NumPy Operations

Arithmetic

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

In [1]:

```
import numpy as np
arr = np.arange(0,10)
```

In [2]:

```
arr
```

Out[2]:

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

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 [7]:

```
# Warning on division by zero, but not an error!
# Just replaced with nan
arr/arr
```

```
/home/punit/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.
py:3: RuntimeWarning: invalid value encountered in true_divide
  This is separate from the ipykernel package so we can avoid doing i
mports until
```

Out[7]:

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

In [8]:

```
2/arr
```

```
/home/punit/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.
py:1: RuntimeWarning: divide by zero encountered in true_divide
  """Entry point for launching an IPython kernel.
```

Out[8]:

```
array([       inf, 2.        , 1.        , 0.66666667, 0.5       ,
       0.4       , 0.33333333, 0.28571429, 0.25      , 0.22222222])
```

In [9]:

```
# Also warning, but not an error instead infinity
1/arr
```

```
/home/punit/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.
py:2: RuntimeWarning: divide by zero encountered in true_divide
```

Out[9]:

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

In [11]:

```
arr
```

Out[11]:

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

In [14]:

```
3**3
```

Out[14]:

```
27
```

In [15]:

```
arr**3
```

Out[15]:

```
array([  0,   1,   8,  27,  64, 125, 216, 343, 512, 729])
```

In [16]:

```
arr
```

Out[16]:

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

Universal Array Functions

Numpy comes with many [universal array functions (http://docs.scipy.org/doc/numpy/reference/ufuncs.html)](http://docs.scipy.org/doc/numpy/reference/ufuncs.html), which are essentially just mathematical operations you can use to perform the operation across the array. Let's show some common ones:

In [17]:

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

Out[17]:

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

In [18]:

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

Out[18]:

```
array([1.00000000e+00, 2.71828183e+00, 7.38905610e+00, 2.00855369e+0
1,
       5.45981500e+01, 1.48413159e+02, 4.03428793e+02, 1.09663316e+0
3,
       2.98095799e+03, 8.10308393e+03])
```

In [20]:

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

Out[20]:

```
9
```

In [22]:

```
np.min(arr)
```

Out[22]:

```
0
```

In [24]:

```
np.cos(arr)
```

Out[24]:

```
array([ 1.        ,  0.54030231, -0.41614684, -0.9899925 , -0.6536436
2,
        0.28366219,  0.96017029,  0.75390225, -0.14550003, -0.9111302
6])
```

In [25]:

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

```
/home/punit/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.
py:1: RuntimeWarning: divide by zero encountered in log
  """Entry point for launching an IPython kernel.
```

Out[25]:

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

In [ ]:

```

```