numpypart1

September 12, 2024

[1]:

```python
import numpy as np

# Creating a NumPy array from a Python list
python_list = [10, 20, 30, 40, 50]
numpy_array = np.array(python_list)
print("NumPy Array:", numpy_array)
```

```
NumPy Array: [10 20 30 40 50]
```

[2]:

```python
import numpy as np

# Creating a 2D array (matrix)
matrix = np.array([[7, 8, 9], [10, 11, 12]])
print("\n2D Array (Matrix):\n", matrix)
```

```

2D Array (Matrix):
 [[ 7  8  9]
 [10 11 12]]
```

[3]:

```python
import numpy as np

# Creating an array
array = np.array([6, 7, 8, 9, 10])

# Add 3 to each element
array_sum = array + 3

# Square each element
array_square = array ** 2

print("\nArray after adding 3:", array_sum)
print("Array after squaring:", array_square)
```

```

Array after adding 3: [ 9 10 11 12 13]
Array after squaring: [ 36  49  64  81 100]
```

[4]:

```python
import numpy as np

# Creating two arrays
array1 = np.array([1, 3, 5, 7, 9])
array2 = np.array([2, 4, 6, 8, 10])

# Element-wise addition
elementwise_sum = array1 + array2
print("\nElement-wise addition:", elementwise_sum)
```

```

Element-wise addition: [ 3  7 11 15 19]
```

[2]:

```python
import numpy as np

# Creating an array
array = np.array([15, 25, 35, 45, 55])

# Calculating mean, max, and min
mean_value = np.mean(array)
max_value = np.max(array)
min_value = np.min(array)

print("\nMean of array:", mean_value)
print("Max of array:", max_value)
print("Min of array:", min_value)
```

```

Mean of array: 35.0
Max of array: 55
Min of array: 15
```

[2]:

```python
import numpy as np

# Creating an array of zeros
zeros_array = np.zeros(4)

# Creating an array of ones
ones_array = np.ones(10)

print("\nArray of zeros:", zeros_array)
print("Array of ones:", ones_array)
```

```

Array of zeros: [0. 0. 0. 0.]
Array of ones: [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

[7]:

```python
import numpy as np

# Creating an array
array = np.array([2, 4, 6, 8,88])

# Reshaping the array
reshaped_array = np.reshape(array, (5, 1))

print("\nReshaped Array (5x1):\n", reshaped_array)
```

```

Reshaped Array (5x1):
 [[ 2]
 [ 4]
 [ 6]
 [ 8]
 [88]]
```

[8]:

```python
import numpy as np

# Creating an array with a range of numbers
range_array = np.arange(5, 15, 2)
print("\nArray with range from 5 to 15 with step 2:", range_array)
```

```

Array with range from 5 to 15 with step 2: [ 5  7  9 11 13]
```

[9]:

```python
import numpy as np

# Creating an identity matrix of size 3x3
identity_matrix = np.identity(3)
print("\nIdentity Matrix (3x3):\n", identity_matrix)
```

```

Identity Matrix (3x3):
 [[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

[10]:

```python
import numpy as np

# Creating a 2D array (matrix)
matrix = np.array([[1, 2, 3], [4, 5, 6]])

# Transposing the matrix
transposed_matrix = np.transpose(matrix)
print("\nOriginal Matrix:\n", matrix)
print("\nTransposed Matrix:\n", transposed_matrix)
```

```
Original Matrix:
 [[1 2 3]
 [4 5 6]]

Transposed Matrix:
 [[1 4]
 [2 5]
 [3 6]]
```

[1]:

```python
import numpy as np

# Create a 3x3x3 array
array_3d = np.array([[[1, 2, 3],
                      [4, 5, 6],
                      [7, 8, 9]],

                     [[10, 11, 12],
                      [13, 14, 15],
                      [16, 17, 18]],

                     [[19, 20, 21],
                      [22, 23, 24],
                      [25, 26, 27]]])

print("3D Array:")
print(array_3d)
```

```
3D Array:
[[[ 1  2  3]
  [ 4  5  6]
  [ 7  8  9]]

 [[10 11 12]
  [13 14 15]
  [16 17 18]]

 [[19 20 21]
  [22 23 24]
  [25 26 27]]]
```

[3]:

```python
import numpy as np

# Create a 3x3x3 array
array_3d = np.array([[[1, 2, 3],
                      [4, 5, 6],
                      [7, 8, 9]],
                     [[10, 11, 12],
                      [13, 14, 15],
                      [16, 17, 18]],

                     [[19, 20, 21],
                      [22, 23, 24],
                      [25, 26, 27]]])

print("3D Array:")
print(array_3d)

# Iterating over each depth slice (2D arrays)
print("\nIterating over each depth slice:")
for depth_slice in array_3d:
    print(depth_slice)
```

```
3D Array:
[[[ 1  2  3]
  [ 4  5  6]
  [ 7  8  9]]

 [[10 11 12]
  [13 14 15]
  [16 17 18]]

 [[19 20 21]
  [22 23 24]
  [25 26 27]]]

Iterating over each depth slice:
[[1 2 3]
 [4 5 6]
 [7 8 9]]
[[10 11 12]
 [13 14 15]
 [16 17 18]]
[[19 20 21]
 [22 23 24]
 [25 26 27]]
```

[9]:

```
# String Functions in Python

# 1. len()
text = "Hello, World!"
print("1. len():", len(text))  # Output: 13
```

```python
# 2. str.lower()
print("2. lower():", text.lower())  # Output: "hello, world!"

# 3. str.upper()
print("3. upper():", text.upper())  # Output: "HELLO, WORLD!"

# 4. str.capitalize()
print("4. capitalize():", text.capitalize())  # Output: "Hello, world!"

# 5. str.title()
print("5. title():", text.title())  # Output: "Hello, World!"

# 6. str.strip()
text_with_spaces = "   Hello, World!   "
print("6. strip():", text_with_spaces.strip())  # Output: "Hello, World!"

# 7. str.lstrip()
print("7. lstrip():", text_with_spaces.lstrip())  # Output: "Hello, World!   "

# 8. str.rstrip()
print("8. rstrip():", text_with_spaces.rstrip())  # Output: "   Hello, World!"

# 9. str.split()
print("9. split():", text.split())  # Output: ['Hello,', 'World!']

# 10. str.join()
words = ['Hello', 'World']
print("10. join():", " ".join(words))  # Output: "Hello World"

# 11. str.replace()
print("11. replace():", text.replace("World", "Python"))  # Output: "Hello, Python!"

# 12. str.find()
print("12. find():", text.find("World"))  # Output: 7

# 13. str.startswith()
print("13. startswith():", text.startswith("Hello"))  # Output: True

# 14. str.endswith()
print("14. endswith():", text.endswith("!"))  # Output: True

# 15. str.count()
print("15. count():", text.count("l"))  # Output: 3

# 16. str.isdigit()
numeric_string = "12345"
```

```
print("16. isdigit():", numeric_string.isdigit())  # Output: True

# 17. str.isalpha()
alpha_string = "HelloWorld"
print("17. isalpha():", alpha_string.isalpha())  # Output: True

# 18. str.isalnum()
alnum_string = "Hello123"
print("18. isalnum():", alnum_string.isalnum())  # Output: True

# 19. str.swapcase()
print("19. swapcase():", text.swapcase())  # Output: "hELLO, wORLD!"

# 20. str.zfill()
number_string = "42"
print("20. zfill():", number_string.zfill(6))  # Output: "00042"
'''
len(): Returns the number of characters in the string.
lower(): Converts all characters to lowercase.
upper(): Converts all characters to uppercase.
capitalize(): Capitalizes the first letter of the string.
title(): Capitalizes the first letter of each word.
strip(): Removes leading and trailing whitespace.
lstrip(): Removes leading whitespace.
rstrip(): Removes trailing whitespace.
split(): Splits the string into a list of words.
join(): Joins a list of strings into a single string with a specified separator.
replace(): Replaces occurrences of a substring with another substring.
find(): Returns the index of the first occurrence of a substring.
startswith(): Checks if the string starts with a specified substring.
endswith(): Checks if the string ends with a specified substring.
count(): Counts the occurrences of a substring in the string.
isdigit(): Checks if the string contains only digits.
isalpha(): Checks if the string contains only alphabetic characters.
isalnum(): Checks if the string contains only alphanumeric characters.
swapcase(): Swaps the case of all characters in the string.
zfill(): Pads the string on the left with zeros to fill the specified width.
'''
```

```
1. len(): 13
2. lower(): hello, world!
3. upper(): HELLO, WORLD!
4. capitalize(): Hello, world!
5. title(): Hello, World!
6. strip(): Hello, World!
7. lstrip(): Hello, World!
8. rstrip():    Hello, World!
```

```
9. split(): ['Hello,', 'World!']
10. join(): Hello World
11. replace(): Hello, Python!
12. find(): 7
13. startswith(): True
14. endswith(): True
15. count(): 3
16. isdigit(): True
17. isalpha(): True
18. isalnum(): True
19. swapcase(): hELLO, wORLD!
20. zfill(): 444442
```

[9]:

```
'\nlen(): Returns the number of characters in the string.\nlower(): Converts all
characters to lowercase.\nupper(): Converts all characters to
uppercase.\ncapitalize(): Capitalizes the first letter of the string.\ntitle():
Capitalizes the first letter of each word.\nstrip(): Removes leading and
trailing whitespace.\nlstrip(): Removes leading whitespace.\nrstrip(): Removes
trailing whitespace.\nsplit(): Splits the string into a list of words.\njoin():
Joins a list of strings into a single string with a specified
separator.\nreplace(): Replaces occurrences of a substring with another
substring.\nfind(): Returns the index of the first occurrence of a
substring.\nstartswith(): Checks if the string starts with a specified
substring.\nendswith(): Checks if the string ends with a specified
substring.\ncount(): Counts the occurrences of a substring in the
string.\nisdigit(): Checks if the string contains only digits.\nisalpha():
Checks if the string contains only alphabetic characters.\nisalnum(): Checks if
the string contains only alphanumeric characters.\nswapcase(): Swaps the case of
all characters in the string.\nzfill(): Pads the string on the left with zeros
to fill the specified width.\n'
```

[ ]: