Array Operations

In this lesson, we will learn about scalar and element-wise array operations.

WE'LL COVER THE FOLLOWING

As we have seen in previous lessons, Python vectors and matrices are not equivalent to mathematical matrices, but they are similar in some ways.

> 💡 Keep in mind that certain Python vector and matrix operations differ from the mathematical vector and matrices.

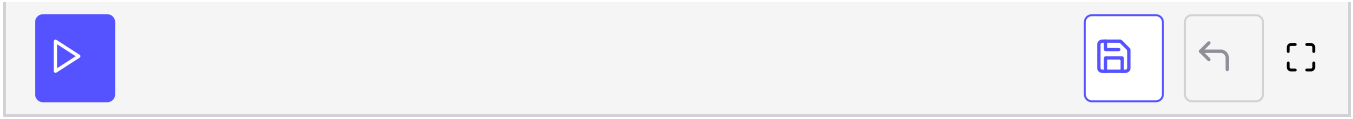
Scalar operations

We can use the usual arithmetic operators to multiply, add, subtract, and divide arrays with scalar numbers. These operations will be performed individually on each element.

Let's see the implementation for vectors below:

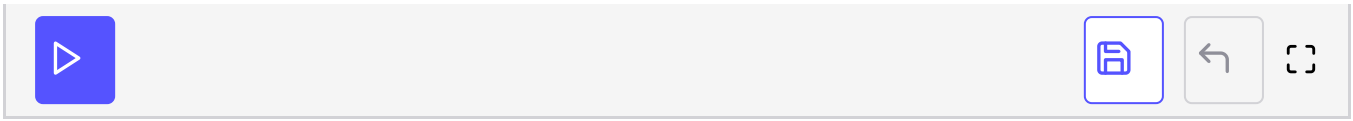
```
import numpy as np

v = np.arange(1, 11)
print(v)

print("-----")
print("Using scalar addition")
print(v + 2) # adding 2 to each element in the vector

print("-----")
print("Using scalar subtraction")
print(v - 2) # subtracting 2 from each element in the vector

print("-----")
print("Using scalar multiplication")
```

```
print(v * 2) # multiplying each element by 2 in the vector

print("-----")
print("Using scalar division")
print(v / 2) # dividing each element by 2 in the vector
```



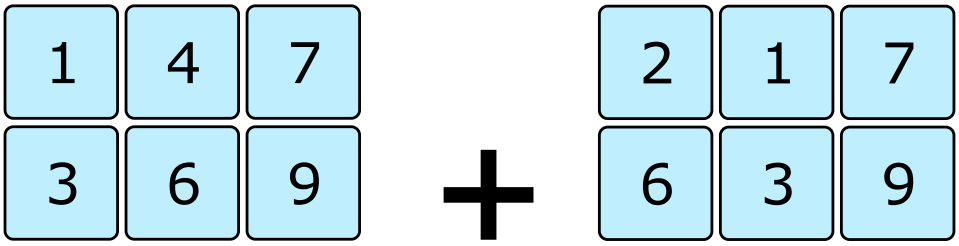
The same rules apply for matrices as well. See the example below:

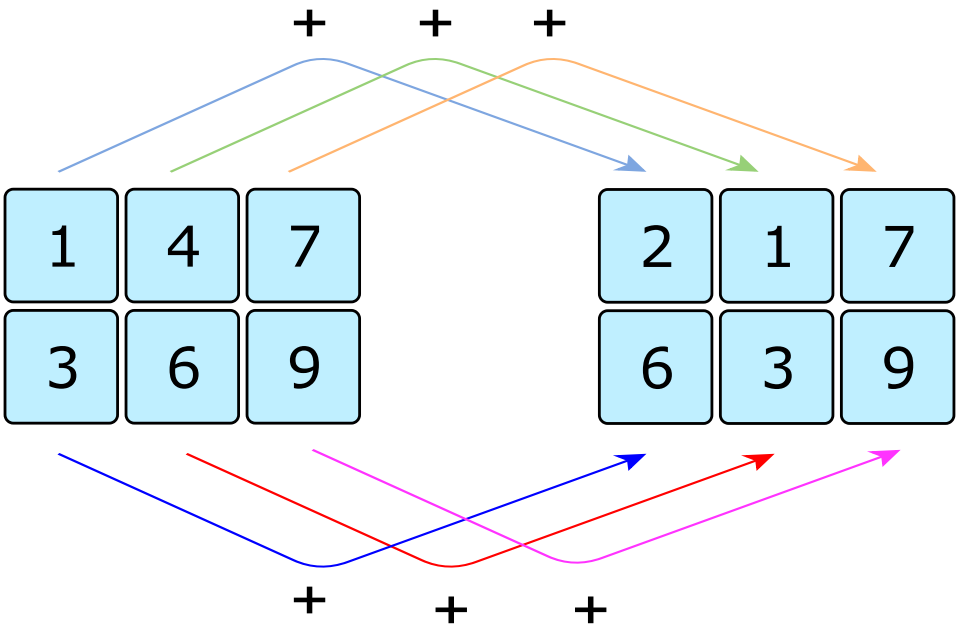
```
import numpy as np

M = np.array([[(n + m * 20) for n in range(5)] for m in range(5)])
print(M)

print("-----")
print("Using scalar addition")
print(M + 2) # adding 2 to each element in the matrix

print("-----")
print("Using scalar subtraction")
print(M - 2) # subtracting 2 from each element in the matrix

print("-----")
print("Using scalar multiplication")
print(M * 2) # multiplying each element by 2 in the matrix

print("-----")
print("Using scalar division")
print(M / 2) # dividing each element by 2 in the matrix
```

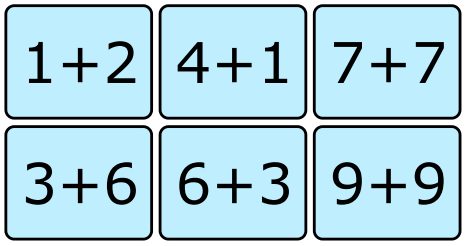


Element-wise operation

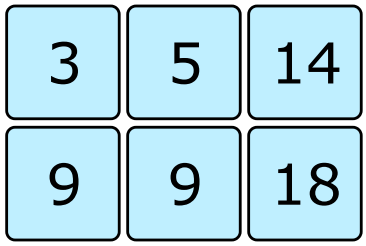
When we add, subtract, divide or multiply arrays, values are operated element-wise. Each element is operated against its corresponding index.


1 4 7
3 6 9
+
2 1 7
6 3 9


+ + +
1 4 7
3 6 9
2 1 7
6 3 9
+ + +



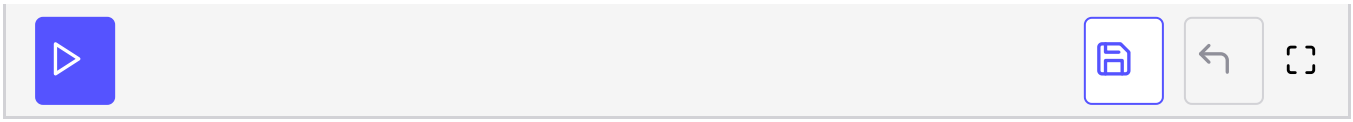
3 of 4



4 of 4

Let’s see an example of element-wise operations for NumPy arrays:

```
import numpy as np

M = np.array([[(n + m * 5) for n in range(3)] for m in range(3)])
print("M")
print(M)

N = np.array([[(n * 5) for n in range(3)] for m in range(3)])
print("N")
print(N)

print("-----")
print("M * M")
print(M * M) # multiplication

print("-----")
print("M + M") # addition
print(M + M)

print("-----")
print("M - N") # subtraction
print(M - N)

print("-----")
print("(M + 1) / (N + 1)")
print((M + 1) / (N + 2)) # division
                         # to avoid division by zero, we have added scalar 1
```



> 💡 Arrays must have **compatible shapes** if they were to be operated with each other.

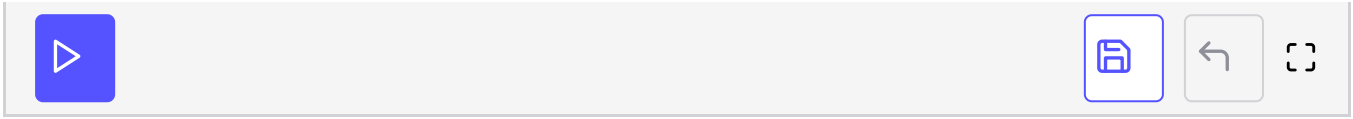
As a rule of thumb:

1. If there is an operation between two matrices, they should have exactly the same shape.
2. If there is an operation between a vector and a matrix, then the number of rows or the number of columns should be the same.

Let's take a look at an operation between a matrix and a single row vector:

```
import numpy as np

M = np.array([[(n + m * 5) for n in range(4)] for m in range(5)])
print("M")
print(M)        # 5 rows and 4 columns

N = np.array([1, 2, 3, 4])
print("N")
print(N)        # 1 row and 4 columns
```

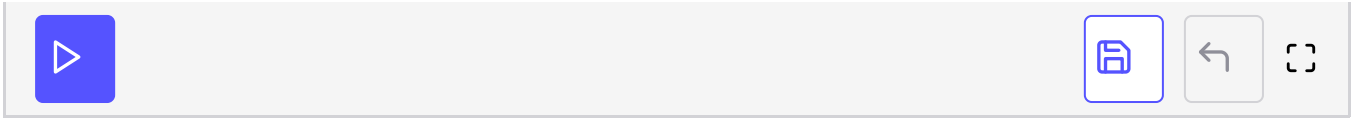
```
print(N)        # 1 row and 4 columns

print("-----")

print("N * M")
print(N * M)
```



The row vector is multiplied element-wise by **each row** of the matrix.

Let's see an operation between a matrix and a single column vector:

```
import numpy as np

M = np.array([[(n + m * 5) for n in range(4)] for m in range(5)])
print("M")
print(M)

N = np.array([[1], [2], [3], [4], [5]])
print("N")
print(N)

print("-----")
print("N * M")
print(N * M)
```



The column vector is multiplied element-wise by **each column** of the matrix.

In the next lesson, we will learn some data processing tools for arrays.