Year
2022

AI Visual Inspection system for road defection

LESSON 1

YOUTH COLLEGE (INTERNATIONAL)

Reference Websites

Python Official	
https://www.python.org/	
Google Colab	
https://colab.research.google.com/	
Python Exercises	
https://www.w3resource.com/python-exercises/ https://www.w3schools.com/python/default.asp	
GitHub	
https://github.com/garyprojects/road_detect	

1. Introduction

Examples of Road Surface Defects

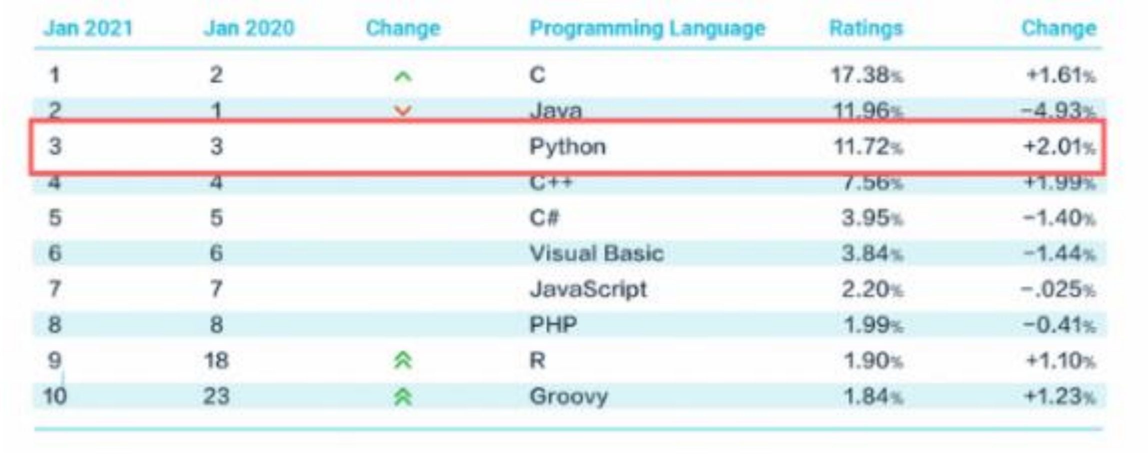
Potholes

Cracking

Decolored

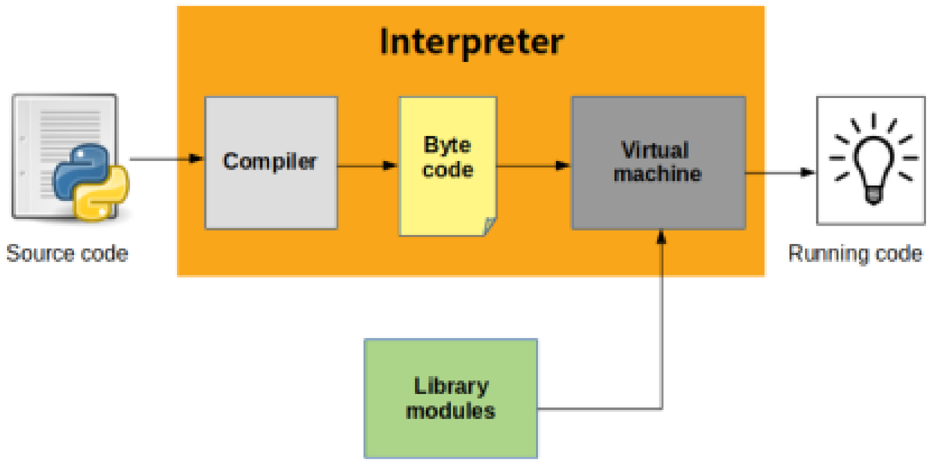
What is Python?

Popular programming language in 2021



Jan 2021	Jan 2020	Change	Programming Language	Ratings	Change
1	2	↑	C	17.38%	+1.61%
2	1	↓	Java	11.96%	−4.93%
3	3		Python	11.72%	+2.01%
4	4		C++	7.56%	+1.99%
5	5		C#	3.95%	−1.40%
6	6		Visual Basic	3.84%	−1.44%
7	7		JavaScript	2.20%	−.025%
8	8		PHP	1.99%	−0.41%
9	18	⇈	R	1.90%	+1.10%
10	23	⇈	Groovy	1.84%	+1.23%

Interpreted language

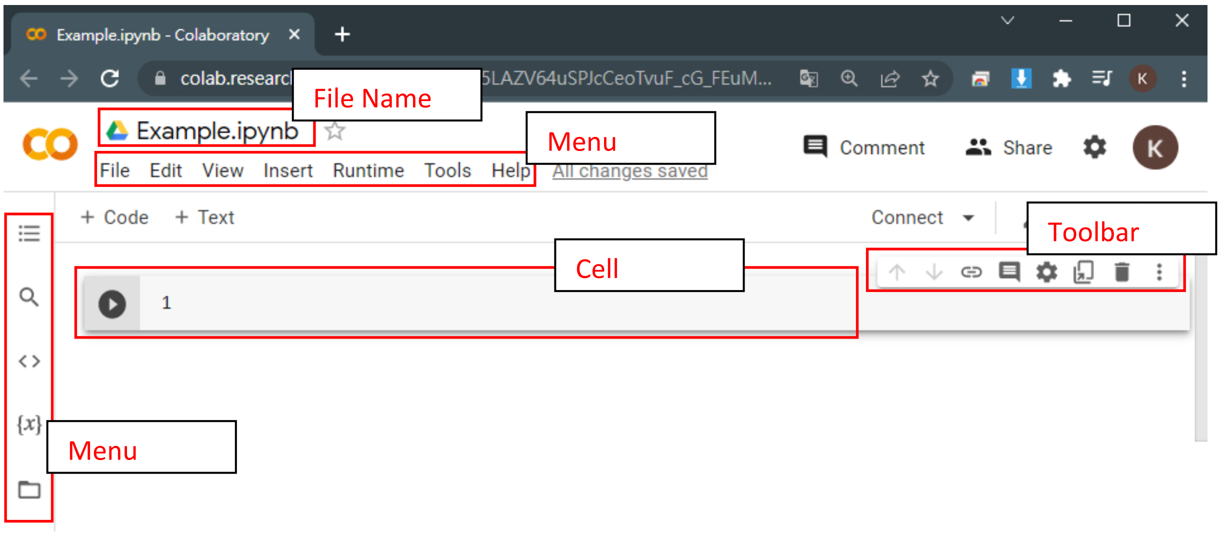


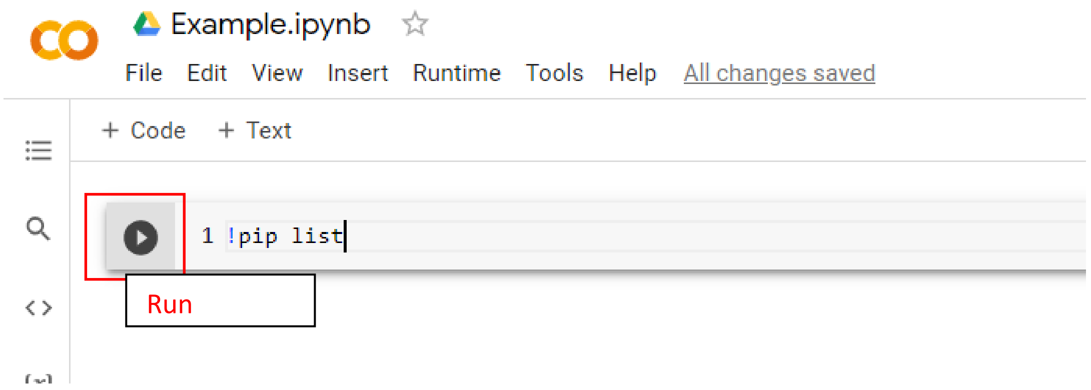
High-level programming language

```
10101001111100
10101000001000
......
......
10101000001000
```

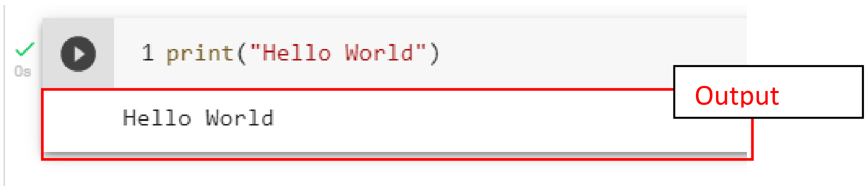
```
if a >b:
    print( ‘a is greater than b’ )
else:
    print( ‘b is greater than b’ )
```

2. Using Google Colab


Example.ipynb - Colaboratory
File Name
Example.ipynb
Menu
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
Connect
Toolbar
Cell
Menu



```
!pip list
```



```
print("Hello World")
```

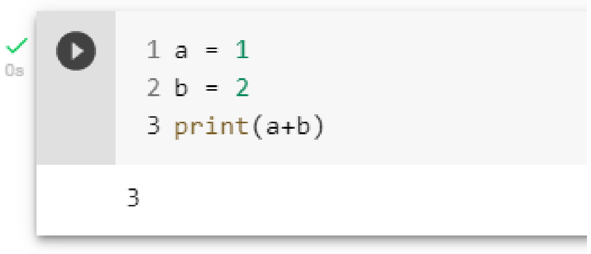
```
Hello World
```

3. Variables and datatypes

Python identifier

name	valid or invalid
ab10c	valid
abc_DE	valid
_	valid
_abc	valid
99	invalid
x+y	invalid
for	invalid
a@	invalid
9abc	invalid

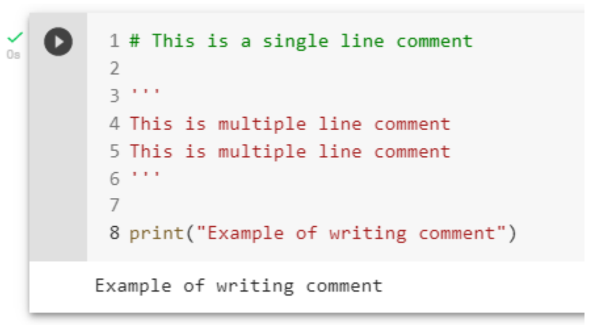
Statement



```
a = 1
b = 2
print(a+b)
```

```
3
```

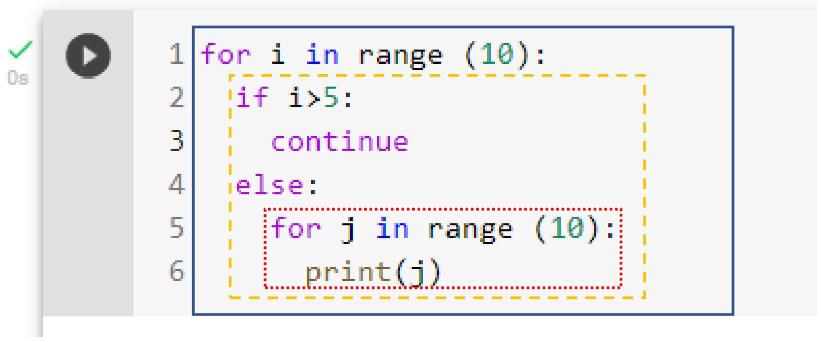
Comment



```
# This is a single line comment

'''
This is multiple line comment
This is multiple line comment
'''

print("Example of writing comment")
```

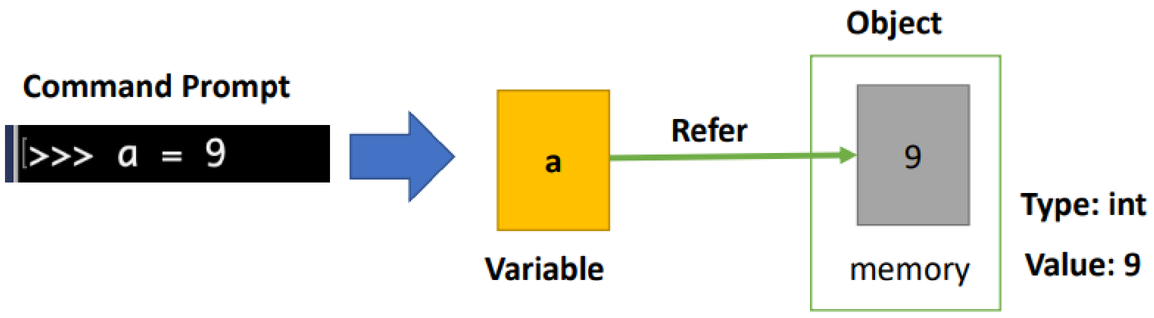
```
Example of writing comment
```

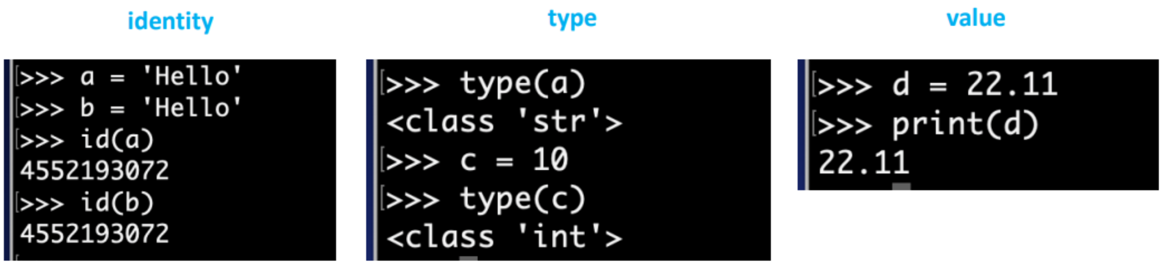
Indentation



```
for i in range (10):
  if i>5:
    continue
  else:
    for j in range (10):
      print(j)
```

Datatype



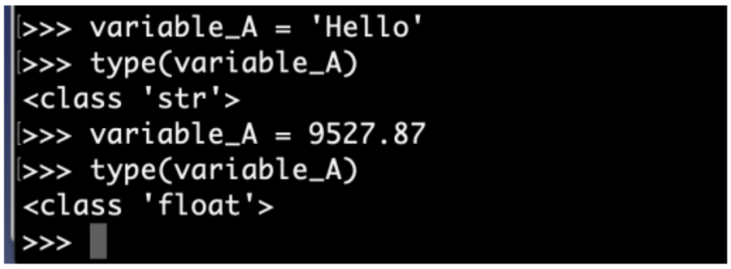


```
>>> a = 'Hello'
>>> b = 'Hello'
>>> id(a)
4552193072
>>> id(b)
4552193072
```

```
>>> type(a)
<class 'str'>
>>> c = 10
>>> type(c)
<class 'int'>
```

```
>>> d = 22.11
>>> print(d)
22.11
```

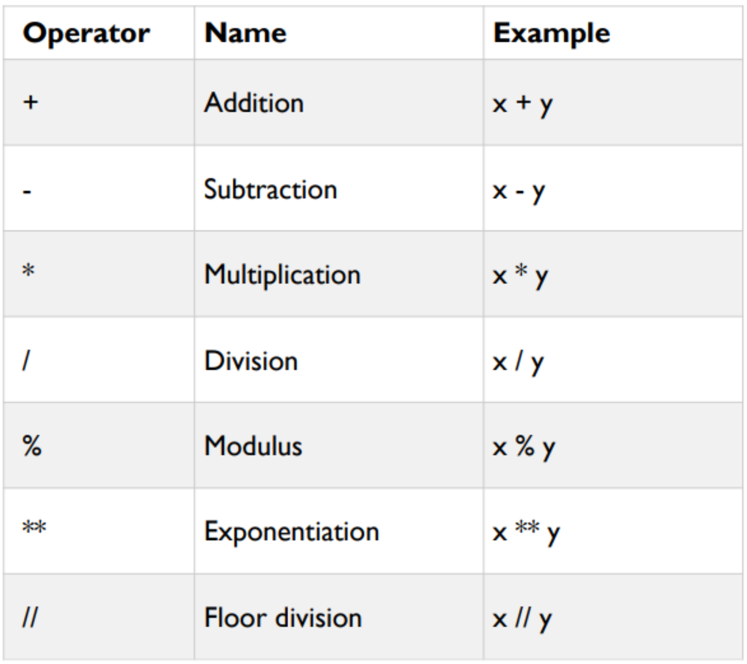
dynamic Data Type



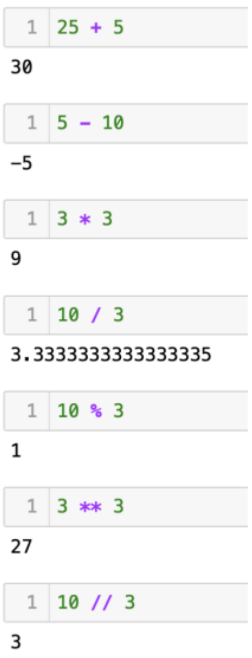
```
>>> variable_A = 'Hello'
>>> type(variable_A)
<class 'str'>
>>> variable_A = 9527.87
>>> type(variable_A)
<class 'float'>
>>> 
```

int	float
5	0.0
22	22.5
-3	-3.89
0b1010	-3.
-0b1011	18.3e+9
0x220	-18.3e+9
-0x220	

Arithmetic Operator



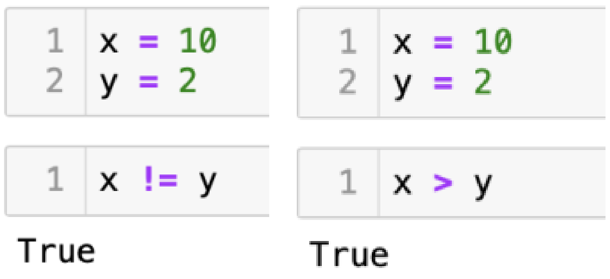
Operator	Name	Example
+	Addition	x + y
-	Subtraction	x - y
*	Multiplication	x * y
/	Division	x / y
%	Modulus	x % y
**	Exponentiation	x ** y
//	Floor division	x // y



```
25 + 5
```
```
30
```
```
5 - 10
```
```
-5
```
```
3 * 3
```
```
9
```
```
10 / 3
```
```
3.3333333333333335
```
```
10 % 3
```
```
1
```
```
3 ** 3
```
```
27
```
```
10 // 3
```
```
3
```

Comparison Operator

Operator	Example	Description
==	x == y	If the values of two operands are equal, then the condition becomes true
!=	x != y	If values of two operands are not equal, then condition becomes true
>	x > y	If the value of left operand is greater than the value of right operand, then condition becomes true.
<	x < y	If the value of left operand is less than the value of right operand, then condition becomes true.
>=	x >= y	If the value of left operand is greater than or equal to the value of right operand, then condition becomes true.
<=	x <= y	If the value of left operand is less than or equal to the value of right operand, then condition becomes true.



```
x = 10
y = 2
```
```
x != y
```
```
True
```

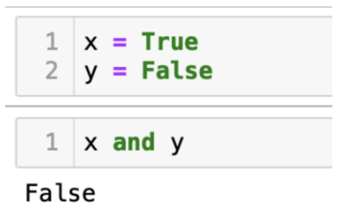
```
x = 10
y = 2
```
```
x > y
```
```
True
```

Assignment Operator

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3
&=	x &= 3	x = x & 3
|=	x |= 3	x = x | 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

Logical Operator

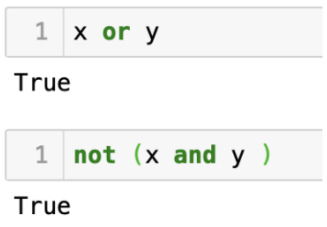
Operator	Description	Example
and	Returns True if both statements are true	x and y
or	Returns True if one of the statements is true	x or y
not	Reverse the result, returns False if the result is true	not(x and y)



```
x = True
y = False
```

```
x and y
```

```
False
```



```
x or y
```

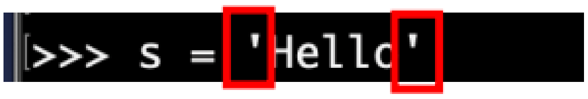
```
True
```

```
not (x and y )
```

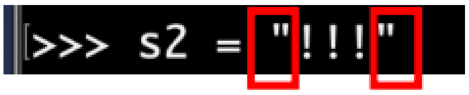
```
True
```

String

- Index



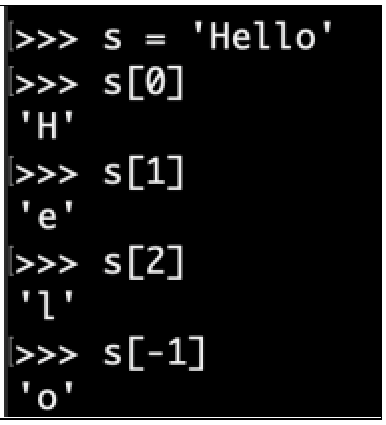
```
>>> s = 'Hello'
```



```
>>> s2 = "!!!"
```

Example: s = 'Hello'

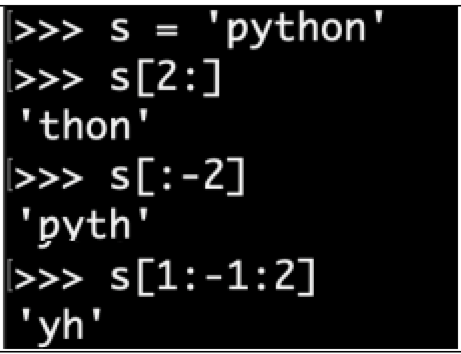
String	H	e	l	l	o
Index from head	0	1	2	3	4
Index from tail	-5	-4	-3	-2	-1



```
>>> s = 'Hello'
>>> s[0]
'H'
>>> s[1]
'e'
>>> s[2]
'l'
>>> s[-1]
'o'
```

- Slicing

s[start : end : interval]



```
>>> s = 'python'
>>> s[2:]
'thon'
>>> s[:-2]
'pyth'
>>> s[1:-1:2]
'yh'
```

Operation	Description
s[i]	Index i of string s
s[i:j]	Slice from index i to j
s[i:j:k]	Slice from index i to j with interval k
s + s2	Connect string s with string s2
s*n or n*s	Multiply n times of string s
len(s)	Length of string s
min(s)	The minimum value of string s
max(s)	The maximum value of string s
x not in s	If object x is not in the string s, return True.
x in s	If object x is in the string s, return True.

List

<u>Example:</u> L = [‘Hello’, ‘Hi’, ‘Hey’, ‘Yo’, ‘Sup’]

List	Hello	Hi	Hey	Yo	Sup
Index from head	0	1	2	3	4
Index from tail	-5	-4	-3	-2	-1

```python
# List + slicing
weekdays = ['Mon', 'Tue', 'Wed', 'Thurs', 'Fri']
weekdays[2:4]
```

```
['Wed', 'Thurs']
```

```python
# List + slicing
weekdays = ['Mon', 'Tue', 'Wed', 'Thurs', 'Fri']
weekdays[1:]
```

```
['Tue', 'Wed', 'Thurs', 'Fri']
```

```python
# List + slicing
weekdays = ['Mon', 'Tue', 'Wed', 'Thurs', 'Fri']
weekdays[::2]
```

```
['Mon', 'Wed', 'Fri']
```

```python
# List + append
weekdays.append('SAT')
print(weekdays)
```

```
['Mon', 'Tue', 'Wed', 'Thurs', 'Fri', 'SAT']
```

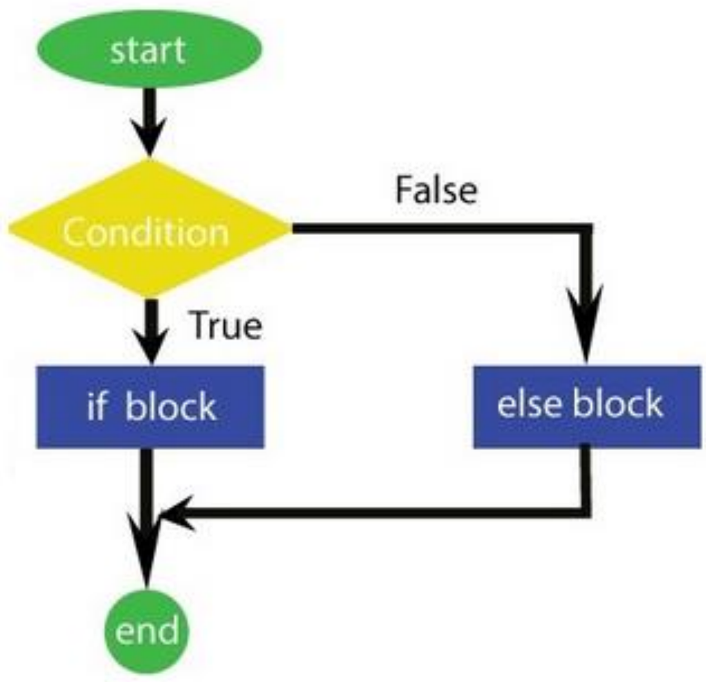
```python
# List + append
weekdays = ['Mon', 'Tue', 'Wed', 'Thurs', 'Fri']
weekend = ['Sat', 'Sun']

weekdays.append(weekend)
print(weekdays)
```

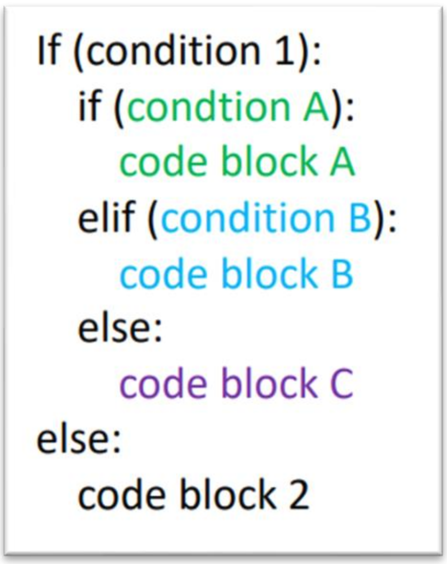
```
['Mon', 'Tue', 'Wed', 'Thurs', 'Fri', ['Sat', 'Sun']]
```

```python
# List + extend
weekdays = ['Mon', 'Tue', 'Wed', 'Thurs', 'Fri']
weekend = ['Sat', 'Sun']

weekdays.extend(weekend)
print(weekdays)
```

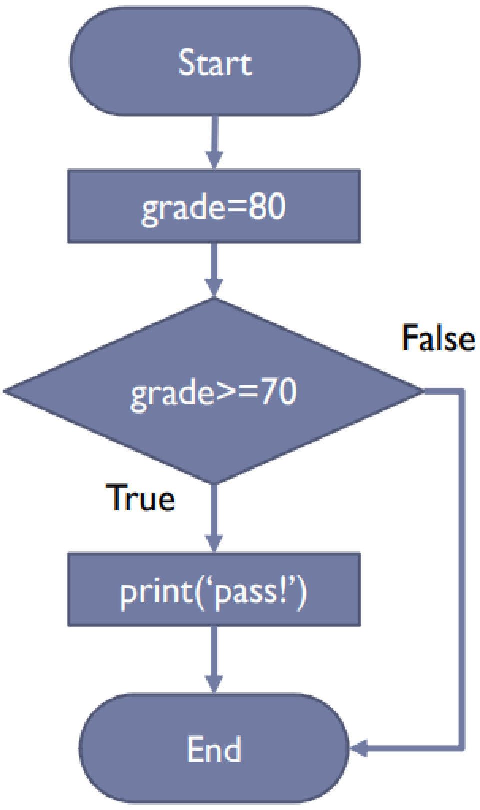
```
['Mon', 'Tue', 'Wed', 'Thurs', 'Fri', 'Sat', 'Sun']
```

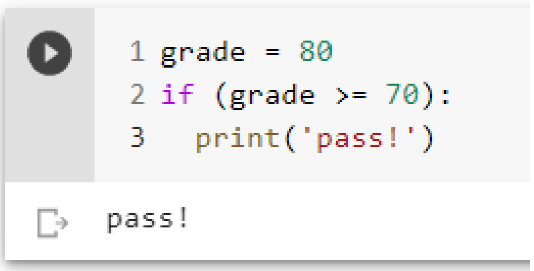
4. Decision (if/else)





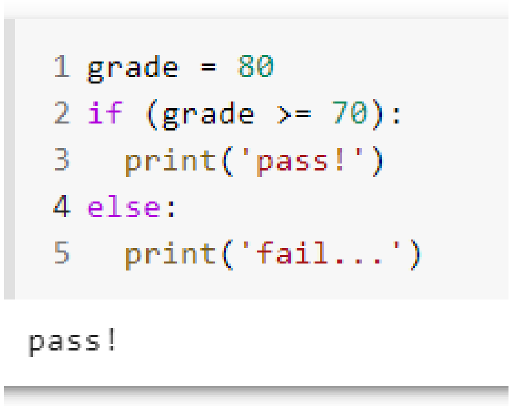
```
If (condition 1):
    if (condtion A):
        code block A
    elif (condition B):
        code block B
    else:
        code block C
else:
    code block 2
```





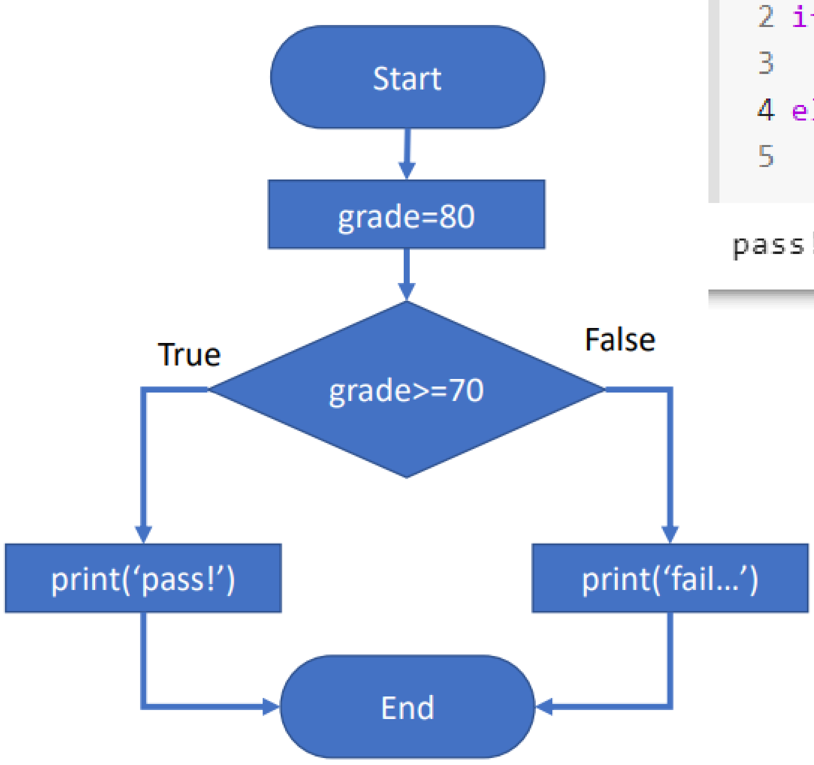
```
grade = 80
if (grade >= 70):
  print('pass!')
```

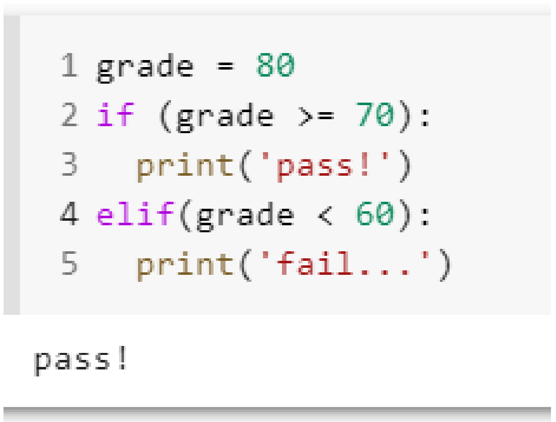
```
pass!
```



```
grade = 80
if (grade >= 70):
  print('pass!')
else:
  print('fail...')
```

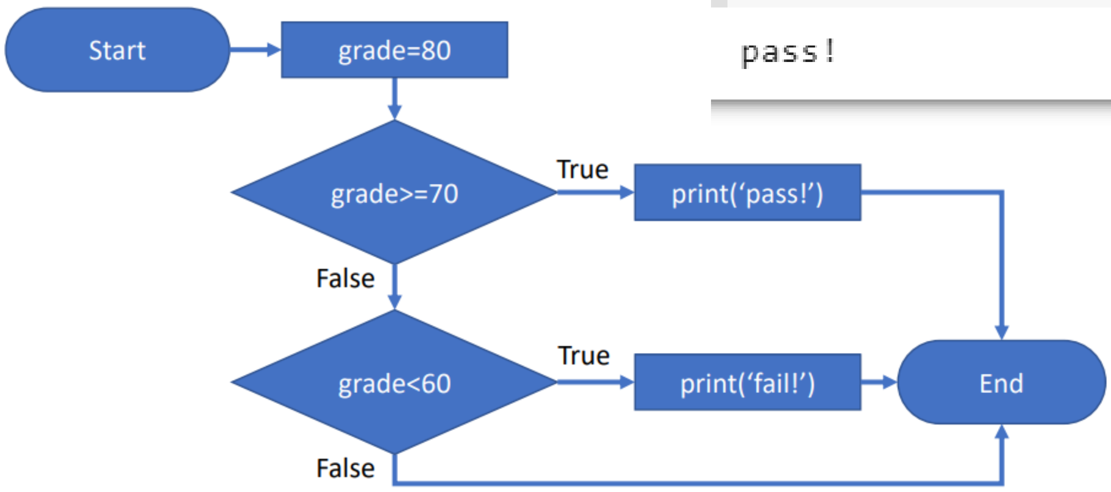
```
pass!
```





```
grade = 80
if (grade >= 70):
  print('pass!')
elif(grade < 60):
  print('fail...')
```

```
pass!
```



5. Loop (for/ while)

```
while (condition):
    code block
```

```
answer = 3
guess = 0

while guess != answer:
    guess = int(input('Please make guess during 1~6. : '))
    if guess > answer:
        print('Hint: bigger than the answer.')
    elif guess < answer:
        print('Hint: smaller than the answer.')
    else:
        print('Bingo!')
```

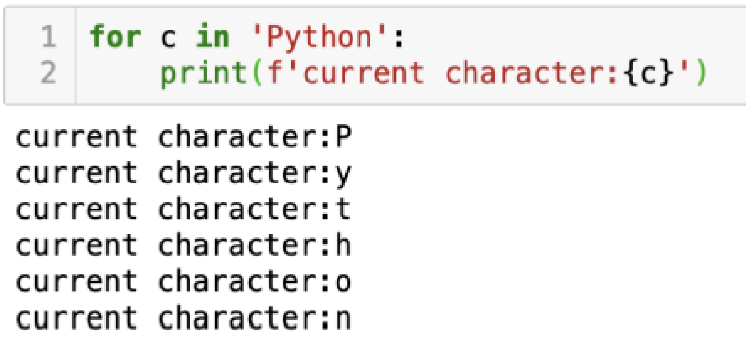
```
Please make guess during 1~6. : 1
Hint: smaller than the answer.
Please make guess during 1~6. : 2
Hint: smaller than the answer.
Please make guess during 1~6. : 3
Bingo!
```

```
while (condition A):
    code block
    if (condition B):
        break
```

```
answer = 3
guess = 0

while True:
    guess = int(input('Please make guess during 1~6. : '))
    if guess > answer:
        print('Hint: bigger than the answer.')
    elif guess < answer:
        print('Hint: smaller than the answer.')
    else:
        print('Bingo!')
        break
```

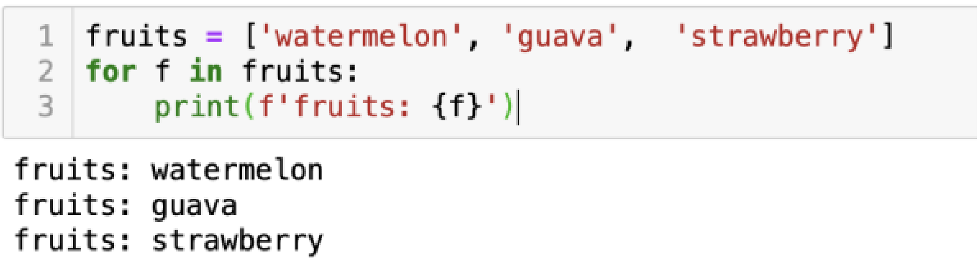
```
Please make guess during 1~6. : 1
Hint: smaller than the answer.
Please make guess during 1~6. : 2
Hint: smaller than the answer.
Please make guess during 1~6. : 3
Bingo!
```

```
for <variable> in (sequence):
    code block
```



```python
for c in 'Python':
    print(f'current character:{c}')
```

```
current character:P
current character:y
current character:t
current character:h
current character:o
current character:n
```

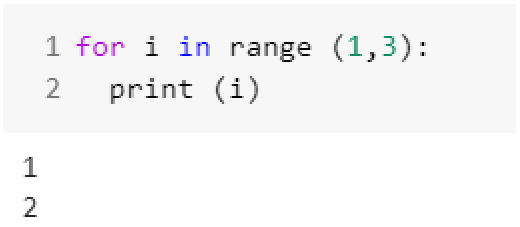


```python
fruits = ['watermelon', 'guava',  'strawberry']
for f in fruits:
    print(f'fruits: {f}')
```

```
fruits: watermelon
fruits: guava
fruits: strawberry
```

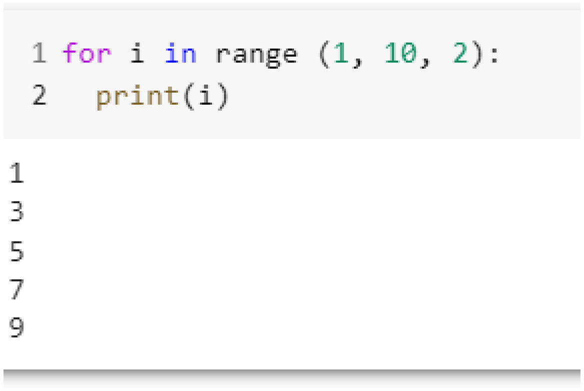
```
for <variable> in range(number) :
    code block
```

```
for <variable> in range(start, end, step) :
    code block
```



```python
for i in range (1,3):
  print (i)
```

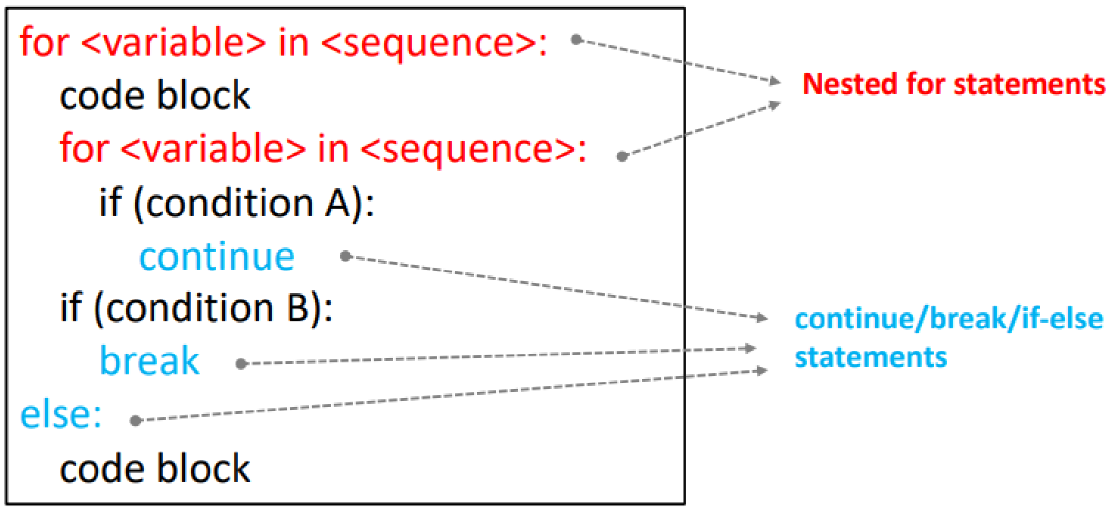
```
1
2
```



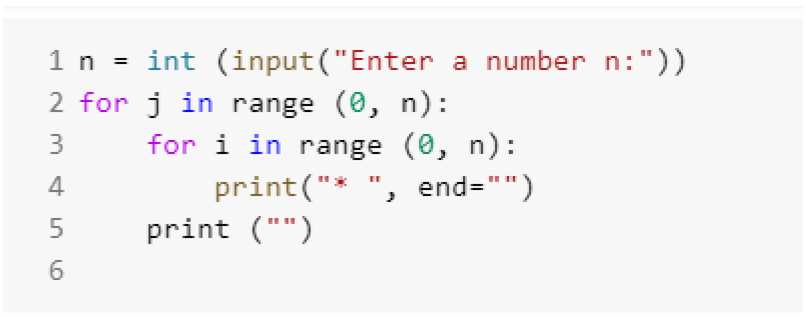
```python
for i in range (1, 10, 2):
  print(i)
```

```
1
3
5
7
9
```

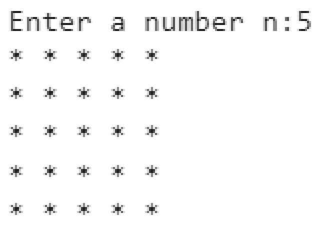
Nested loop



```
for <variable> in <sequence>:
    code block
    for <variable> in <sequence>:
        if (condition A):
            continue
    if (condition B):
        break
else:
    code block
```



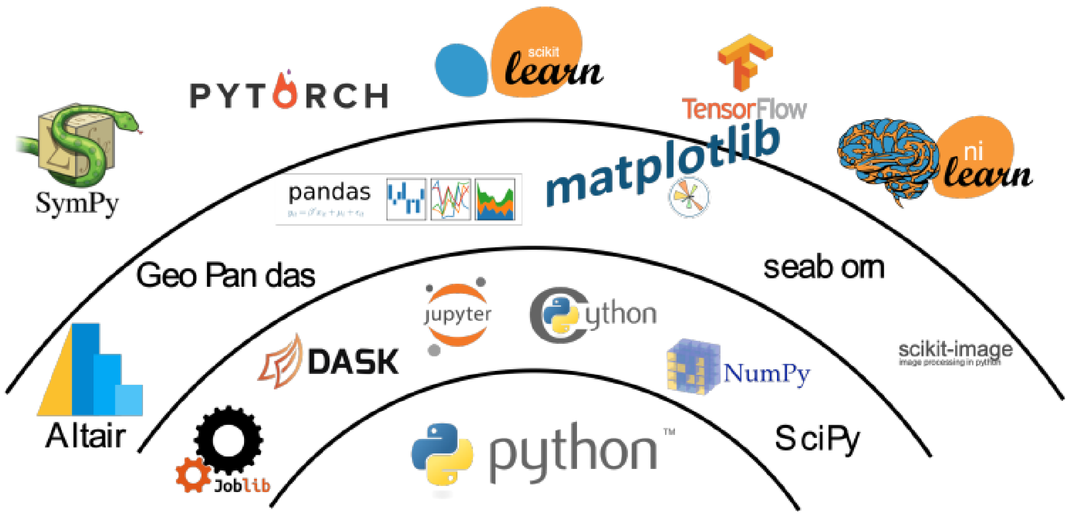
```python
n = int (input("Enter a number n:"))
for j in range (0, n):
    for i in range (0, n):
        print("* ", end="")
    print ("")

```

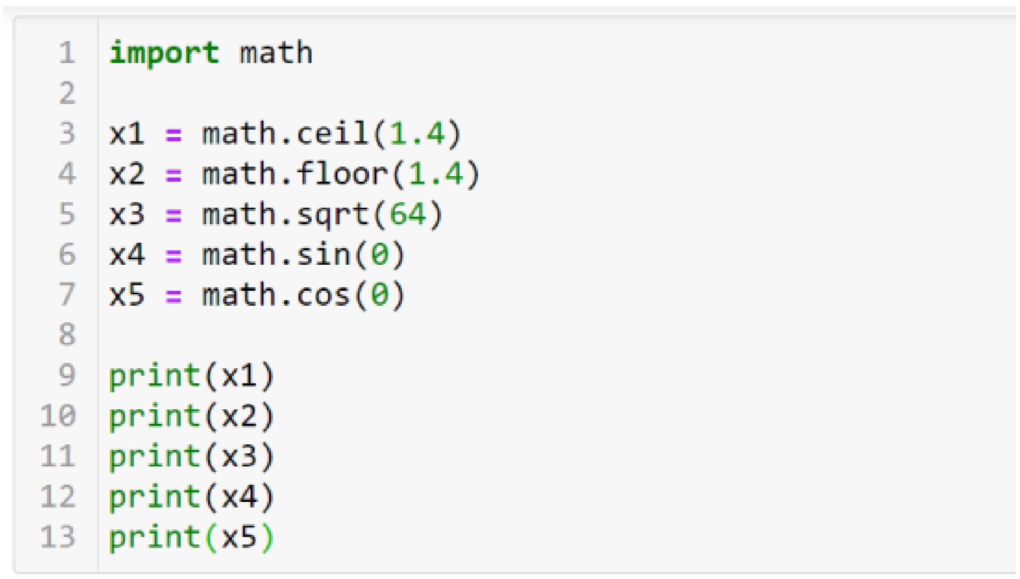


```
Enter a number n:5
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

6. Function and Library

Common Libraries

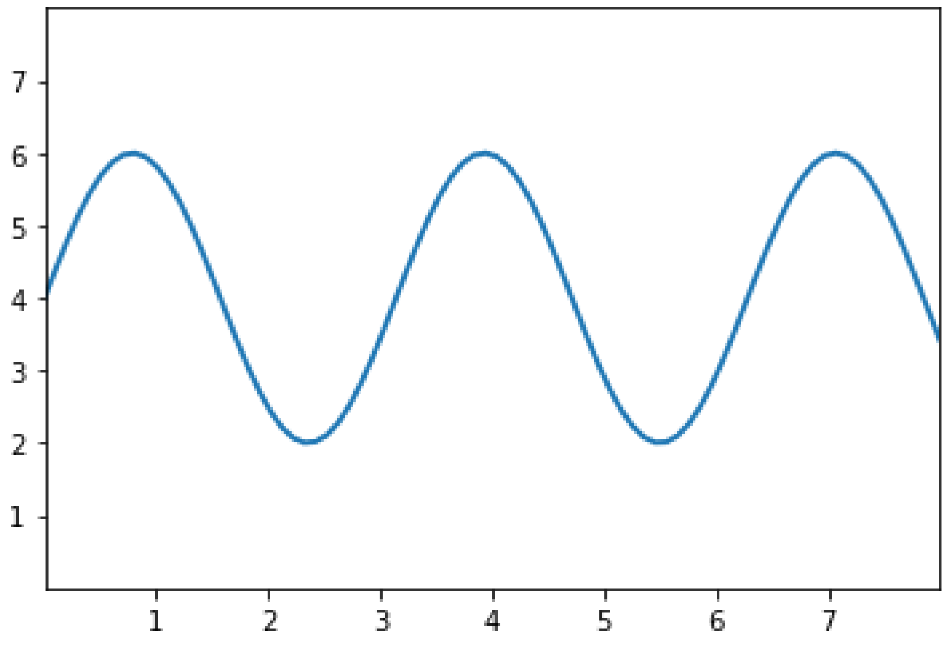




```
import math

x1 = math.ceil(1.4)
x2 = math.floor(1.4)
x3 = math.sqrt(64)
x4 = math.sin(0)
x5 = math.cos(0)

print(x1)
print(x2)
print(x3)
print(x4)
print(x5)
```

```
2
1
8.0
0.0
1.0
```

```
import matplotlib.pyplot as plt
import numpy as np

# make data
x = np.linspace(0, 10, 100)
y = 4 + 2 * np.sin(2 * x)

# plot
fig, ax = plt.subplots()

ax.plot(x, y, linewidth=2.0)

ax.set(xlim=(0, 8), xticks=np.arange(1, 8),
       ylim=(0, 8), yticks=np.arange(1, 8))

plt.show()
```



Customized Function

```
def printString( strPrint, n):
    for i in range (0, n):
        print (strPrint)

printString ("Hello World", 5)
```

Output:

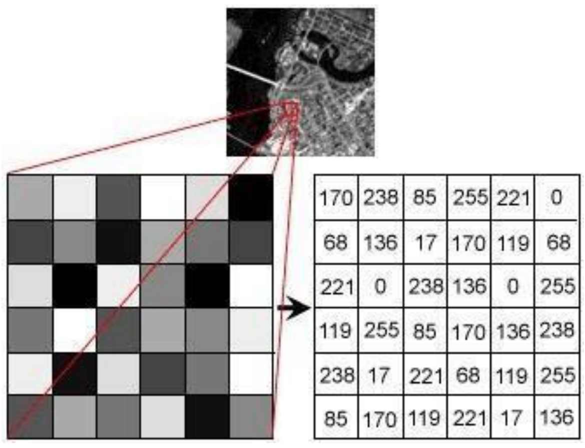
```
Hello World
Hello World
Hello World
Hello World
Hello World
```

```
def addition(no1, no2):
    answer = no1 + no2
    return answer
a = float(input("a="))
b = float(input("b="))
print (a, "+", b, "= %.2f" % addition(a, b))
```

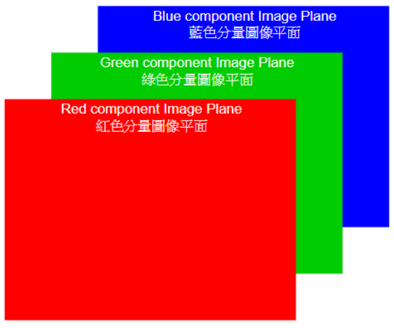
Output:

```
a=1
b=2
1.0 + 2.0 = 3.00
```

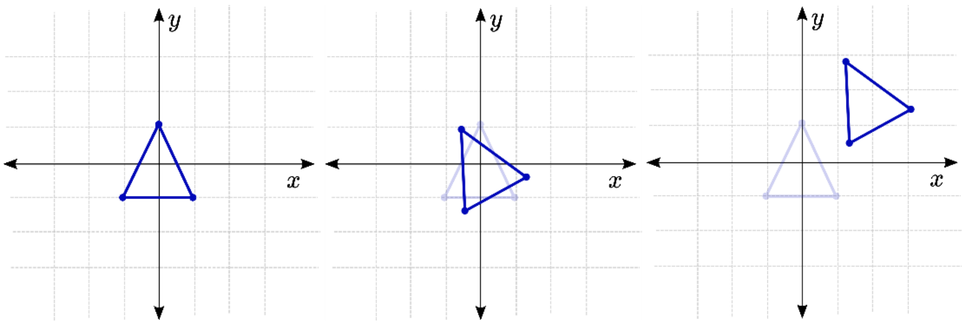
7. Image and computer vision



170	238	85	255	221	0
68	136	17	170	119	68
221	0	238	136	0	255
119	255	85	170	136	238
238	17	221	68	119	255
85	170	119	221	17	136


Blue component Image Plane
藍色分量圖像平面
Green component Image Plane
綠色分量圖像平面
Red component Image Plane
紅色分量圖像平面

RGB Image is a three layered image
RGB圖像是個三層圖像



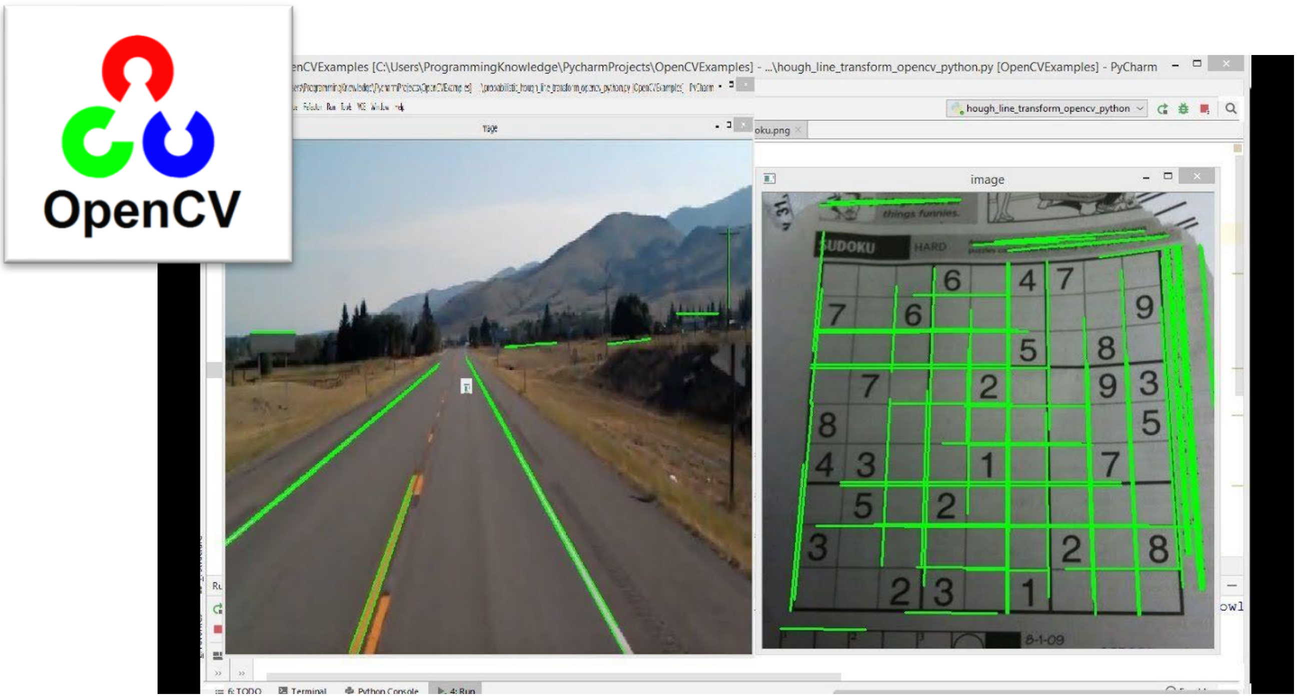
$$\underbrace{\begin{bmatrix}x'\\y'\end{bmatrix}}_{\text{Transformed points}} = A\underbrace{\begin{bmatrix}x\\y\end{bmatrix}}_{\text{Input points}} + B$$

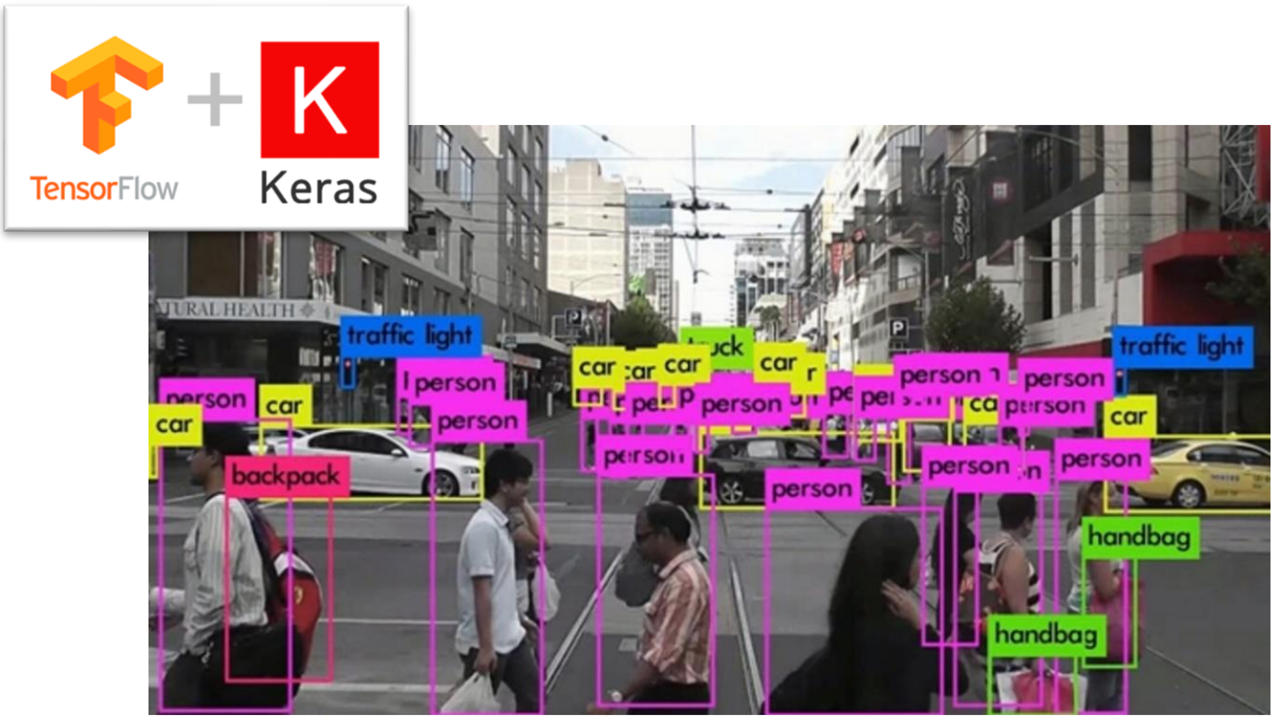
Here, $A = \begin{bmatrix}a_{00} & a_{01}\\ a_{10} & a_{11}\end{bmatrix}$; $B = \begin{bmatrix}b_{00}\\ b_{10}\end{bmatrix}$

Combining A and B we can write,

$$\begin{bmatrix}x'\\y'\end{bmatrix} = \underbrace{\begin{bmatrix}a_{00} & a_{01} & b_{00}\\ a_{10} & a_{11} & b_{10}\end{bmatrix}}_{\text{Transformation Matrix (M)}}\begin{bmatrix}x\\y\\1\end{bmatrix}$$

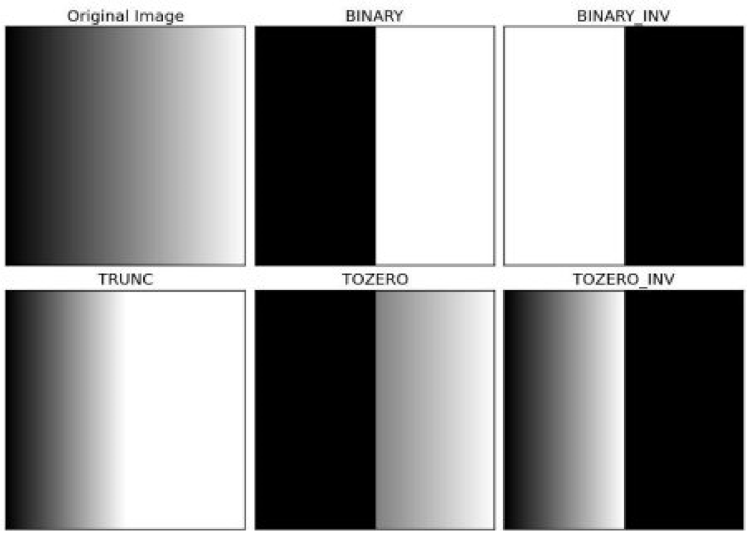
$$\begin{bmatrix}x'\\y'\end{bmatrix} = \begin{bmatrix}a_{00}x + a_{01}y + b_{00}\\ a_{10}x + a_{11}y + b_{10}\end{bmatrix}$$


OpenCV
OpenCVExamples [C:\Users\ProgrammingKnowledge\PycharmProjects\OpenCVExamples] - ...\hough_line_transform_opencv_python.py [OpenCVExamples] - PyCharm
hough_line_transform_opencv_python
image
SUDOKU
HARD
6: TODO
Terminal
Python Console
4: Run


TensorFlow
Keras
traffic light
person
car
truck
backpack
handbag

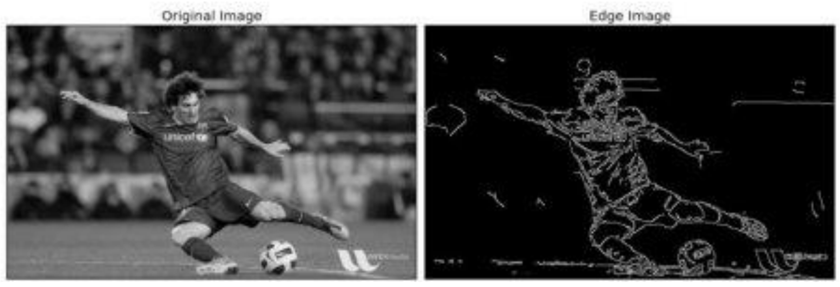
8. Feature extraction

Binary image



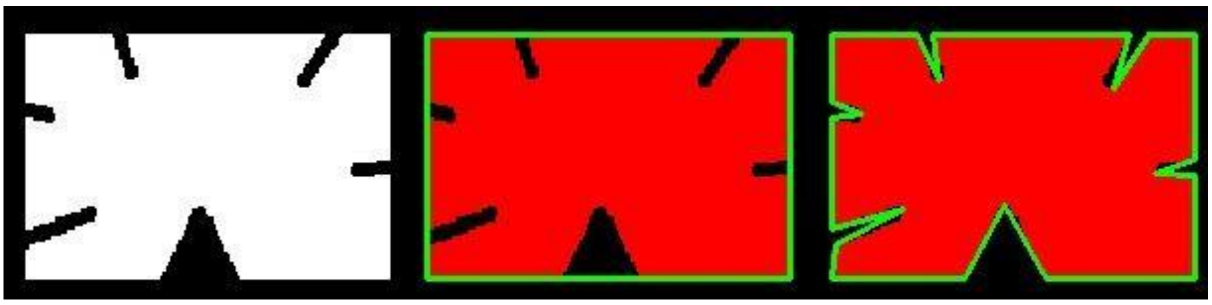
https://docs.opencv.org/3.4/d7/d4d/tutorial_py_thresholding.html

Canny Edge detection



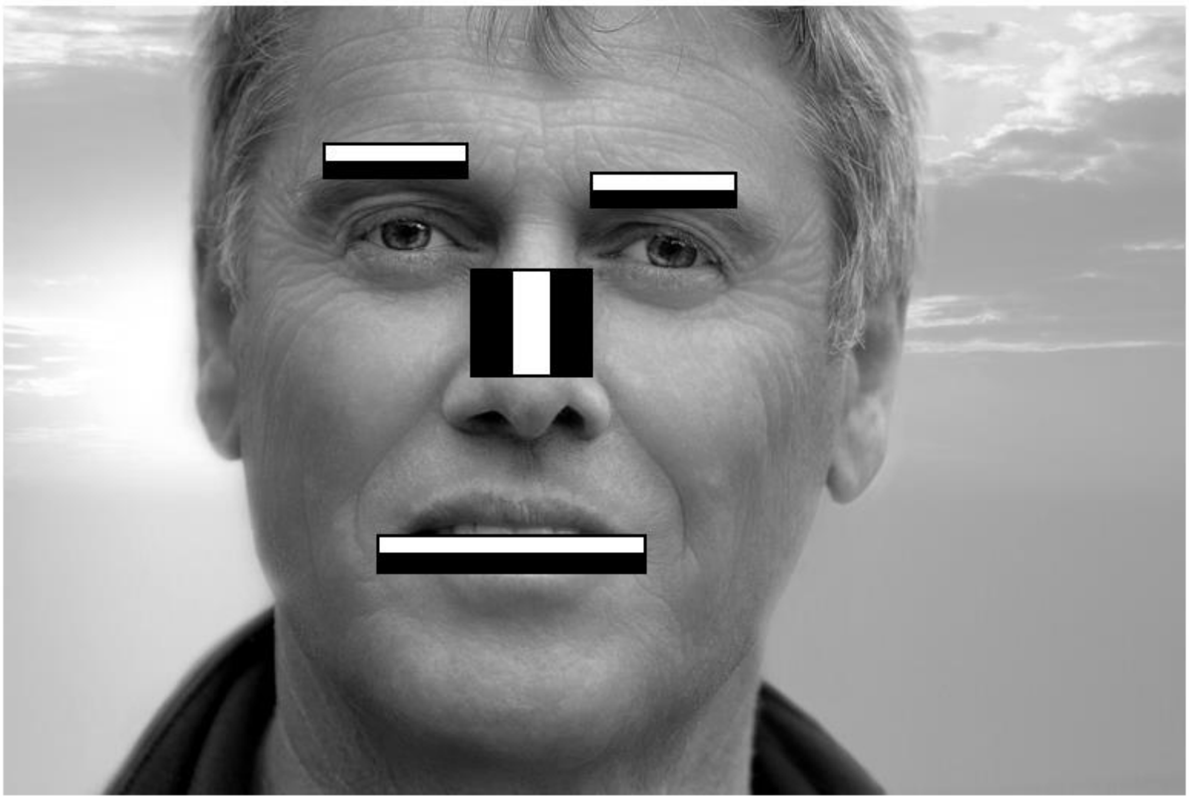
https://docs.opencv.org/3.4/da/d22/tutorial_py_canny.html

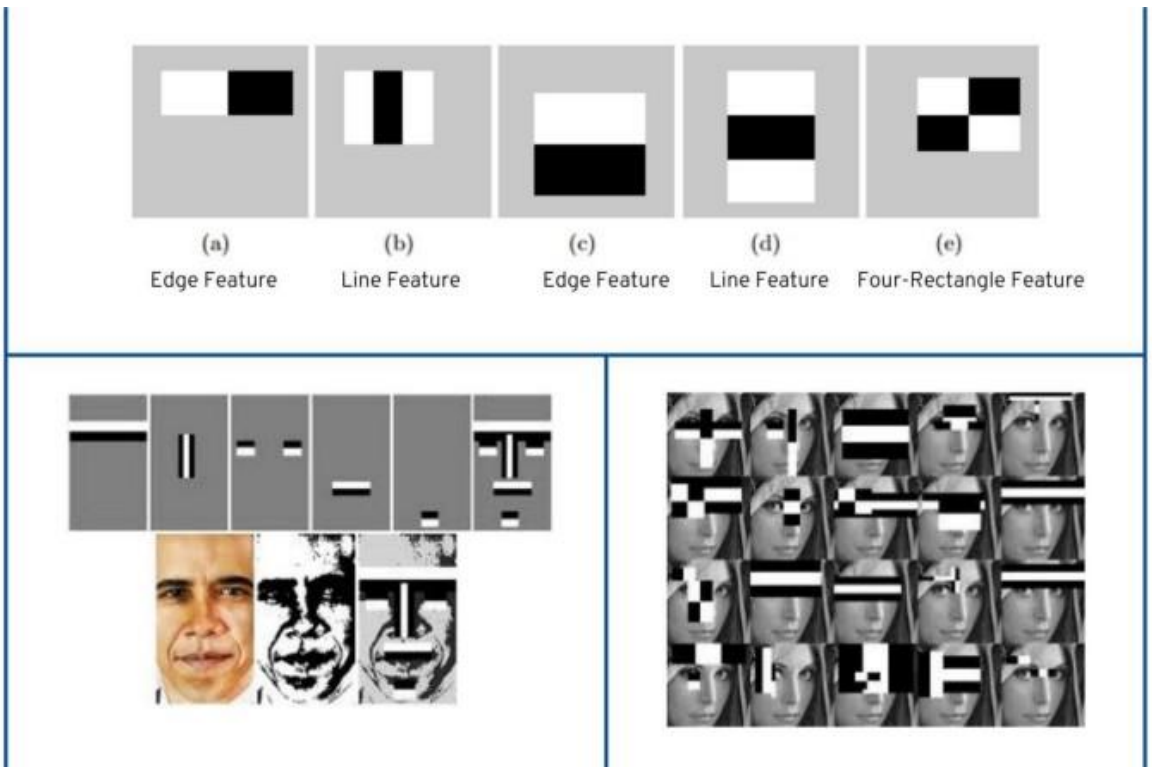
Contour



https://docs.opencv.org/4.x/dd/d49/tutorial_py_contour_features.html

Haar-Like Features



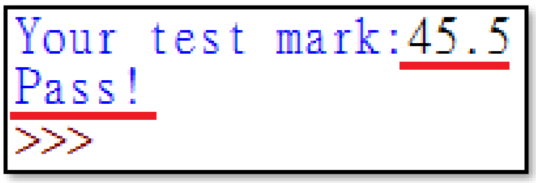


https://www.youtube.com/watch?v=RPoUdDGonWc

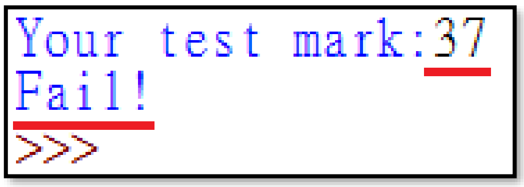
Exercise 0 – Warm up

File: basicPython.ipynb

Task1:

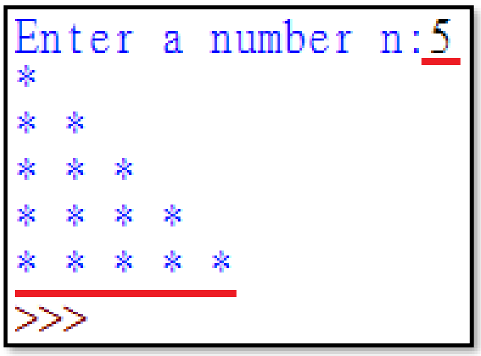


```
Your test mark:45.5
Pass!
>>>
```

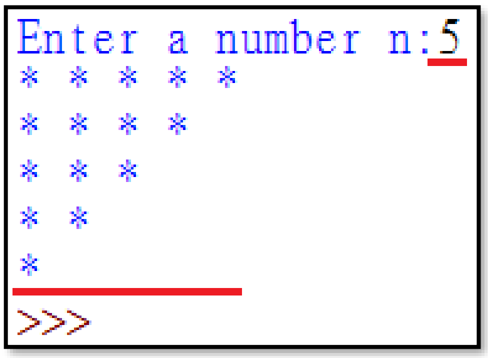


```
Your test mark:37
Fail!
>>>
```

Task2:

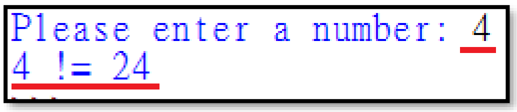


```
Enter a number n:5
*
* *
* * *
* * * *
* * * * *
>>>
```

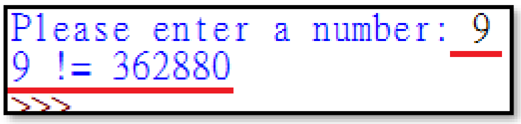


```
Enter a number n:5
* * * * *
* * * *
* * *
* *
*
>>>
```

Task3:



```
Please enter a number: 4
4 != 24
```

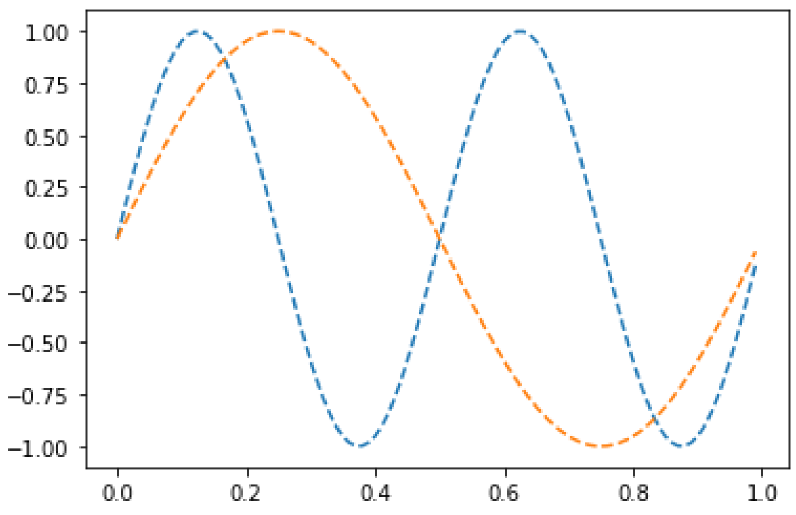


```
Please enter a number: 9
9 != 362880
>>>
```

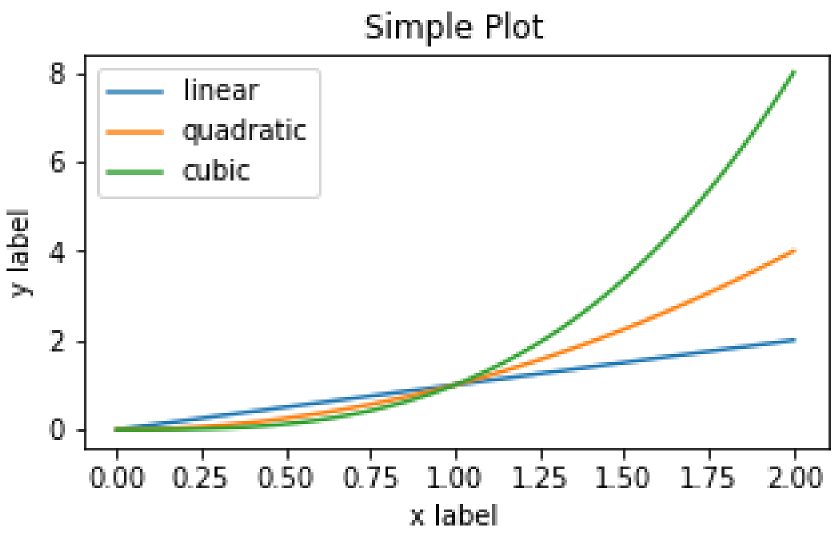
Exercise 1 – Data visualization

File: dataVisualization.ipynb

Task1:

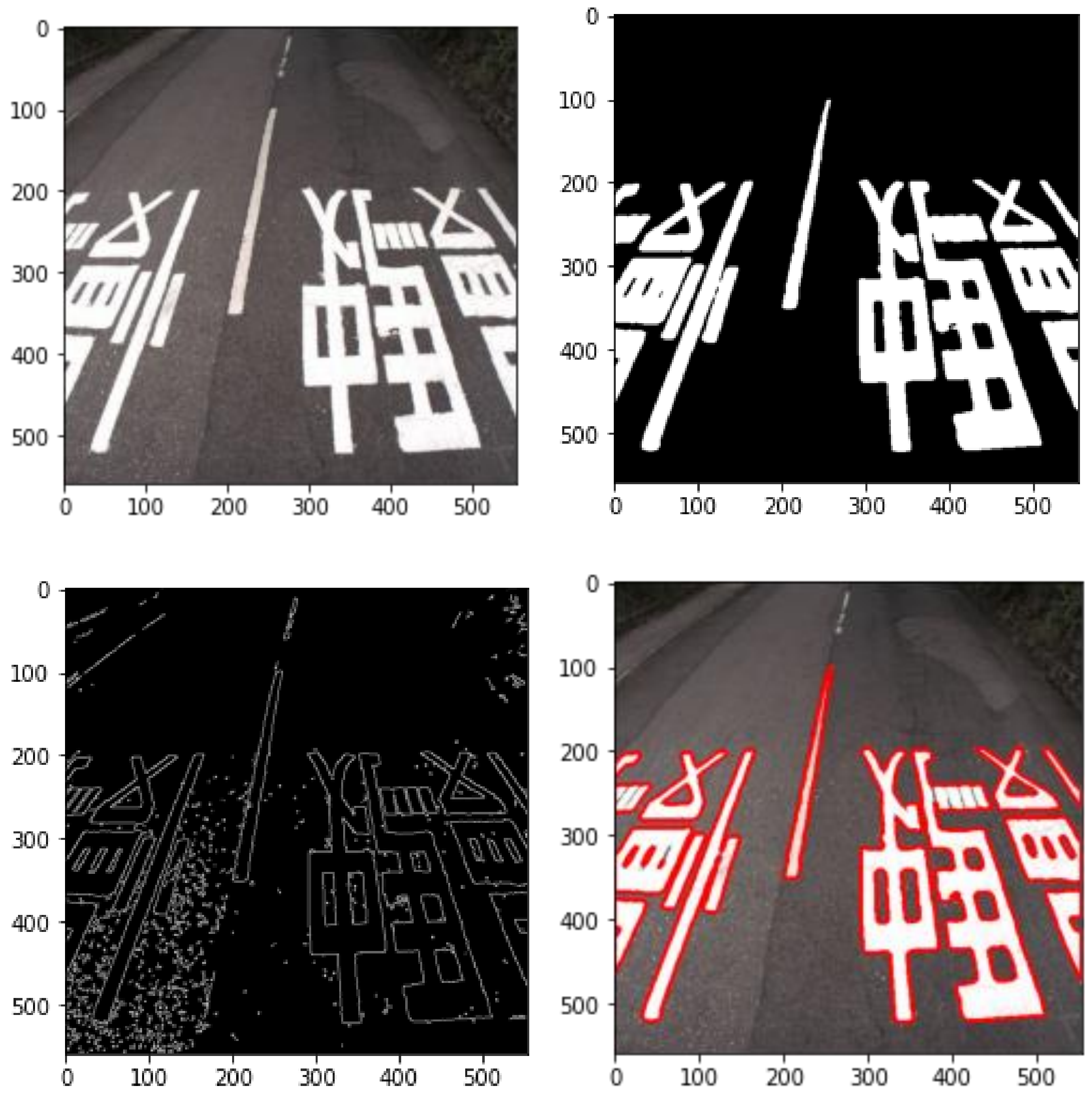


Task2:



Exercise 2a – Features extraction

File: openCV.ipynb



Exercise 2b – Face detection

File: openCV.ipynb