

# 1806554 Ganesh Bhandarkar Assignment 5

## Numpy

```
In [35]: import numpy as np
```

```
In [36]: a = np.array([1,2,3,4,5])
print(a)
```

```
[1 2 3 4 5]
```

```
In [37]: a = np.zeros(5)
print(a)
```

```
[0. 0. 0. 0. 0.]
```

```
In [38]: a = np.ones(5)
print(a)
```

```
[1. 1. 1. 1. 1.]
```

```
In [39]: print(dir(np),end="")
```

```
['ALLOW_THREADS', 'AxisError', 'BUFSIZE', 'CLIP', 'ComplexWarning', 'DataSource', 'ERR_CALL', 'ERR_DEFAULT', 'ERR_IGNORE', 'ERR_LOG', 'ERR_PRINT', 'ERR_RAISE', 'ERR_WARN', 'FLOATING_POINT_SUPPORT', 'FPE_DIVIDEBYZERO', 'FPE_INVALID', 'FPE_OVERFLOW', 'FPE_UNDERFLOW', 'False_', 'Inf', 'Infinity', 'MAXDIMS', 'MAY_SHARE_BOUNDS', 'MAY_SHARE_EXACT', 'MachAr', 'ModuleDeprecationWarning', 'NAN', 'NINF', 'NZERO', 'NaN', 'PINF', 'PZERO', 'RAISE', 'RankWarning', 'SHIFT_DIVIDEBYZERO', 'SHIFT_INVALID', 'SHIFT_OVERFLOW', 'SHIFT_UNDERFLOW', 'ScalarType', 'Tester', 'TooHardError', 'True_', 'UFUNC_BUFSIZE_DEFAULT', 'UFUNC_PYVALS_NAME', 'VisibleDeprecationWarning', 'WRAP', '_NoValue', '_UFUNC_API', '__NUMPY_SETUP__', '__all__', '__builtins__', '__cached__', '__config__', '__dir__', '__doc__', '__file__', '__getattr__', '__git_revision__', '__loader__', '__mkl_version__', '__name__', '__package__', '__path__', '__spec__', '__version__', '_add_newdoc_ufunc', '_distributor_init', '_globals', '_mat', '_pytesttester', 'abs', 'absolute', 'add', 'add_docstring', 'add_newdoc', 'add_newdoc_ufu
```

nc', 'alen', 'all', 'allclose', 'alltrue', 'amax', 'amin', 'angle', 'any', 'append', 'apply\_along\_axis', 'apply\_over\_axes', 'arange', 'arccos', 'arccosh', 'arcsin', 'arcsinh', 'arctan', 'arctan2', 'arctanh', 'argmax', 'argmin', 'argpartition', 'argsort', 'argwhere', 'around', 'array', 'array2string', 'array\_equal', 'array\_equiv', 'array\_repr', 'array\_split', 'array\_str', 'asanyarray', 'asarray', 'asarray\_chkfinite', 'ascontiguousarray', 'asfarray', 'asfortranarray', 'asmatrix', 'asscalar', 'atleast\_1d', 'atleast\_2d', 'atleast\_3d', 'average', 'bartlett', 'base\_repr', 'binary\_repr', 'bincount', 'bitwise\_and', 'bitwise\_not', 'bitwise\_or', 'bitwise\_xor', 'blackman', 'block', 'bmat', 'bool', 'bool8', 'bool\_', 'broadcast', 'broadcast\_arrays', 'broadcast\_shape', 'busday\_count', 'busday\_offset', 'busdaycalendar', 'byte', 'byte\_bounds', 'bytes0', 'bytes\_', 'c\_', 'can\_cast', 'cast', 'cbart', 'cdouble', 'ceil', 'cfloat', 'char', 'character', 'chararray', 'choose', 'clip', 'clongdouble', 'clongfloat', 'column\_stack', 'common\_type', 'compare\_chararrays', 'compat', 'complex', 'complex128', 'complex64', 'complex\_', 'complexfloating', 'compress', 'concatenate', 'conj', 'conjugate', 'convolve', 'copy', 'copysign', 'copyto', 'core', 'corrcoef', 'correlate', 'cos', 'cosh', 'count\_nonzero', 'cov', 'cross', 'csingle', 'ctypeslib', 'cumprod', 'cumproduct', 'cumsum', 'datetime64', 'datetime\_as\_string', 'datetime\_data', 'deg2rad', 'degrees', 'delete', 'deprecate', 'deprecate\_with\_doc', 'diag', 'diag\_indices', 'diag\_indices\_from', 'diagflat', 'diagonal', 'diff', 'digitize', 'disp', 'divide', 'divmod', 'dot', 'double', 'dsplit', 'dstack', 'dtype', 'e', 'ediff1d', 'einsum', 'einsum\_path', 'emath', 'empty', 'empty\_like', 'equal', 'errstate', 'euler\_gamma', 'exp', 'exp2', 'expand\_dims', 'expm1', 'extract', 'eye', 'fabs', 'fastCopyAndTranspose', 'fft', 'fill\_diagonal', 'find\_common\_type', 'finfo', 'fix', 'flatiter', 'flatnonzero', 'flexible', 'flip', 'fliplr', 'flipud', 'float', 'float16', 'float32', 'float64', 'float\_', 'float\_power', 'floating', 'floor', 'floor\_divide', 'fmax', 'fmin', 'fmod', 'format\_float\_positional', 'format\_float\_scientific', 'format\_parser', 'frexp', 'frombuffer', 'fromfile', 'fromfunction', 'fromiter', 'frompyfunc', 'fromregex', 'fromstring', 'full', 'full\_like', 'fv', 'gcd', 'generic', 'genfromtxt', 'geomspace', 'get\_array\_wrap', 'get\_include', 'get\_printoptions', 'getbufsize', 'geterr', 'geterrcall', 'geterrobj', 'gradient', 'greater', 'greater\_equal', 'half', 'hamming', 'hanning', 'heaviside', 'histogram', 'histogram2d', 'histogram\_bin\_edges', 'histogramdd', 'hsplit', 'hstack', 'hypot', 'i0', 'identity', 'iinfo', 'imag', 'in1d', 'index\_exp', 'indices', 'inexact', 'inf', 'info', 'infty', 'inner', 'insert', 'int', 'int0', 'int16', 'int32', 'int64', 'int8', 'int\_', 'intc', 'integer', 'interp', 'intersect1d', 'intp', 'invert', 'ipmt', 'irr', 'is\_busday', 'isclose', 'iscomplex', 'iscomplexobj', 'isfinite', 'isfortran', 'isin', 'isinf', 'isnan', 'isnat', 'isneginf', 'isposinf', 'isreal', 'isrealobj', 'isscalar', 'issctype', 'issubclass\_', 'issubdtype', 'issubdtype', 'issubdtype', 'iterable', 'ix\_', 'kaiser', 'kron', 'lcm', 'ldexp', 'left\_shift', 'less', 'less\_equal', 'lexsort', 'lib', 'linalg', 'linspace', 'little\_endian', 'load', 'loads', 'loadtxt', 'log', 'log10', 'log1p', 'log2', 'logaddexp', 'logaddexp2', 'logical\_and', 'logical\_not', 'logical\_or', 'logical\_xor', 'logspace', 'long', 'longcomplex', 'longdouble', 'longfloat', 'longlong', 'lookfor', 'ma', 'mafromtxt', 'mask\_indices', 'mat', 'math', 'matmul', 'matrix', 'matrixlib', 'max', 'maximum', 'maximum\_sctype', 'may\_share\_memory', 'mean', 'median', 'memmap', 'meshgrid', 'mgrid', 'min', 'min\_scalar\_type', 'minimum', 'mintypecode', 'mirr', 'mkl', 'mod', 'modf', 'moveaxis', 'msort', 'multiply', 'nan', 'nan\_to\_num', 'nanargmax', 'nanargmin', 'nancumprod', 'nancumsum', 'nanmax', 'nanmean', 'nanmedian', 'nanmin', 'nanpercentile', 'nanprod', 'nanquantile', 'nanstd', 'nansum', 'nanvar', 'nbytes', 'ndarray', 'ndenumerate', 'ndfromtxt', 'ndim', 'ndindex', 'nditer', 'negative', 'nested\_iters', 'newaxis', 'nextafter', 'nonzero', 'not\_equal', 'nper', 'npv', 'numarray', 'number', 'obj2sctype', 'object', 'object0', 'object\_', 'ogrid', 'oldnumeric', 'ones', 'ones\_like', 'os', 'outer', 'packbits', 'pad', 'partition', 'percentile', 'pi', 'piecewise', 'place', 'pmt', 'poly', 'poly1d', 'polyadd', 'polyder', 'polydiv', 'polyfit', 'polyint', 'polymul', 'polynomial', 'polysub', 'polyval', 'positive', 'power', 'ppmt', 'printoptions', 'prod', 'product', 'promote\_types', 'ptp', 'put', 'put\_along\_axis', 'putmask', 'pv', 'quantile', 'r\_', 'rad2deg', 'radians', 'random', 'rate', 'ravel', 'ravel\_multi\_index', 'real', 'real\_if\_close', 'rec', 'recarray', 'recfromcsv', 'recfromtxt', 'reciprocal', 'record', 'remainder', 'repeat', 'require', 'reshape', 'resize', 'result\_type', 'right\_shift', 'rint', 'roll', 'rollaxis', 'roots', 'rot90', 'round', 'round\_', 'row\_stack', 's\_', 'safe\_eval', 'save', 'savetxt', 'savez', 'savez\_compressed', 'sctype2char', 'sctypeDict', 'sctypeNA', 'sctypes', 'searchsorted', 'select', 'set\_numeric\_ops', 'set\_printoptions', 'set\_string\_function', 'setbufsize', 'setdiff1d', 'seterr', 'seterrcall', 'seterrobj', 'setxor1d', 'shape', 'shares\_memory', 'short', 'show\_config', 's

```
ign', 'signbit', 'signedinteger', 'sin', 'sinc', 'single', 'singlecomplex', 'sinh', 'size', 'sometrue', 'sort', 'sort_complex', 'source', 'spacing', 'split', 'sqrt', 'square', 'squeeze', 'stack', 'std', 'str', 'str0', 'str_', 'string_', 'subtract', 'sum', 'swapaxes', 'sys', 'take', 'take_along_axis', 'tan', 'tanh', 'tenordot', 'test', 'testing', 'tile', 'timedelta64', 'trace', 'tracemalloc_domain', 'transpose', 'trapz', 'tri', 'tril', 'tril_indices', 'tril_indices_from', 'trim_zeros', 'triu', 'triu_indices', 'triu_indices_from', 'true_divide', 'trunc', 'typeDict', 'typeNA', 'typecodes', 'typename', 'ubyte', 'ufunc', 'uint', 'uint0', 'uint16', 'uint32', 'uint64', 'uint8', 'uintc', 'uintp', 'ulonglong', 'unicode', 'unicode_', 'unionld', 'unique', 'unpackbits', 'unravel_index', 'unsignedinteger', 'unwrap', 'use_hugepage', 'ushort', 'vander', 'var', 'vdot', 'vectorize', 'version', 'void', 'void0', 'vsplit', 'vstack', 'warnings', 'where', 'who', 'zeros', 'zeros_...
```

In [40]:

```
x = np.unique(a, return_counts = True)
x
```

Out[40]: (array([1.]), array([5], dtype=int64))

In [41]:

```
s = np.sum(a)
s
```

Out[41]: 5.0

In [42]:

```
p = np.prod(a)
p
```

Out[42]: 1.0

In [44]:

```
d = np.divide([2,5],[1,2])
d
```

Out[44]: array([2. , 2.5])

In [46]:

```
ss = np.array_split(a,2)
ss
```

```
Out[46]: [array([1., 1., 1.]), array([1., 1.])]
```

```
In [47]: sss = np.cumsum(a)

        sss
```

```
Out[47]: array([1., 2., 3., 4., 5.])
```

```
In [48]: diff = np.diff(a)

        diff
```

```
Out[48]: array([0., 0., 0., 0.])
```

## Pandas

```
In [13]: import pandas as pd
```

```
In [15]: print(dir(pd),end="")
```

```
['BooleanDtype', 'Categorical', 'CategoricalDtype', 'CategoricalIndex', 'DataFrame', 'DateOffset', 'DatetimeIndex', 'DatetimeTZDtype', 'ExcelFile', 'ExcelWriter', 'Float64Index', 'Grouper', 'HDFStore', 'Index', 'IndexSlice', 'Int16Dtype', 'Int32Dtype', 'Int64Dtype', 'Int64Index', 'Int8Dtype', 'Interval', 'IntervalDtype', 'IntervalIndex', 'MultiIndex', 'NA', 'NaT', 'NamedAgg', 'Period', 'PeriodDtype', 'PeriodIndex', 'RangeIndex', 'Series', 'SparseDtype', 'StringDtype', 'Timedelta', 'TimedeltaIndex', 'Timestamp', 'UInt16Dtype', 'UInt32Dtype', 'UInt64Dtype', 'UInt64Index', 'UInt8Dtype', '__builtins__', '__cached__', '__doc__', '__docformat__', '__file__', '__getattr__', '__git_version__', '__loader__', '__name__', '__package__', '__path__', '__spec__', '__version__', '__config__', '__hashtable__', '__is_numpy_dev__', '__lib__', '__libs__', '__np_version_under1p16__', '__np_version_under1p17__', '__np_version_under1p18__', '__testing__', '__tslib__', '__typing__', '__version__', '__api__', '__array__', '__arrays__', '__bdate_range__', '__compat__', '__concat__', '__core__', '__crosstab__', '__cut__', '__date_range__', '__describe_option__', '__errors__', '__eval__', '__factorize__', '__get_dummies__', '__get_option__', '__infer_freq__', '__interval_range__', '__io__', '__isna__', '__isnull__', '__json_normalize__', '__lreshape__', '__melt__', '__merge__', '__merge_asof__', '__merge_ordered__', '__notna__', '__notnull__', '__offsets__', '__option_context__', '__options__', '__pandas__', '__period_range__', '__pivot__', '__pivot_table__', '__plotting__', '__qcut__', '__read_clipboard__', '__read_csv__', '__read_excel__', '__read_feather__', '__read_fwf__', '__read_gbq__', '__read_hdf__', '__read_html__', '__read_json__', '__read_orc__', '__read_parquet__', '__read_pickle__', '__read_sas__', '__read_spss__', '__read_sql__', '__read_sql_query__', '__read_sql_table__', '__read_stata__', '__read_table__', '__reset_option__', '__set_eng_float_format__', '__set_option__', '__show_versions__', '__test__', '__testing__', '__timedelta_range__', '__to_datetime__', '__to_numeric__', '__to_pickle__', '__to_timedelta__', '__tries__', '__unique__', '__util__', '__value_counts__', '__wide_to_long__']
```

```
In [16]: d = pd.read_csv("info_large.csv")
```

```
In [18]: d.head()
```

```
Out[18]:
```

	Ganesh	Bhandarkar	1806554
0	Moti	Doggy	13
1	Ram	Turtle	11

```
In [19]: d.tail()
```

```
Out[19]:
```

	Ganesh	Bhandarkar	1806554
0	Moti	Doggy	13
1	Ram	Turtle	11

```
In [20]: len(d.value_counts())
```

```
Out[20]: 2
```

```
In [21]: d.nunique()
```

```
Out[21]:
```

Ganesh	2
Bhandarkar	2
1806554	2

dtype: int64

```
In [22]: d.describe()
```

```
Out[22]:
```

	1806554
count	2.000000

**1806554**

<b>mean</b>	12.000000
<b>std</b>	1.414214
<b>min</b>	11.000000
<b>25%</b>	11.500000
<b>50%</b>	12.000000
<b>75%</b>	12.500000

```
In [23]: d.isna().any()
```

```
Out[23]: Ganesh      False
         Bhandarkar  False
         1806554     False
         dtype: bool
```

```
In [24]: d.isna().sum()
```

```
Out[24]: Ganesh      0
         Bhandarkar  0
         1806554     0
         dtype: int64
```

```
In [25]: d.mean()
```

```
Out[25]: 1806554     12.0
         dtype: float64
```

## Scipy

```
In [5]: from scipy import linalg, ndimage
import cv2
import matplotlib.pyplot as plt
```

```
In [32]: mat = np.array([[2,1],[4,3]])
sc.linalg.det(mat)
```

```
Out[32]: 2.0
```

```
In [33]: linalg.inv(mat)
```

```
Out[33]: array([[ 1.5, -0.5],
               [-2. ,  1. ]])
```

```
In [34]: linalg.svd(mat)
```

```
Out[34]: (array([[-0.40455358, -0.9145143 ],
               [-0.9145143 ,  0.40455358]]),
         array([5.4649857 , 0.36596619]),
         array([[-0.81741556, -0.57604844],
               [-0.57604844,  0.81741556]]))
```

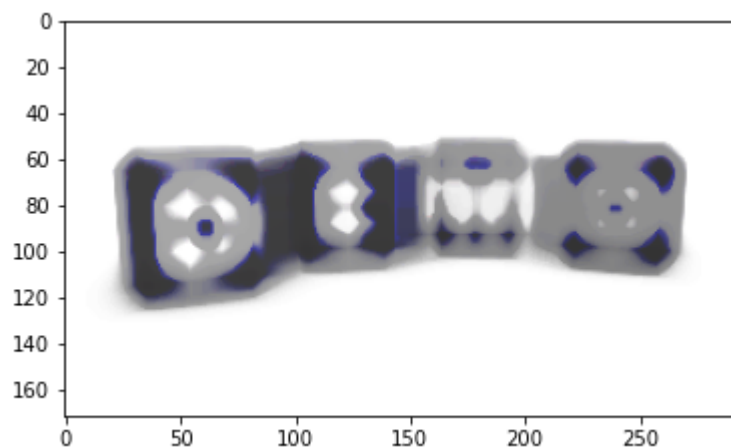
```
In [12]: image = cv2.imread('demo.png')
a = ndimage.rotate(image, 18)
plt.imshow(a)
```

```
Out[12]: <matplotlib.image.AxesImage at 0x16159d818b0>
```



```
In [13]: b = ndimage.median_filter(image,20)
plt.imshow(b)
```

```
Out[13]: <matplotlib.image.AxesImage at 0x16159dd8460>
```



```
In [21]: c = ndimage.binary_opening(image)
d = ndimage.binary_erosion(image)
e = ndimage.binary_dilation(image)
f = ndimage.binary_closing(image)
print(c,d,e,f,end="")
```

```
[[False False False]
```



```
[False True False]
[False True False]
...
[False True False]
[False True False]
[False False False]]

[[False True False]
 [ True True True]
 [ True True True]
...
 [ True True True]
 [ True True True]
 [False True False]]

[[False True False]
 [ True True True]
 [ True True True]
...
 [ True True True]
 [ True True True]
 [False True False]]

[[False True False]
 [ True True True]
 [ True True True]
...
 [ True True True]
 [ True True True]
 [False True False]]

[[False False False]
 [False True False]
 [False True False]
...
 [False True False]
 [False True False]
 [False False False]] [[False False False]
 [False False False]
```

```
[False False False]
...
[False False False]
[False False False]
[False False False]]

[[False False False]
 [False True False]
 [False True False]
 ...
 [False True False]
 [False True False]
 [False False False]]

[[False False False]
 [False True False]
 [False True False]
 ...
 [False True False]
 [False True False]
 [False False False]]

...

[[False False False]
 [False True False]
 [False True False]
 ...
 [False True False]
 [False True False]
 [False False False]]

[[False False False]
 [False True False]
 [False True False]
 ...
 [False True False]
 [False True False]
 [False False False]]

[[False False False]
 [False False False]
 [False False False]
 ...
 [False False False]
 [False False False]
 [False False False]] [[ True True True]
 [ True True True]
 [ True True True]
```

```
...
[ True  True  True]
[ True  True  True]
[ True  True  True]]

[[ True  True  True]
 [ True  True  True]
 [ True  True  True]

...
[ True  True  True]
[ True  True  True]
[ True  True  True]]

[[ True  True  True]
 [ True  True  True]
 [ True  True  True]

...
[ True  True  True]
[ True  True  True]
[ True  True  True]]

[[ True  True  True]
 [ True  True  True]
 [ True  True  True]

...
[ True  True  True]
[ True  True  True]
[ True  True  True]] [[False False False]
[False False False]
[False False False]
...
```

```
[False False False]
[False False False]
[False False False]]
```

```
[[False False False]
 [False  True False]
 [False  True False]
 ...
 [False  True False]
 [False  True False]
 [False False False]]
```

```
[[False False False]
 [False  True False]
 [False  True False]
 ...
 [False  True False]
 [False  True False]
 [False False False]]
```

```
...
```

```
[[False False False]
 [False  True False]
 [False  True False]
 ...
 [False  True False]
 [False  True False]
 [False False False]]
```

```
[[False False False]
 [False  True False]
 [False  True False]
 ...
 [False  True False]
 [False  True False]
 [False False False]]
```

```
[[False False False]
 [False False False]
 [False False False]
 ...
 [False False False]
 [False False False]]
```

# Time

```
In [28]: import time,calendar
```

```
In [25]: localtime = time.localtime(time.time())
print(localtime)
```

```
time.struct_time(tm_year=2021, tm_mon=2, tm_mday=15, tm_hour=10, tm_min=33, tm_sec=5, tm_wday=0, tm_yday=46,
tm_isdst=0)
```

```
In [32]: cal = calendar.month(2021, 2)
print(cal)
```

```
February 2021
Mo Tu We Th Fr Sa Su
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
```

```
In [34]: localtime = time.asctime( time.localtime(time.time()))
localtime
```

```
Out[34]: 'Mon Feb 15 10:35:55 2021'
```

```
In [35]: srctime = time.strptime("30 Nov 00", "%d %b %y")
srctime
```

```
Out[35]: time.struct_time(tm_year=2000, tm_mon=11, tm_mday=30, tm_hour=0, tm_min=0, tm_sec=0, tm_wday=3, tm_yday=335,
tm_isdst=-1)
```

## MoviePy

In [83]:

```
!pip install moviepy  
#from moviepy.editor import *
```

```
Requirement already satisfied: moviepy in c:\users\kiit\anaconda3\lib\site-packages (1.0.3)  
Requirement already satisfied: imageio-ffmpeg>=0.2.0; python_version >= "3.4" in c:\users\kiit\anaconda3\lib\site-packages (from moviepy) (0.4.3)  
Requirement already satisfied: proglog<=1.0.0 in c:\users\kiit\anaconda3\lib\site-packages (from moviepy) (0.1.9)  
Requirement already satisfied: tqdm<5.0,>=4.11.2 in c:\users\kiit\anaconda3\lib\site-packages (from moviepy) (4.50.2)  
Requirement already satisfied: numpy; python_version >= "2.7" in c:\users\kiit\anaconda3\lib\site-packages (from moviepy) (1.19.2)  
Requirement already satisfied: decorator<5.0,>=4.0.2 in c:\users\kiit\anaconda3\lib\site-packages (from moviepy) (4.4.2)  
Requirement already satisfied: imageio<3.0,>=2.5; python_version >= "3.4" in c:\users\kiit\anaconda3\lib\site-packages (from moviepy) (2.9.0)  
Requirement already satisfied: requests<3.0,>=2.8.1 in c:\users\kiit\anaconda3\lib\site-packages (from moviepy) (2.24.0)  
Requirement already satisfied: pillow in c:\users\kiit\anaconda3\lib\site-packages (from imageio<3.0,>=2.5; python_version >= "3.4"->moviepy) (8.0.1)  
Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 in c:\users\kiit\anaconda3\lib\site-packages (from requests<3.0,>=2.8.1->moviepy) (1.25.11)  
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\kiit\anaconda3\lib\site-packages (from requests<3.0,>=2.8.1->moviepy) (3.0.4)  
Requirement already satisfied: certifi>=2017.4.17 in c:\users\kiit\anaconda3\lib\site-packages (from requests<3.0,>=2.8.1->moviepy) (2020.6.20)  
Requirement already satisfied: idna<3,>=2.5 in c:\users\kiit\anaconda3\lib\site-packages (from requests<3.0,>=2.8.1->moviepy) (2.10)
```

In [93]:

```
from moviepy.editor import *  
clip = VideoFileClip("myHoliday.mp4").subclip(50, 60)  
  
clip = clip.volumex(0.8)  
  
txt_clip = TextClip("My Holidays 2020", fontsize=70, color='white')  
  
txt_clip = txt_clip.set_pos('center').set_duration(15)  
  
video = CompositeVideoClip([clip, txt_clip])  
  
video.write_videofile("myHoliday.mp4")
```

## Requests

In [1]:

```
import requests
```

In [2]:

```
response = requests.get('https://api.github.com')
```

In [3]:

```
response
```

Out[3]: &lt;Response [200]&gt;

```
In [4]: if response.status_code == 200:
        print('Success!')
        elif response.status_code == 404:
            print('Not Found.')
```

Success!

```
In [5]: response.json()
```

```
Out[5]: {'current_user_url': 'https://api.github.com/user',
'current_user_authorizations_html_url': 'https://github.com/settings/connections/applications{/client_id}',
'authorizations_url': 'https://api.github.com/authorizations',
'code_search_url': 'https://api.github.com/search/code?q={query}{&page,per_page,sort,order}',
'commit_search_url': 'https://api.github.com/search/commits?q={query}{&page,per_page,sort,order}',
'emails_url': 'https://api.github.com/user/emails',
'emojis_url': 'https://api.github.com/emojis',
'events_url': 'https://api.github.com/events',
'feeds_url': 'https://api.github.com/feeds',
'followers_url': 'https://api.github.com/user/followers',
'following_url': 'https://api.github.com/user/following{/target}',
'gists_url': 'https://api.github.com/gists{/gist_id}',
'hub_url': 'https://api.github.com/hub',
'issue_search_url': 'https://api.github.com/search/issues?q={query}{&page,per_page,sort,order}',
'issues_url': 'https://api.github.com/issues',
'keys_url': 'https://api.github.com/user/keys',
'label_search_url': 'https://api.github.com/search/labels?q={query}&repository_id={repository_id}{&page,per_page}',
'notifications_url': 'https://api.github.com/notifications',
'organization_url': 'https://api.github.com/orgs/{org}',
'organization_repositories_url': 'https://api.github.com/orgs/{org}/repos{?type,page,per_page,sort}',
'organization_teams_url': 'https://api.github.com/orgs/{org}/teams',
'public_gists_url': 'https://api.github.com/gists/public',
'rate_limit_url': 'https://api.github.com/rate_limit',
'repository_url': 'https://api.github.com/repos/{owner}/{repo}',
'repository_search_url': 'https://api.github.com/search/repositories?q={query}{&page,per_page,sort,order}',
'current_user_repositories_url': 'https://api.github.com/user/repos{?type,page,per_page,sort}',
'starred_url': 'https://api.github.com/user/starred{/owner}/{repo}',
'starred_gists_url': 'https://api.github.com/gists/starred',
'user_url': 'https://api.github.com/users/{user}',
'user_organizations_url': 'https://api.github.com/user/orgs',
'user_repositories_url': 'https://api.github.com/users/{user}/repos{?type,page,per_page,sort}',
'user_search_url': 'https://api.github.com/search/users?q={query}{&page,per_page,sort,order}'}
```



# Cpython

```
In [72]: !pip install Cython
```

Requirement already satisfied: Cython in c:\users\kiit\anaconda3\lib\site-packages (0.29.21)

```
In [96]: %load_ext cython
```

```
In [97]: %%cython

cdef int a = 0
for i in range(10):
    a += i
print(a)
```

45

# Bokeh

```
In [9]: import bokeh
import matplotlib.pyplot as plt
```

In [15]:

```
import numpy as np

from bokeh.layouts import gridplot
from bokeh.plotting import figure, output_file, show

N = 100
x = np.linspace(0, 4*np.pi, N)
y0 = np.sin(x)
y1 = np.cos(x)
y2 = np.sin(x) + np.cos(x)
output_file("linked_panning.html")

s1 = figure(width=250, plot_height=250, title=None)
s1.circle(x, y0, size=10, color="navy", alpha=0.5)

s2 = figure(width=250, height=250, x_range=s1.x_range, y_range=s1.y_range,
title=None)
s2.triangle(x, y1, size=10, color="firebrick", alpha=0.5)

s3 = figure(width=250, height=250, x_range=s1.x_range, title=None)
s3.square(x, y2, size=10, color="olive", alpha=0.5)

p = gridplot([[s1, s2, s3]], toolbar_location=None)
```

In [16]:

```
import numpy as np

from bokeh.plotting import figure, output_file, show

N = 4000
x = np.random.random(size=N) * 100
y = np.random.random(size=N) * 100
radii = np.random.random(size=N) * 1.5
colors = [
    "%02x%02x%02x" % (int(r), int(g), 150) for r, g in zip(50+2*x, 30+2*y)
]

output_file("color_scatter.html", title="color_scatter.py example", mode="cdn")

TOOLS = "crosshair,pan,wheel_zoom,box_zoom,reset,box_select,lasso_select"

p = figure(tools=TOOLS, x_range=(0, 100), y_range=(0, 100))

p.circle(x, y, radius=radii, fill_color=colors, fill_alpha=0.6,
line_color=None)

show(p)
```

# TextBlob

In [18]:

```
! pip install textblob
```

Collecting textblob

Downloading textblob-0.15.3-py2.py3-none-any.whl (636 kB)

Requirement already satisfied: nltk>=3.1 in c:\users\kiit\anaconda3\lib\site-packages (from textblob) (3.5)

Requirement already satisfied: tqdm in c:\users\kiit\anaconda3\lib\site-packages (from nltk>=3.1->textblob) (4.50.2)

Requirement already satisfied: click in c:\users\kiit\anaconda3\lib\site-packages (from nltk>=3.1->textblob) (7.1.2)

Requirement already satisfied: regex in c:\users\kiit\anaconda3\lib\site-packages (from nltk>=3.1->textblob) (2020.10.15)

Requirement already satisfied: joblib in c:\users\kiit\anaconda3\lib\site-packages (from nltk>=3.1->textblob) (0.17.0)

Installing collected packages: textblob

Successfully installed textblob-0.15.3

In [24]:

```
from textblob import TextBlob
```

```
wiki = TextBlob("Python is a high-level, general-purpose programming  
language.")
```

In [26]:

```
wiki.tags
```

Out[26]:

```
[('Python', 'NNP'),  
 ('is', 'VBZ'),  
 ('a', 'DT'),  
 ('high-level', 'JJ'),  
 ('general-purpose', 'JJ'),  
 ('programming', 'NN'),  
 ('language', 'NN')]
```

```
In [27]: wiki.sentiment
```

```
Out[27]: Sentiment(polarity=0.0, subjectivity=0.0)
```

```
In [28]: wiki.words
```

```
Out[28]: WordList(['Python', 'is', 'a', 'high-level', 'general-purpose', 'programming', 'language'])
```

```
In [29]: wiki.sentences
```

```
Out[29]: [Sentence("Python is a high-level, general-purpose programming language.")]
```

## PyAudioAnalysis

```
In [57]: !pip install plotly
```

```
Collecting plotly
  Downloading plotly-4.14.3-py2.py3-none-any.whl (13.2 MB)
Requirement already satisfied: six in c:\users\kiit\anaconda3\lib\site-packages (from plotly) (1.15.0)
Collecting retrying>=1.3.3
  Downloading retrying-1.3.3.tar.gz (10 kB)
Building wheels for collected packages: retrying
  Building wheel for retrying (setup.py): started
  Building wheel for retrying (setup.py): finished with status 'done'
  Created wheel for retrying: filename=retrying-1.3.3-py3-none-any.whl size=11434 sha256=06af50b83549c20d4d8
18b4ad03467510e336a98c7e81869ad87cc3711828278
  Stored in directory: c:\users\kiit\appdata\local\pip\cache\wheels\c4\a7\48\0a434133f6d56e878ca511c0e6c3832
6907c0792f67b476e56
Successfully built retrying
Installing collected packages: retrying, plotly
Successfully installed plotly-4.14.3 retrying-1.3.3
```

In [82]:

```
from pyAudioAnalysis import audioTrainTest as aT
aT.extract_features_and_train(["classifierData/music", "classifierData/speech"],
1.0, 1.0, aT.shortTermWindow, aT.shortTermStep, "svm", "svmSMtemp", False)
aT.file_classification("python scripts/cool.wav", "svmSMtemp", "svm")
```

```
trainSVM_feature ERROR: No data found in any input folder!
fileClassification: input model_name not found!
```

Out[82]: (-1, -1, -1)

## gTTS

In [31]:

```
! pip install gTTS
```

Collecting gTTS

Downloading gTTS-2.2.2-py3-none-any.whl (25 kB)

Requirement already satisfied: click in c:\users\kiit\anaconda3\lib\site-packages (from gTTS) (7.1.2)

Requirement already satisfied: requests in c:\users\kiit\anaconda3\lib\site-packages (from gTTS) (2.24.0)

Requirement already satisfied: six in c:\users\kiit\anaconda3\lib\site-packages (from gTTS) (1.15.0)

Requirement already satisfied: idna<3,>=2.5 in c:\users\kiit\anaconda3\lib\site-packages (from requests->gTTS) (2.10)

Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in c:\users\kiit\anaconda3\lib\site-packages (from requests->gTTS) (1.25.11)

Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\kiit\anaconda3\lib\site-packages (from requests->gTTS) (3.0.4)

Requirement already satisfied: certifi>=2017.4.17 in c:\users\kiit\anaconda3\lib\site-packages (from requests->gTTS) (2020.6.20)

Installing collected packages: gTTS

Successfully installed gTTS-2.2.2

In [1]:

```
from gtts import gTTS
tts = gTTS('hello')
tts.save('hello.mp3')
```

In [1]:

```
# !pip install pyaudio
```

In [26]:

```
import speech_recognition as sr
import pyttsx3

# Initialize the recognizer
r = sr.Recognizer()

# Function to convert text to
# speech
def SpeakText(command):

    # Initialize the engine
    engine = pyttsx3.init()
    engine.say(command)
    engine.runAndWait()

while(1):

    # Exception handling to handle
    # exceptions at the runtime
    try:

        # use the microphone as source for input.
        with sr.Microphone() as source2:
```



```
        r.adjust_for_ambient_noise(source2, duration=0.2)

        #listens for the user's input
        audio2 = r.listen(source2)

        # Using ggogle to recognize audio
        MyText = r.recognize_google(audio2)
        MyText = MyText.lower()

        print("Did you say "+MyText)
        SpeakText(MyText)

    except sr.RequestError as e:
        print("Could not request results; {0}".format(e))

    except sr.UnknownValueError:
        print("unknown error occured")
```

In [ ]: