NumPy Arithmetic Array Operations

NumPy provides a wide range of operations that can perform on arrays, including arithmetic operations.

NumPy's arithmetic operations are widely used due to their ability to perform simple and efficient calculations on arrays.

In this tutorial, we will explore some commonly used arithmetic operations in NumPy and learn how to use them to manipulate arrays.

List of Arithmetic Operations

Here's a list of various arithmetic operations along with their associated operators and built-in functions:

Element-wise Operation	Operator	Function
Addition	+	add()
Subtraction	-	subtract()
Multiplication	*	multiply()
Division	/	divide()
Exponentiation	**	power()
Modulus	%	mod()

To perform each operation, we can either use the associated operator or built-in functions. For example, **to perform addition**, we can either use the + operator or the add() built-in function.

Next, we will see examples of each of these operations.

NumPy Array Element-Wise Addition

As mentioned earlier, we can use the both + operator and the built-in function add() to perform element-wise addition between two NumPy arrays. For example,

```
import numpy as np

first_array = np.array([1, 3, 5, 7])
second_array = np.array([2, 4, 6, 8])

# using the + operator
result1 = first_array + second_array
print("Using the + operator:",result1)

# using the add() function
result2 = np.add(first_array, second_array)
print("Using the add() function:",result2)
```

Output

```
Using the + operator: [ 3  7 11 15]
Using the add() function: [ 3  7 11 15]
```

In the above example, first we created two arrays named: first_array and second_array.

Then, we used the + operator and the add() function to perform element-wise addition respectively.

As we can see, both + and add() give the same result.

NumPy Array Element-Wise Subtraction

In NumPy, we can either use the - operator or the subtract() function to perform element-wise subtraction between two NumPy arrays. For example,

```
import numpy as np

first_array = np.array([3, 9, 27, 81])
second_array = np.array([2, 4, 8, 16])

# using the - operator
result1 = first_array - second_array
print("Using the - operator:",result1)

# using the subtract() function
result2 = np.subtract(first_array, second_array)
print("Using the subtract() function:",result2)
```

Output

```
Using the - operator: [ 1  5 19 65]
Using the subtract() function: [ 1  5 19 65]
```

Here, we have performed subtraction between first_array and second_array using both the - operator and the subtract() function.

NumPy Array Element-Wise Multiplication

For element-wise multiplication, we can use the * operator or the multiply() function. For example,

```
import numpy as np

first_array = np.array([1, 3, 5, 7])
second_array = np.array([2, 4, 6, 8])

# using the * operator
result1 = first_array * second_array
print("Using the * operator:",result1)

# using the multiply() function
result2 = np.multiply(first_array, second_array)
print("Using the multiply() function:",result2)
```

Output

```
Using the * operator: [ 2 12 30 56]
Using the multiply() function: [ 2 12 30 56]
```

Here, we have used `*` and `multiply()` to demonstrate two ways of multiplying the two arrays element-wise.

NumPy Array Element-Wise Division

We can use either the `/` operator or the `divide()` function to perform element-wise division between two numpy arrays. For example,

```
import numpy as np

first_array = np.array([1, 2, 3])
second_array = np.array([4, 5, 6])

# using the / operator
result1 = first_array / second_array
print("Using the / operator:",result1)

# using the divide() function
result2 = np.divide(first_array, second_array)
print("Using the divide() function:",result2)
```

Output

```
Using the / operator: [0.25 0.4  0.5 ]
Using the divide() function: [0.25 0.4  0.5 ]
```

Here, we can see both `/` and `divide()` give the same result.

NumPy Array Element-Wise Exponentiation

Array exponentiation refers to raising each element of an array to a given power.

In NumPy, we can use either the `**` operator or the `power()` function to perform the element-wise exponentiation operation. For example,

```
import numpy as np

array1 = np.array([1, 2, 3])

# using the ** operator
result1 = array1 ** 2
print("Using the ** operator:",result1)

# using the power() function
result2 = np.power(array1, 2)
print("Using the power() function:",result2)
```

Output

```
Using the ** operator: [1 4 9]
Using the power() function: [1 4 9]
```

In the above example, we have used the `**` operator and the `power()` function to perform exponentiation operation respectively.

Here, `**` and `power()` both raise each element of `array1` to the power of **2**.

NumPy Array Element-Wise Modulus

We can perform a modulus operation in NumPy arrays using the `%` operator or the `mod()` function. This operation calculates the remainder of element-wise division between two arrays.

Let's see an example.

```
import numpy as np

first_array = np.array([9, 10, 20])
second_array = np.array([2, 5, 7])

# using the % operator
result1 = first_array % second_array
print("Using the % operator:",result1)

# using the mod() function
result2 = np.mod(first_array, second_array)
print("Using the mod() function:",result2)
```

Output

```
Using the % operator: [1 0 6]
Using the mod() function: [1 0 6]
```

Here, we have performed the element-wise modulus operation using both the `%` operator and the `mod()` function.