

Practice Exercise 3: Multiplication Table

Problem: Create Multiplication Table

```
import numpy as np

# Problem: Create a 10x10 multiplication table using broadcasting
# Result should be:
# [[1, 2, 3, ..., 10],
#  [2, 4, 6, ..., 20],
#  ...
#  [10, 20, 30, ..., 100]]
```

Solution: Using Broadcasting

```
# Create row and column vectors
rows = np.arange(1, 11).reshape(-1, 1) # Column vector
cols = np.arange(1, 11) # Row vector

# Multiplication table via broadcasting
table = rows * cols

print("Multiplication table:")
print(table)
```

Alternative Solutions

```
# Method 2: Using outer product
table2 = np.outer(np.arange(1, 11), np.arange(1, 11))
print("Using outer product:")
print(table2)

# Method 3: Using meshgrid
x, y = np.meshgrid(np.arange(1, 11), np.arange(1, 11))
table3 = x * y
print("\nUsing meshgrid:")
print(table3)

# Verify all methods give same result
print("\nAll methods equal:",
      np.array_equal(table, table2) and
      np.array_equal(table2, table3))
```

Visualizing the Multiplication Table

```
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 8))
plt.imshow(table, cmap='viridis', aspect='auto')
plt.colorbar(label='Product')
plt.title('Multiplication Table (1-10)', fontsize=14, fontweight='bold')
plt.xlabel('Multiplicand', fontsize=12)
plt.ylabel('Multiplier', fontsize=12)
plt.xticks(range(10), range(1, 11))
plt.yticks(range(10), range(1, 11))
# Add annotations for first 5x5
```

```

for i in range(5):
    for j in range(5):
        plt.text(j, i, str(table[i, j]),
                 ha="center", va="center",
                 color="white", fontsize=10)

plt.tight_layout()
plt.show()

```

Extended: Custom Range and Step

```

def create_multiplication_table(start=1, stop=11, step=1):
    """Create multiplication table with custom range"""
    values = np.arange(start, stop, step)
    rows = values.reshape(-1, 1)
    cols = values
    return rows * cols

# Example: Even numbers 2-20
even_table = create_multiplication_table(2, 21, 2)
print("Even numbers multiplication table (2-20):")
print(even_table)

# Example: Decimals
decimal_table = create_multiplication_table(0.1, 1.1, 0.1)
print("\nDecimal multiplication table (0.1-1.0):")
print(np.round(decimal_table, 2))

```