modules-practice\numpy-demo.py

```python
# numpy_learning.py

import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# Basic Section
print("### Basic Section ###")

# Introduction
print("\n## Introduction ##")
# NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices.
# It also includes a large collection of high-level mathematical functions to operate on these arrays.

# Getting Started
print("\n## Getting Started ##")
arr = np.array([1, 2, 3, 4, 5])
print("Array:", arr)

# Creating Arrays
print("\n## Creating Arrays ##")
arr = np.array([1, 2, 3, 4, 5])
print("Array:", arr)

# Array Indexing
print("\n## Array Indexing ##")
print("First element:", arr[0])
print("Third element:", arr[2])

# Array Slicing
print("\n## Array Slicing ##")
print("Elements from index 1 to 3:", arr[1:4])

# Data Types
print("\n## Data Types ##")
print("Data type of array:", arr.dtype)

# Copy vs View
print("\n## Copy vs View ##")
arr_copy = arr.copy()
arr_view = arr.view()
print("Copy:", arr_copy)
print("View:", arr_view)

# Array Shape
print("\n## Array Shape ##")
arr_2d = np.array([[1, 2, 3], [4, 5, 6]])
print("Shape of 2D array:", arr_2d.shape)

# Array Reshape
```

```
print("\n## Array Reshape ##")
arr_reshaped = arr.reshape(5, 1)
print("Reshaped array:\n", arr_reshaped)

# Array Iterating
print("\n## Array Iterating ##")
print("Iterating through array:")
for element in arr:
    print(element)

# Array Join
print("\n## Array Join ##")
arr1 = np.array([1, 2, 3])
arr2 = np.array([4, 5, 6])
arr_joined = np.concatenate((arr1, arr2))
print("Joined array:", arr_joined)

# Array Split
print("\n## Array Split ##")
arr_split = np.array_split(arr_joined, 3)
print("Split array:", arr_split)

# Array Search
print("\n## Array Search ##")
print("Index of value 3:", np.where(arr == 3))

# Array Sort
print("\n## Array Sort ##")
arr_unsorted = np.array([3, 1, 2])
arr_sorted = np.sort(arr_unsorted)
print("Sorted array:", arr_sorted)

# Array Filter
print("\n## Array Filter ##")
filtered_arr = arr[arr > 2]
print("Filtered array (elements greater than 2):", filtered_arr)

# Random Section
print("\n### Random Section ###")

# Random Intro
print("\n## Random Intro ##")
random_arr = np.random.randint(0, 10, 5)
print("Random array:", random_arr)

# Data Distribution
print("\n## Data Distribution ##")
uniform_dist = np.random.uniform(0, 1, 1000)
plt.hist(uniform_dist, bins=30)
plt.title("Uniform Distribution")
plt.show()

# Random Permutation
print("\n## Random Permutation ##")
```

```
permuted_arr = np.random.permutation(arr)
print("Permuted array:", permuted_arr)

# Seaborn Module
print("\n## Seaborn Module ##")
sns.distplot(uniform_dist, hist=False)
plt.title("Seaborn Distribution Plot")
plt.show()

# Normal Dist.
print("\n## Normal Dist. ##")
normal_dist = np.random.normal(0, 1, 1000)
plt.hist(normal_dist, bins=30)
plt.title("Normal Distribution")
plt.show()

# Binomial Dist.
print("\n## Binomial Dist. ##")
binomial_dist = np.random.binomial(10, 0.5, 1000)
plt.hist(binomial_dist, bins=30)
plt.title("Binomial Distribution")
plt.show()

# Poisson Dist.
print("\n## Poisson Dist. ##")
poisson_dist = np.random.poisson(5, 1000)
plt.hist(poisson_dist, bins=30)
plt.title("Poisson Distribution")
plt.show()

# Uniform Dist.
print("\n## Uniform Dist. ##")
uniform_dist = np.random.uniform(0, 1, 1000)
plt.hist(uniform_dist, bins=30)
plt.title("Uniform Distribution")
plt.show()

# Logistic Dist.
print("\n## Logistic Dist. ##")
logistic_dist = np.random.logistic(0, 1, 1000)
plt.hist(logistic_dist, bins=30)
plt.title("Logistic Distribution")
plt.show()

# Multinomial Dist.
print("\n## Multinomial Dist. ##")
multinomial_dist = np.random.multinomial(20, [1/6.]*6, size=1)
print("Multinomial distribution:", multinomial_dist)

# Exponential Dist.
print("\n## Exponential Dist. ##")
exponential_dist = np.random.exponential(1, 1000)
plt.hist(exponential_dist, bins=30)
plt.title("Exponential Distribution")
```

```
plt.show()

# Chi Square Dist.
print("\n## Chi Square Dist. ##")
chi_square_dist = np.random.chisquare(2, 1000)
plt.hist(chi_square_dist, bins=30)
plt.title("Chi Square Distribution")
plt.show()

# Rayleigh Dist.
print("\n## Rayleigh Dist. ##")
rayleigh_dist = np.random.rayleigh(1, 1000)
plt.hist(rayleigh_dist, bins=30)
plt.title("Rayleigh Distribution")
plt.show()

# Pareto Dist.
print("\n## Pareto Dist. ##")
pareto_dist = np.random.pareto(3, 1000)
plt.hist(pareto_dist, bins=30)
plt.title("Pareto Distribution")
plt.show()

# Zipf Dist.
print("\n## Zipf Dist. ##")
zipf_dist = np.random.zipf(2, 1000)
plt.hist(zipf_dist, bins=30)
plt.title("Zipf Distribution")
plt.show()

# ufunc Section
print("\n### ufunc Section ###")

# ufunc Intro
print("\n## ufunc Intro ##")
# Universal functions (ufuncs) are functions that operate on ndarrays in an element-by-element fashion.

# Create Function
print("\n## Create Function ##")
def my_ufunc(x):
    return x + 1

my_ufunc = np.frompyfunc(my_ufunc, 1, 1)
print("Custom ufunc output:", my_ufunc(arr))

# Simple Arithmetic
print("\n## Simple Arithmetic ##")
print("Addition:", np.add(arr, arr))
print("Subtraction:", np.subtract(arr, arr))
print("Multiplication:", np.multiply(arr, arr))
print("Division:", np.divide(arr, arr))

# Rounding Decimals
```

```
print("\n## Rounding Decimals ##")
arr_float = np.array([1.234, 2.567, 3.891])
print("Rounded array:", np.around(arr_float, 2))

# Logs
print("\n## Logs ##")
print("Natural log:", np.log(arr))
print("Base 10 log:", np.log10(arr))

# Summations
print("\n## Summations ##")
print("Sum of array:", np.sum(arr))

# Products
print("\n## Products ##")
print("Product of array:", np.prod(arr))

# Differences
print("\n## Differences ##")
print("Differences in array:", np.diff(arr))

# Finding LCM
print("\n## Finding LCM ##")
print("LCM of 4 and 6:", np.lcm(4, 6))

# Finding GCD
print("\n## Finding GCD ##")
print("GCD of 8 and 12:", np.gcd(8, 12))

# Trigonometric
print("\n## Trigonometric ##")
print("Sine of array:", np.sin(arr))
print("Cosine of array:", np.cos(arr))

# Hyperbolic
print("\n## Hyperbolic ##")
print("Hyperbolic sine of array:", np.sinh(arr))
print("Hyperbolic cosine of array:", np.cosh(arr))

# Set Operations
print("\n## Set Operations ##")
arr_set1 = np.array([1, 2, 3, 4])
arr_set2 = np.array([3, 4, 5, 6])
print("Union:", np.union1d(arr_set1, arr_set2))
print("Intersection:", np.intersect1d(arr_set1, arr_set2))
print("Difference:", np.setdiff1d(arr_set1, arr_set2))
```