

## modules-practice\numpy-demo.py

```
1 # numpy_learning.py
2
3 import numpy as np
4 import seaborn as sns
5 import matplotlib.pyplot as plt
6
7 # Basic Section
8 print("### Basic Section ###")
9
10 # Introduction
11 print("\n## Introduction ##")
12 # NumPy is a library for the Python programming language, adding support for large, multi-
13 # dimensional arrays and matrices.
14 # It also includes a large collection of high-level mathematical functions to operate on
15 # these arrays.
16
17 # Getting Started
18 print("\n## Getting Started ##")
19 arr = np.array([1, 2, 3, 4, 5])
20 print("Array:", arr)
21
22 # Creating Arrays
23 print("\n## Creating Arrays ##")
24 arr = np.array([1, 2, 3, 4, 5])
25 print("Array:", arr)
26
27 # Array Indexing
28 print("\n## Array Indexing ##")
29 print("First element:", arr[0])
30 print("Third element:", arr[2])
31
32 # Array Slicing
33 print("\n## Array Slicing ##")
34 print("Elements from index 1 to 3:", arr[1:4])
35
36 # Data Types
37 print("\n## Data Types ##")
38 print("Data type of array:", arr.dtype)
39
40 # Copy vs View
41 print("\n## Copy vs View ##")
42 arr_copy = arr.copy()
43 arr_view = arr.view()
44 print("Copy:", arr_copy)
45 print("View:", arr_view)
46
47 # Array Shape
48 print("\n## Array Shape ##")
49 arr_2d = np.array([[1, 2, 3], [4, 5, 6]])
50 print("Shape of 2D array:", arr_2d.shape)
51
52 # Array Reshape
```

```
51 print("\n## Array Reshape ##")
52 arr_resaped = arr.reshape(5, 1)
53 print("Reshaped array:\n", arr_resaped)
54
55 # Array Iterating
56 print("\n## Array Iterating ##")
57 print("Iterating through array:")
58 for element in arr:
59     print(element)
60
61 # Array Join
62 print("\n## Array Join ##")
63 arr1 = np.array([1, 2, 3])
64 arr2 = np.array([4, 5, 6])
65 arr_joined = np.concatenate((arr1, arr2))
66 print("Joined array:", arr_joined)
67
68 # Array Split
69 print("\n## Array Split ##")
70 arr_split = np.array_split(arr_joined, 3)
71 print("Split array:", arr_split)
72
73 # Array Search
74 print("\n## Array Search ##")
75 print("Index of value 3:", np.where(arr == 3))
76
77 # Array Sort
78 print("\n## Array Sort ##")
79 arr_unsorted = np.array([3, 1, 2])
80 arr_sorted = np.sort(arr_unsorted)
81 print("Sorted array:", arr_sorted)
82
83 # Array Filter
84 print("\n## Array Filter ##")
85 filtered_arr = arr[arr > 2]
86 print("Filtered array (elements greater than 2):", filtered_arr)
87
88 # Random Section
89 print("\n### Random Section ###")
90
91 # Random Intro
92 print("\n## Random Intro ##")
93 random_arr = np.random.randint(0, 10, 5)
94 print("Random array:", random_arr)
95
96 # Data Distribution
97 print("\n## Data Distribution ##")
98 uniform_dist = np.random.uniform(0, 1, 1000)
99 plt.hist(uniform_dist, bins=30)
100 plt.title("Uniform Distribution")
101 plt.show()
102
103 # Random Permutation
104 print("\n## Random Permutation ##")
```

```
105 permuted_arr = np.random.permutation(arr)
106 print("Permuted array:", permuted_arr)
107
108 # Seaborn Module
109 print("\n## Seaborn Module ##")
110 sns.distplot(uniform_dist, hist=False)
111 plt.title("Seaborn Distribution Plot")
112 plt.show()
113
114 # Normal Dist.
115 print("\n## Normal Dist. ##")
116 normal_dist = np.random.normal(0, 1, 1000)
117 plt.hist(normal_dist, bins=30)
118 plt.title("Normal Distribution")
119 plt.show()
120
121 # Binomial Dist.
122 print("\n## Binomial Dist. ##")
123 binomial_dist = np.random.binomial(10, 0.5, 1000)
124 plt.hist(binomial_dist, bins=30)
125 plt.title("Binomial Distribution")
126 plt.show()
127
128 # Poisson Dist.
129 print("\n## Poisson Dist. ##")
130 poisson_dist = np.random.poisson(5, 1000)
131 plt.hist(poisson_dist, bins=30)
132 plt.title("Poisson Distribution")
133 plt.show()
134
135 # Uniform Dist.
136 print("\n## Uniform Dist. ##")
137 uniform_dist = np.random.uniform(0, 1, 1000)
138 plt.hist(uniform_dist, bins=30)
139 plt.title("Uniform Distribution")
140 plt.show()
141
142 # Logistic Dist.
143 print("\n## Logistic Dist. ##")
144 logistic_dist = np.random.logistic(0, 1, 1000)
145 plt.hist(logistic_dist, bins=30)
146 plt.title("Logistic Distribution")
147 plt.show()
148
149 # Multinomial Dist.
150 print("\n## Multinomial Dist. ##")
151 multinomial_dist = np.random.multinomial(20, [1/6.]*6, size=1)
152 print("Multinomial distribution:", multinomial_dist)
153
154 # Exponential Dist.
155 print("\n## Exponential Dist. ##")
156 exponential_dist = np.random.exponential(1, 1000)
157 plt.hist(exponential_dist, bins=30)
158 plt.title("Exponential Distribution")
```

```
159 plt.show()
160
161 # Chi Square Dist.
162 print("\n## Chi Square Dist. ##")
163 chi_square_dist = np.random.chisquare(2, 1000)
164 plt.hist(chi_square_dist, bins=30)
165 plt.title("Chi Square Distribution")
166 plt.show()
167
168 # Rayleigh Dist.
169 print("\n## Rayleigh Dist. ##")
170 rayleigh_dist = np.random.rayleigh(1, 1000)
171 plt.hist(rayleigh_dist, bins=30)
172 plt.title("Rayleigh Distribution")
173 plt.show()
174
175 # Pareto Dist.
176 print("\n## Pareto Dist. ##")
177 pareto_dist = np.random.pareto(3, 1000)
178 plt.hist(pareto_dist, bins=30)
179 plt.title("Pareto Distribution")
180 plt.show()
181
182 # Zipf Dist.
183 print("\n## Zipf Dist. ##")
184 zipf_dist = np.random.zipf(2, 1000)
185 plt.hist(zipf_dist, bins=30)
186 plt.title("Zipf Distribution")
187 plt.show()
188
189 # ufunc Section
190 print("\n### ufunc Section ###")
191
192 # ufunc Intro
193 print("\n## ufunc Intro ##")
194 # Universal functions (ufuncs) are functions that operate on ndarrays in an element-by-
    element fashion.
195
196 # Create Function
197 print("\n## Create Function ##")
198 def my_ufunc(x):
199     return x + 1
200
201 my_ufunc = np.frompyfunc(my_ufunc, 1, 1)
202 print("Custom ufunc output:", my_ufunc(arr))
203
204 # Simple Arithmetic
205 print("\n## Simple Arithmetic ##")
206 print("Addition:", np.add(arr, arr))
207 print("Subtraction:", np.subtract(arr, arr))
208 print("Multiplication:", np.multiply(arr, arr))
209 print("Division:", np.divide(arr, arr))
210
211 # Rounding Decimals
```

```
212 print("\n## Rounding Decimals ##")
213 arr_float = np.array([1.234, 2.567, 3.891])
214 print("Rounded array:", np.around(arr_float, 2))
215
216 # Logs
217 print("\n## Logs ##")
218 print("Natural log:", np.log(arr))
219 print("Base 10 log:", np.log10(arr))
220
221 # Summations
222 print("\n## Summations ##")
223 print("Sum of array:", np.sum(arr))
224
225 # Products
226 print("\n## Products ##")
227 print("Product of array:", np.prod(arr))
228
229 # Differences
230 print("\n## Differences ##")
231 print("Differences in array:", np.diff(arr))
232
233 # Finding LCM
234 print("\n## Finding LCM ##")
235 print("LCM of 4 and 6:", np.lcm(4, 6))
236
237 # Finding GCD
238 print("\n## Finding GCD ##")
239 print("GCD of 8 and 12:", np.gcd(8, 12))
240
241 # Trigonometric
242 print("\n## Trigonometric ##")
243 print("Sine of array:", np.sin(arr))
244 print("Cosine of array:", np.cos(arr))
245
246 # Hyperbolic
247 print("\n## Hyperbolic ##")
248 print("Hyperbolic sine of array:", np.sinh(arr))
249 print("Hyperbolic cosine of array:", np.cosh(arr))
250
251 # Set Operations
252 print("\n## Set Operations ##")
253 arr_set1 = np.array([1, 2, 3, 4])
254 arr_set2 = np.array([3, 4, 5, 6])
255 print("Union:", np.union1d(arr_set1, arr_set2))
256 print("Intersection:", np.intersect1d(arr_set1, arr_set2))
257 print("Difference:", np.setdiff1d(arr_set1, arr_set2))
```