the dimensions (`ndim`), shape (`shape`), and size (`size`) of each array.
- Print the data type (`dtype`) and total memory size (`nbytes`) of the three-dimensional array.

Example Output

```
x3 ndim: 3
x3 shape: (3, 4, 5)
x3 size: 60
dtype: int64
nbytes: 480 bytes
```

Task 2: Accessing and Modifying Array Elements

Objective: Practice accessing and modifying elements in NumPy arrays.

Task Description

Array indexing allows you to access specific elements within an array. You can also modify the values of elements using their indices. In this task, you will practice accessing and modifying elements of one-dimensional and two-dimensional arrays.

Task

Write a Python script to:

- Create a one-dimensional array and access the first and last elements.
- Create a two-dimensional array and access specific elements using indices.
- Modify an element in the two-dimensional array and observe the changes.

Example Output

```
Original array:
[[3, 5, 2, 4],
 [7, 6, 8, 8],
 [1, 6, 7, 7]]

Modified array:
[[12, 5, 2, 4],
 [7, 6, 8, 8],
 [1, 6, 7, 7]]
```

Task 3: Slicing and Subarrays

Objective: Learn to extract subarrays using slicing.

Task Description

Slicing allows you to extract specific subarrays from larger arrays. This task will help you practice slicing one-dimensional and two-dimensional arrays using various start, stop, and step parameters.

Task

Write a Python script to:

- Create a one-dimensional array and extract the first five elements, the last five elements, and a middle segment.
- Create a two-dimensional array and extract specific rows and columns using slicing.

Example Output

```
Original array: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
Subarray: [0, 1, 2, 3, 4]
```

Task 4: Reshaping Arrays

Objective: Practice reshaping arrays into different dimensions.

Task Description

Reshaping allows you to change the shape of an existing array without changing its data. This task will help you understand how to reshape arrays and convert 1D arrays into 2D row and column vectors.

Task

Write a Python script to:

- Create a one-dimensional array of 9 elements and reshape it into a 3x3 grid.
- Convert a one-dimensional array into a row vector and a column vector.

Example Output

```
Original array: [1, 2, 3, 4, 5, 6, 7, 8, 9]
Reshaped array:
[[1, 2, 3],
 [4, 5, 6],
 [7, 8, 9]]
```

Task 5: Combining and Splitting Arrays

Objective: Learn to combine and split arrays using NumPy functions.

Task Description

Combining and splitting arrays are essential operations for data manipulation. In this task, you will learn to concatenate arrays and split a single array into multiple subarrays.

Task

Write a Python script to:

- Create two one-dimensional arrays and concatenate them into a single array.
- Split a one-dimensional array into three smaller arrays using specified indices.
- Create a two-dimensional array and split it vertically and horizontally.

Example Output

```
Original array: [1, 2, 3, 99, 99, 3, 2, 1]
Split arrays: [1, 2, 3], [99, 99], [3, 2, 1]
```