- Saving and loading notebooks in GitHub
- Interactive forms
- Interactive widgets

Working with data

- Loading data: Drive, Sheets and Google Cloud Storage
- Charts: visualising data
- Getting started with BigQuery

Machine learning crash course

These are a few of the notebooks from Google's online machine learning course. See the full course website for more.

- Intro to Pandas DataFrame
- Linear regression with tf.keras using synthetic data

Using accelerated hardware

- TensorFlow with GPUs
- TensorFlow with TPUs

Featured examples

- NeMo voice swap: Use Nvidia NeMo conversational AI toolkit to swap a voice in an audio fragment with a computer-generated one.
- Retraining an Image Classifier: Build a Keras model on top of a pre-trained image classifier to distinguish flowers.
- Text Classification: Classify IMDB film reviews as either *positive* or *negative*.
- Style Transfer: Use deep learning to transfer style between images.
- Multilingual Universal Sentence Encoder Q&A: Use a machine-learning model to answer questions from the SQuAD dataset.
- Video Interpolation: Predict what happened in a video between the first and the last frame.

```
import numpy as np
```

```
a=np.array([[1,2,3],[4,5,6]])
a
```

```
array([[1, 2, 3],
       [4, 5, 6]])
```

```
#Attributes of Numpy
```

```
a
```

```
array([[1, 2, 3],
       [4, 5, 6]])
```

```
#shape
```

```
a.shape
```

```
(2, 3)
```

```
#Size
a.size
```

```
6
```

```
#Reshape
```

```
a.reshape(3,2)
```

```
array([[1, 2],
       [3, 4],
       [5, 6]])
```

```
a
```

```
array([[1, 2, 3],
       [4, 5, 6]])
```

```
#shape for Parment change

a.shape=(3,2)
a
```

```
array([[1, 2],
       [3, 4],
       [5, 6]])
```

```
a
```

```
array([[1, 2],
       [3, 4],
       [5, 6]])
```

```
#Transpose

a.T
```

```
array([[1, 3, 5],
       [2, 4, 6]])
```

```
#ndim

a.ndim
```

```
2
```

```
#funcation

np.arange(1,7)
```

```
array([1, 2, 3, 4, 5, 6])
```

```
np.arange(1,7).reshape(2,3)
```

```
array([[1, 2, 3],
       [4, 5, 6]])
```

```
a=np.arange(1,7).reshape(3,2)
b=np.arange(7,13).reshape(3,2)
a
```

```
array([[1, 2],
       [3, 4],
       [5, 6]])
```

```
b
```

```
array([[ 7,  8],
       [ 9, 10],
       [11, 12]])
```

```
np.concatenate((a,b))
```

```
array([[ 1,  2],
       [ 3,  4],
       [ 5,  6],
       [ 7,  8],
       [ 9, 10],
       [11, 12]])
```

```
np.hstack((a,b))
```

```
array([[ 1,  2,  7,  8],
       [ 3,  4,  9, 10],
       [ 5,  6, 11, 12]])
```

```
np.vstack((a,b))
```

```
array([[ 1,  2],
       [ 3,  4],
       [ 5,  6],
       [ 7,  8],
       [ 9, 10],
       [11, 12]])
```

```
c=np.arange(1,7)
c
```

```
array([1, 2, 3, 4, 5, 6])
```

```
np.append(c,[4,5,6])
```

```
array([1, 2, 3, 4, 5, 6, 4, 5, 6])
```

```
np.append(c,[111])
```

```
array([  1,   2,   3,   4,   5,   6, 111])
```

```
c
```

```
array([1, 2, 3, 4, 5, 6])
```

```
d=np.arange(2,6)
d
```

```
array([2, 3, 4, 5])
```

```
np.insert(d,0,12)
```

```
array([12,  2,  3,  4,  5])
```

```
d
```

```
array([2, 3, 4, 5])
```

```
np.insert(d,[1,2],[15,25])
```

```
array([ 2, 15,  3, 25,  4,  5])
```

```
np.append(d,[25,48,8,6])
```

```
array([ 2,  3,  4,  5, 25, 48,  8,  6])
```

Start coding or generate with AI.

Start coding or generate with AI.

```
a=np.arange(1,6)
a
```

```
array([1, 2, 3, 4, 5])
```

```
np.delete(a,[0])
```

```
array([2, 3, 4, 5])
```

```
np.delete(a,[0,2])
```

```
array([2, 4, 5])
```

```
a
```

```
array([1, 2, 3, 4, 5])
```

```
#Array Creation Routine
```

```
#1. Empty
```

```
x=np.empty([3,2])
x
```

```
array([[7.10515805e-320, 0.00000000e+000],
       [2.89804015e+262, 4.13352893e+122],
       [2.04990036e-309, 3.50676843e+286]])
```

```
x=np.empty([])
x
```

```
array(3.50676843e+286)
```

```
#Zeros

np.zeros(5)
```

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

```
a=np.zeros(10)
a
```

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

```
#ones

np.ones(5)
```

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

```
a=np.ones(7)
a
```

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

```
#eye

s=np.eye(2)
s
```

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

```
s=np.eye(5)
s
```

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

```
#linespace

np.linspace(11,15,5)
```

```
array([11., 12., 13., 14., 15.])
```

```
a=np.linspace(11,20,15)
a
```

```
array([11.        , 11.64285714, 12.28571429, 12.92857143, 13.57142857,
       14.21428571, 14.85714286, 15.5       , 16.14285714, 16.78571429,
       17.42857143, 18.07142857, 18.71428571, 19.35714286, 20.        ])
```

```
#slicing & Indexing of an array

arr=np.arange(1,6)
arr
```

```
array([1, 2, 3, 4, 5])
```

```
arr[0]
```

```
1
```

```
arr[4]
```

```
5
```

Double-click (or enter) to edit

```
arr[0:2]
```

```
array([1, 2])
```

```
arr[1:4]
```

```
array([2, 3, 4])
```

```
#Indexing & slicing on 2 Dimesional array


arra_2d=np.array([[5,10,15],[20,25,30],[35,40,45]])
arra_2d
```

```
array([[ 5, 10, 15],
       [20, 25, 30],
       [35, 40, 45]])
```

```
arra_2d[0]
```

```
array([ 5, 10, 15])
```

```
arra_2d[0:2]
```

```
array([[ 5, 10, 15],
       [20, 25, 30]])
```

```
arra_2d[1:,:2]
```

```
array([[20, 25],
       [35, 40]])
```

```
arr[:10]=100
arr
```

```
array([100, 100, 100, 100, 100])
```

```
###braodcasting or view creation

arr=np.arange(0,11)
arr
```

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

```
arr[0:6]
```

```
array([0, 1, 2, 3, 4, 5])
```

```
aview=arr[0:6]
aview
```

```
array([0, 1, 2, 3, 4, 5])
```

```
acopy=arr.copy()
acopy
```

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

```
#Fancy Indexing
x=np.zeros((10,10))
x
```

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

```
x.shape[0]
```

```
10
```

```
length=x.shape[1]
length
```

```
10
```

```
for i in range(length):
  x[i]=i
x
```

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

```
#numpay -Array Selection

z=np.arange(1,11)
z
```

```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

```
a=list(range(1,11))
a
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
a*4
```

```
[1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
 10,
 1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
 10,
 1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
 10,
 1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
 10]
```

```
z*4
```

```
array([ 4,  8, 12, 16, 20, 24, 28, 32, 36, 40])
```

```
z*9
```

```
array([ 9, 18, 27, 36, 45, 54, 63, 72, 81, 90])
```

```
z
```

```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

```
z>5
```

```
array([False, False, False, False, False,  True,  True,  True,  True,
       True])
```

```
mask=z>5
mask
```

```
array([False, False, False, False, False,  True,  True,  True,  True,
       True])
```

```
z[mask]
```

```
array([ 6,  7,  8,  9, 10])
```

```
z[z>5] #masking / Selection/filtering
```

```
array([ 6,  7,  8,  9, 10])
```

```
#Filtering / Masking / Selection
```

```
names=np.array(["Pratik", "Faizan", "Alok", "Pratik","Ayush", "Pratik", "Faizan"])
names
```

```
array(['Pratik', 'Faizan', 'Alok', 'Pratik', 'Ayush', 'Pratik', 'Faizan'],
      dtype='<U6')
```

```
mask=names=="Pratik"
names[mask]
```

```
array(['Pratik', 'Pratik', 'Pratik'], dtype='<U6')
```

```
names[names=="Pratik"]
```

```
array(['Pratik', 'Pratik', 'Pratik'], dtype='<U6')
```

```
names[names!="Pratik"]
```

```
array(['Faizan', 'Alok', 'Ayush', 'Faizan'], dtype='<U6')
```

```
#OR

(names=="Pratik")|(names=="Pratik")
```

```
array([ True, False, False,  True, False,  True, False])
```

```
names[(names=="Pratik")|(names=="Faizan")]
```

```
array(['Pratik', 'Faizan', 'Pratik', 'Pratik', 'Faizan'], dtype='<U6')
```

```
#: #select all the values, which is not pratik and assign them as Purvesh
names[names!="Pratik"]="Purvesh"
names
```

```
array(['Pratik', 'Purves', 'Purves', 'Pratik', 'Purves', 'Pratik',
       'Purves'], dtype='<U6')
```

```
np.unique(names)
```

```
array(['Pratik', 'Purves'], dtype='<U6')
```

```
names
```

```
array(['Pratik', 'Purves', 'Purves', 'Pratik', 'Purves', 'Pratik',
       'Purves'], dtype='<U6')
```