NumPy Operations

Arithmetic

You can easily perform array with array arithmetic, or scalar with array arithmetic. Let's see some examples:

```
In [2]: import numpy as np
        arr = np.arange(0,10)
```

```
In [3]: arr + arr
```

```
Out[3]: array([ 0,  2,  4,  6,  8, 10, 12, 14, 16, 18])
```

```
In [4]: arr * arr
```

```
Out[4]: array([ 0,  1,  4,  9, 16, 25, 36, 49, 64, 81])
```

```
In [5]: arr - arr
```

```
Out[5]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

```
In [6]: # Warning on division by zero, but not an error!
        # Just replaced with nan
        arr/arr
```

```
C:\Users\admin\AppData\Local\Temp\ipykernel_1964\2878212635.py:3: RuntimeWarning: inv
alid value encountered in true_divide
  arr/arr
```

```
Out[6]: array([nan,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.])
```

```
In [7]: # Also warning, but not an error instead infinity
        1/arr
```

```
C:\Users\admin\AppData\Local\Temp\ipykernel_1964\1360216608.py:2: RuntimeWarning: div
ide by zero encountered in true_divide
  1/arr
```

```
Out[7]: array([       inf, 1.        , 0.5       , 0.33333333, 0.25      ,
               0.2       , 0.16666667, 0.14285714, 0.125     , 0.11111111])
```

```
In [8]: arr**3
```

```
Out[8]: array([  0,   1,   8,  27,  64, 125, 216, 343, 512, 729], dtype=int32)
```

Universal Array Functions

Numpy comes with many Universal Array Functions , which are essentially just mathematical operations you can use to perform the operation across the array. Let's show some common ones:

In [9]:

```
#Taking Square Roots
np.sqrt(arr)
```

Out[9]:

```
array([0.        , 1.        , 1.41421356, 1.73205081, 2.        ,
       2.23606798, 2.44948974, 2.64575131, 2.82842712, 3.        ])
```

In [10]:

```
#Calcualting exponential (e^)
np.exp(arr)
```

Out[10]:

```
array([1.00000000e+00, 2.71828183e+00, 7.38905610e+00, 2.00855369e+01,
       5.45981500e+01, 1.48413159e+02, 4.03428793e+02, 1.09663316e+03,
       2.98095799e+03, 8.10308393e+03])
```

In [11]:

```
np.max(arr) #same as arr.max()
```

Out[11]:

```
9
```

In [12]:

```
np.sin(arr)
```

Out[12]:

```
array([ 0.        ,  0.84147098,  0.90929743,  0.14112001, -0.7568025 ,
       -0.95892427, -0.2794155 ,  0.6569866 ,  0.98935825,  0.41211849])
```

In [13]:

```
np.log(arr)
```

```
C:\Users\admin\AppData\Local\Temp\ipykernel_1964\3120950136.py:1: RuntimeWarning: div
ide by zero encountered in log
  np.log(arr)
```

Out[13]:

```
array([      -inf, 0.        , 0.69314718, 1.09861229, 1.38629436,
       1.60943791, 1.79175947, 1.94591015, 2.07944154, 2.19722458])
```

In [18]:

```
arrTranspose = arr.T
arrTranspose
```

Out[18]:

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [22]:

```
arr2 = [[1,0],[0,1]]
np.linalg.inv(arr2) # Inverse of a matrix
```

Out[22]:

```
array([[1., 0.],
       [0., 1.]])
```