Find Trignometry Sin(),Cos() & Tan() Using NumPy

Trignometry Functions & Random Sampling

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt
```

In [ ]:
```
np.sin(180)
```

Out[ ]:
```
-0.8011526357338304
```

In [ ]:
```
np.cos(180)
```

Out[ ]:
```
-0.5984600690578582
```

In [ ]:
```
np.tan(180)
```

Out[ ]:
```
1.3386902103511544
```

In [ ]:
```
x_sin = np.arange(0, 3*np.pi, 0.1)
```

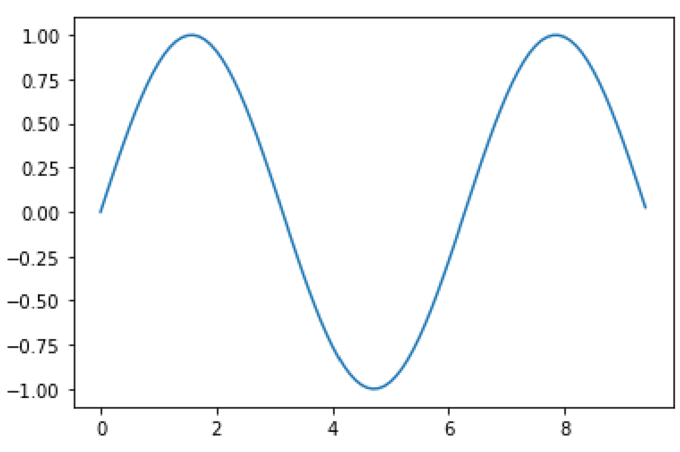
In [ ]:
```
x_sin
```

Out[ ]:
```
array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. , 1.1, 1.2,
       1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2. , 2.1, 2.2, 2.3, 2.4, 2.5,
       2.6, 2.7, 2.8, 2.9, 3. , 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8,
       3.9, 4. , 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 5. , 5.1,
       5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 6. , 6.1, 6.2, 6.3, 6.4,
       6.5, 6.6, 6.7, 6.8, 6.9, 7. , 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7,
       7.8, 7.9, 8. , 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, 8.8, 8.9, 9. ,
       9.1, 9.2, 9.3, 9.4])
```

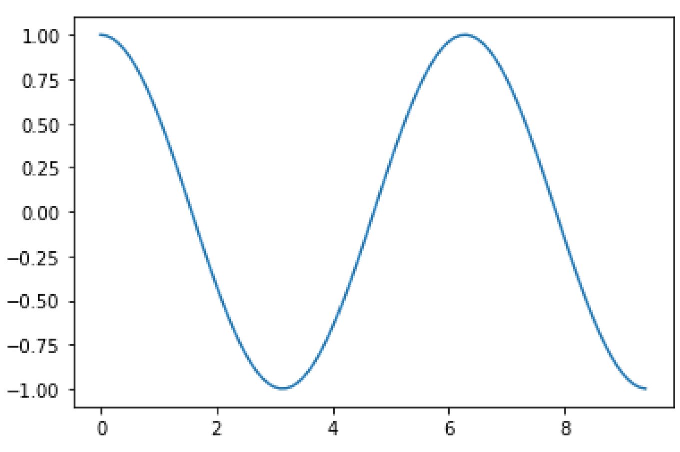
In [ ]:
```
y_sin = np.sin(x_sin)
y_sin
```

Out[ ]:
```
array([ 0.        ,  0.09983342,  0.19866933,  0.29552021,  0.38941834,
        0.47942554,  0.56464247,  0.64421769,  0.71735609,  0.78332691,
        0.84147098,  0.89120736,  0.93203909,  0.96355819,  0.98544973,
        0.99749499,  0.9995736 ,  0.99166481,  0.97384763,  0.94630009,
        0.90929743,  0.86320937,  0.8084964 ,  0.74570521,  0.67546318,
        0.59847214,  0.51550137,  0.42737988,  0.33498815,  0.23924933,
        0.14112001,  0.04158066, -0.05837414, -0.15774569, -0.2555411 ,
       -0.35078323, -0.44252044, -0.52983614, -0.61185789, -0.68776616,
       -0.7568025 , -0.81827711, -0.87157577, -0.91616594, -0.95160207,
       -0.97753012, -0.993691  , -0.99992326, -0.99616461, -0.98245261,
       -0.95892427, -0.92581468, -0.88345466, -0.83226744, -0.77276449,
       -0.70554033, -0.63126664, -0.55068554, -0.46460218, -0.37387666,
       -0.2794155 , -0.1821625 , -0.0830894 ,  0.0168139 ,  0.1165492 ,
        0.21511999,  0.31154136,  0.40484992,  0.49411335,  0.57843976,
        0.6569866 ,  0.72896904,  0.79366786,  0.85043662,  0.8987081 ,
        0.93799998,  0.96791967,  0.98816823,  0.99854335,  0.99894134,
        0.98935825,  0.96988981,  0.94073056,  0.90217183,  0.85459891,
        0.79848711,  0.7343971 ,  0.66296923,  0.58491719,  0.50102086,
        0.41211849,  0.31909836,  0.22288991,  0.12445442,  0.02477543])
```

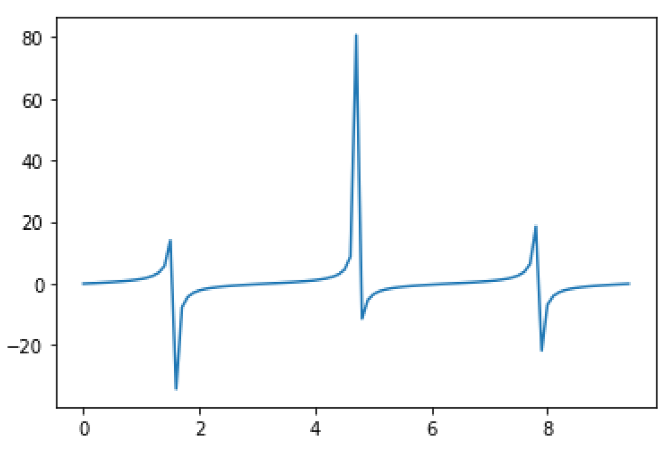
In [ ]:
```
plt.plot(x_sin, y_sin)
plt.show()
```



In [ ]:
```
y_cos = np.cos(x_sin)
plt.plot(x_sin, y_cos)
plt.show()
```



In [ ]:
```
y_tan = np.tan(x_sin)
plt.plot(x_sin, y_tan)
plt.show()
```



Random Sampling With Numpy

In [ ]:
```
import numpy as np
import random
```

In [ ]:
```
np.random.random(1)
```

Out[ ]:
```
array([0.8027798])
```

In [ ]:
```
np.random.random((3, 3))
```

Out[ ]:
```
array([[0.78412146, 0.26106899, 0.07138804],
       [0.54579565, 0.92090285, 0.58617832],
       [0.14585943, 0.44304812, 0.5926764 ]])
```

In [ ]:
```
np.random.randint(1, 4)
```

Out[ ]:
```
3
```

In [ ]:
```
## np.random. rand int (values, shape)
np.random.randint(1, 4, (4,4))
```

Out[ ]:
```
array([[1, 2, 2, 1],
       [3, 3, 2, 2],
       [1, 2, 2, 1],
       [3, 2, 2, 3]])
```

In [ ]:
```
np.random.randint(1, 4, (2,4,4))
```

Out[ ]:
```
array([[[2, 1, 1, 1],
        [1, 1, 3, 2],
        [3, 2, 3, 2],
        [1, 3, 2, 3]],

       [[3, 2, 2, 3],
        [2, 3, 3, 1],
        [3, 1, 3, 2],
        [1, 3, 2, 2]]])
```

In [ ]:
```
np.random.seed(10)
np.random.randint(1, 4, (2,4,4))
```

Out[ ]:
```
array([[[2, 2, 1, 1],
        [2, 1, 2, 2],
        [1, 2, 2, 3],
        [1, 2, 1, 3]],

       [[1, 3, 1, 1],
        [1, 3, 1, 3],
        [3, 2, 1, 1],
        [3, 2, 3, 2]]])
```

In [ ]:
```
np.random.rand(3)
```

Out[ ]:
```
array([0.13145815, 0.41366737, 0.77872881])
```

In [ ]:
```
np.random.rand(3,3)
```

Out[ ]:
```
array([[0.58390137, 0.18263144, 0.82608225],
       [0.10540183, 0.28357668, 0.06556327],
       [0.05644419, 0.76545582, 0.01178803]])
```

In [ ]:
```
np.random.randn(3,3)
```

Out[ ]:
```
array([[-1.58494101,  1.05535316, -1.92657911],
       [ 0.69858388, -0.74620143, -0.15662666],
       [-0.19363594,  1.13912535,  0.36221796]])
```

In [ ]:
```
x = [1,2,3,4]
np.random.choice(x)
```

Out[ ]:
```
1
```

In [ ]:
```
x = [1,2,3,4]
np.random.choice(x)
```

Out[ ]:
```
1
```

In [ ]:
```
for i in range(20):
   print(np.random.choice(x))
```

```
1
2
1
4
1
1
3
1
3
3
1
4
2
1
4
2
3
3
2
3
```

In [ ]:
```
x
```

Out[ ]:
```
[1, 2, 3, 4]
```

In [ ]:
```
np.random.permutation(x)
```

Out[ ]:
```
array([2, 4, 1, 3])
```

In [ ]:
```
## Scipy.org for further Radom Function
```