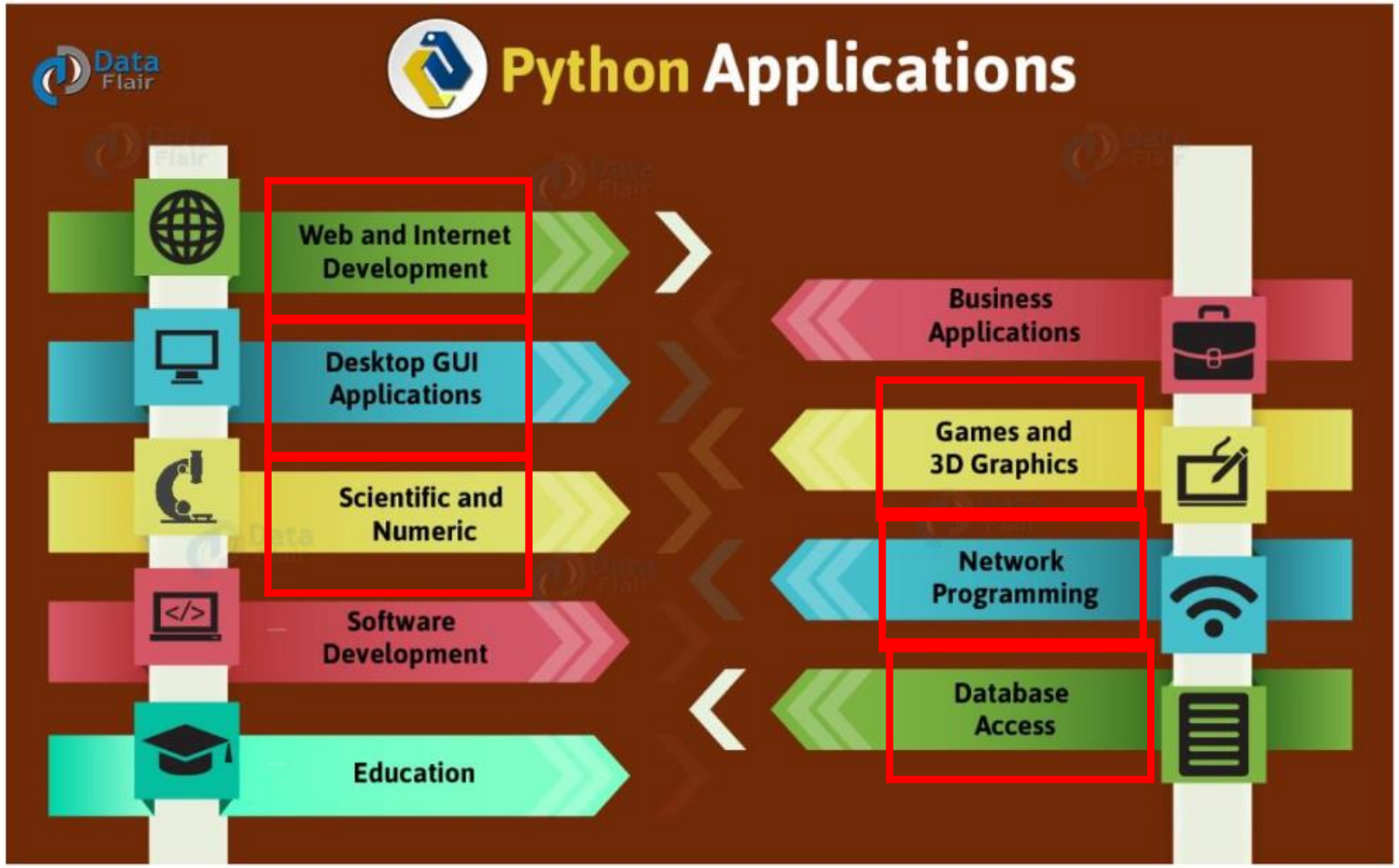
Hexaton Class : Package & Framework

Presented by Hyuk Jun Yoo 2021-01-09

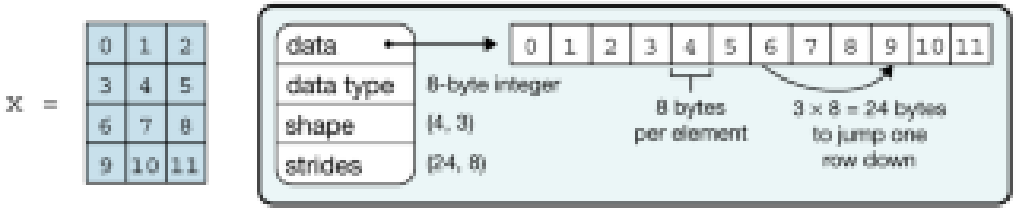
Field


Data Flair
Python Applications
Web and Internet Development
Desktop GUI Applications
Scientific and Numeric
Software Development
Education
Business Applications
Games and 3D Graphics
Network Programming
Database Access

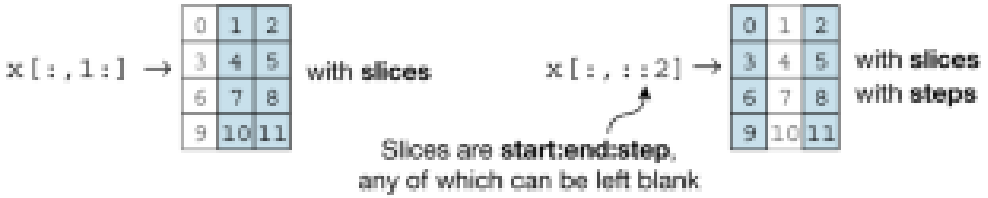
Data Science : Math

Data Science : Numpy

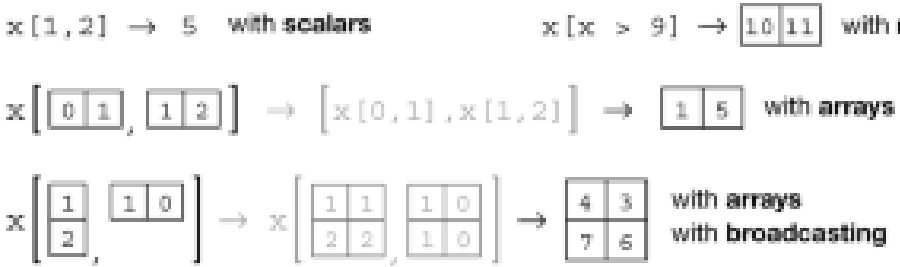
a Data structure



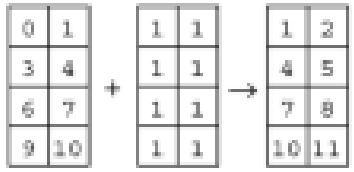
b Indexing (view)



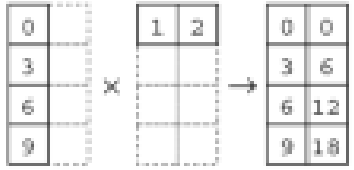
c Indexing (copy)



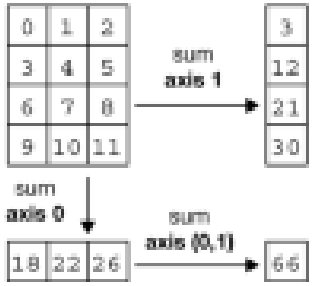
d Vectorization



e Broadcasting



f Reduction



g Example

```
In [1]: import numpy as np

In [2]: x = np.arange(12)

In [3]: x = x.reshape(4, 3)

In [4]: x
Out[4]:
array([[ 0,  1,  2],
       [ 3,  4,  5],
       [ 6,  7,  8],
       [ 9, 10, 11]])

In [5]: np.mean(x, axis=0)
Out[5]: array([4.5, 5.5, 6.5])

In [6]: x = x - np.mean(x, axis=0)

In [7]: x
Out[7]:
array([[-4.5, -4.5, -4.5],
       [-1.5, -1.5, -1.5],
       [ 1.5,  1.5,  1.5],
       [ 4.5,  4.5,  4.5]])
```

Data Science : Pandas

```
1  anime.groupby(["type"]).agg({
2    "rating": "sum",
3    "episodes": "count",
4    "name": "last"
5  }).reset_index()
```

	type	rating	episodes	name
0	Movie	14512.58	2348	Yasuji no Pornorama: Yacchimae!!
1	Music	2727.43	488	Yuu no Mahou
2	ONA	3679.43	659	Docchi mo Maid
3	OVA	20942.60	3311	Violence Gekiga Shin David no Hoshi: Inma Dens...
4	Special	10900.77	1676	Junjou Shoujo Et Cetera Specials
5	TV	25338.34	3787	Yuuki Yuuna wa Yuusha de Aru: Yuusha no Shou

Data Science : Scipy, SymPy

In [26]:
```
_RIGHT = _1T2 * _2T3
_RIGHT.simplify()
_RIGHT
```

Out[26]:

$$\begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & l_1 + l_2\cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & l_2\sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In [40]:
```
eq1 = _RIGHT.row(0).col(3)
eq1
```

Out[40]: $\left[l_1 + l_2\cos(\theta_2) \right]$

In [41]:
```
type(eq1)
```

Out[41]: sympy.matrices.dense.MutableDenseMatrix

In [48]:
```
eq1 = l1+l2*sp.cos(q2)
eq1
```

Out[48]: $l_1 + l_2\cos(\theta_2)$

In [5]:
```
from sympy import *
x,y,z = symbols('x y z')
init_printing()
```

In [10]:
```
Derivative(sqrt(1/x**2),x)
```

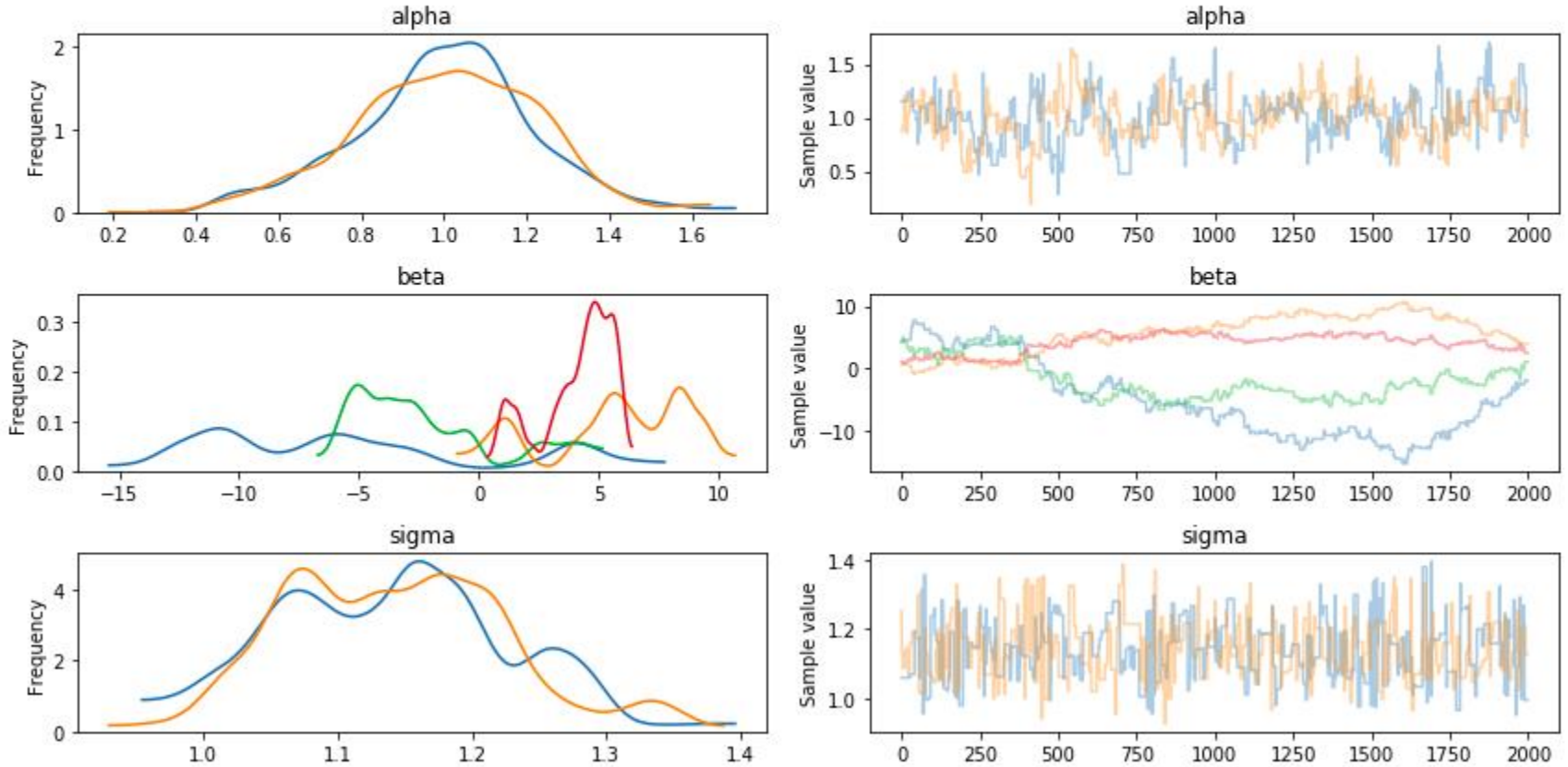
Out[10]:

$$\frac{d}{dx}\sqrt{\frac{1}{x^2}}$$

In [ ]:

Data Science : PyMC3

```
## Trace Plot of the Metropolis-Hastings Sampler
from pymc3 import traceplot

traceplot(trace_MH)
plt.show()
```

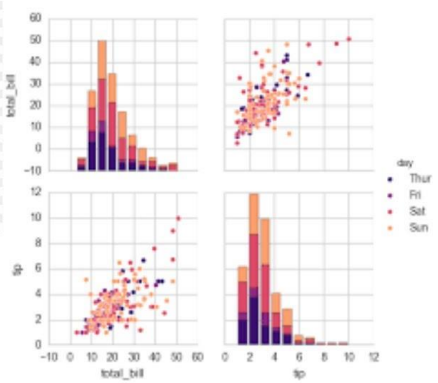


Data Science : Data Visualization

matplotlib

Seaborn

SEABORN
PAIRPLOT



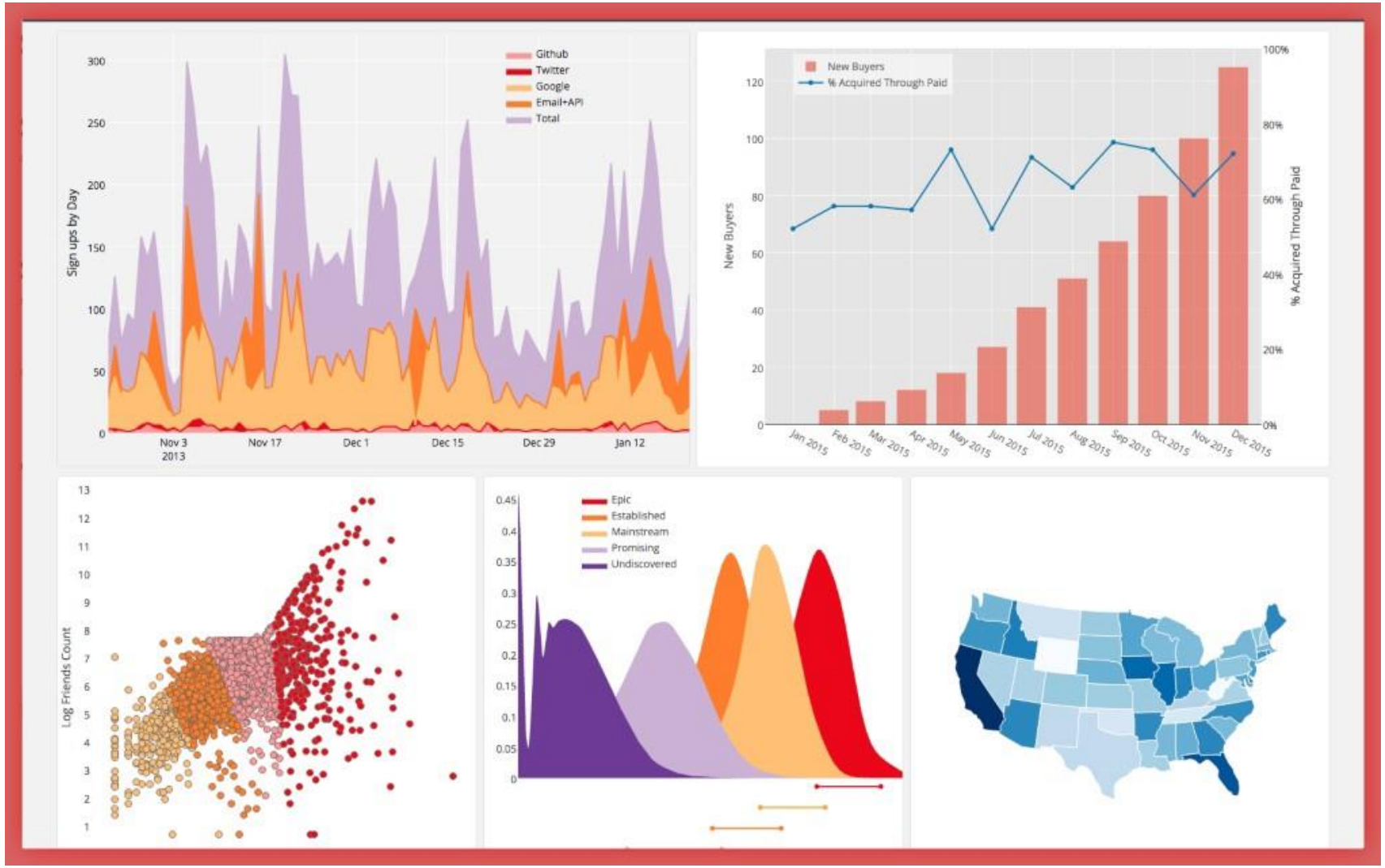
DATA VISUALIZATION
PYTHON

kibana

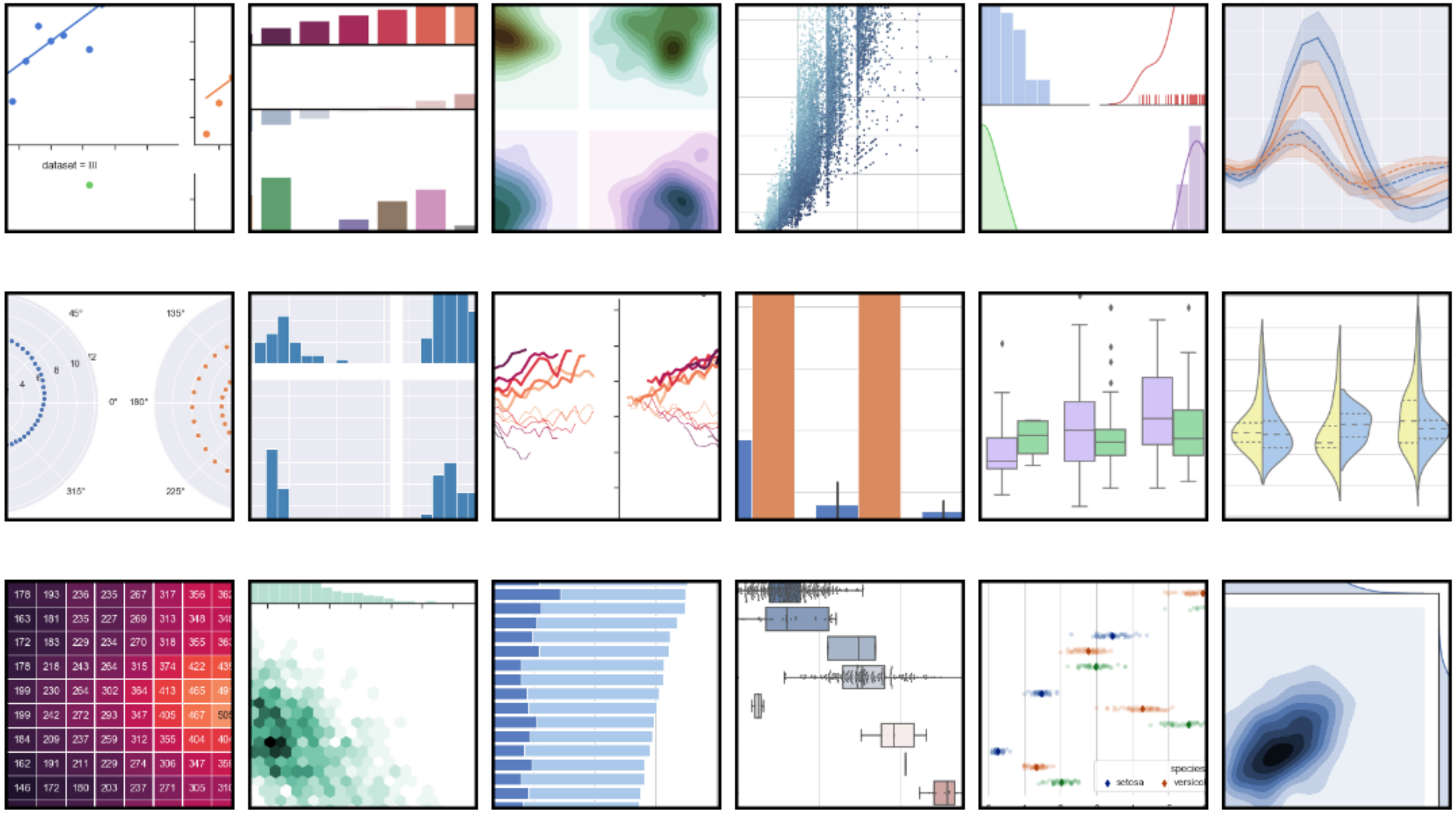
elasticsearch.

NetworkX
Network Analysis in Python

Data Science : Matplotlib

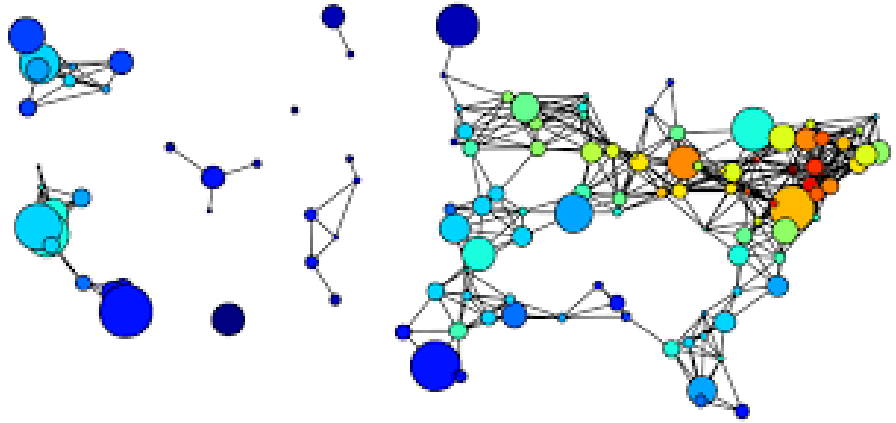


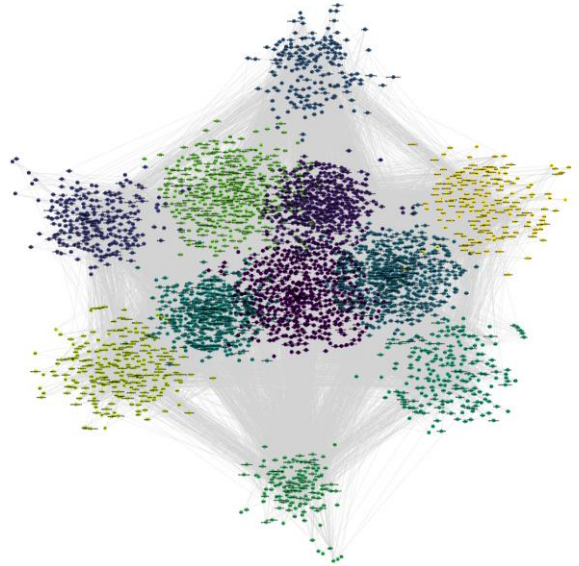
Data Science : Seaborn

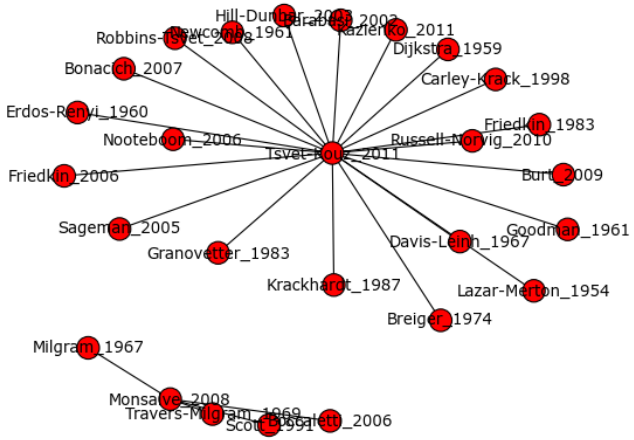


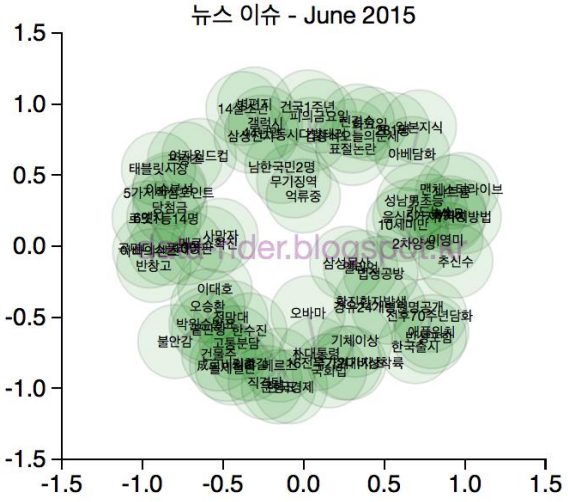
Data Science : Kibana & ElasticSearch

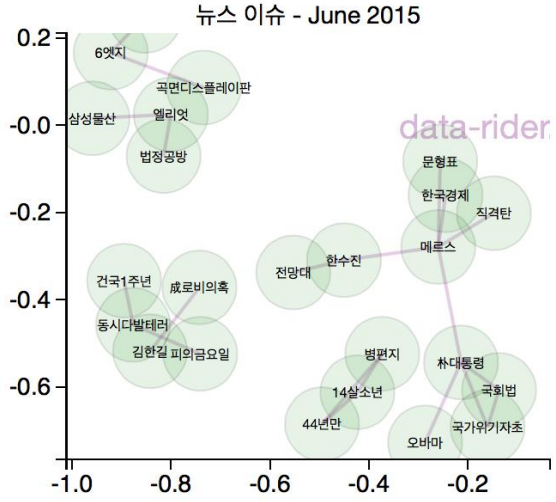
Data Science : NetworkX







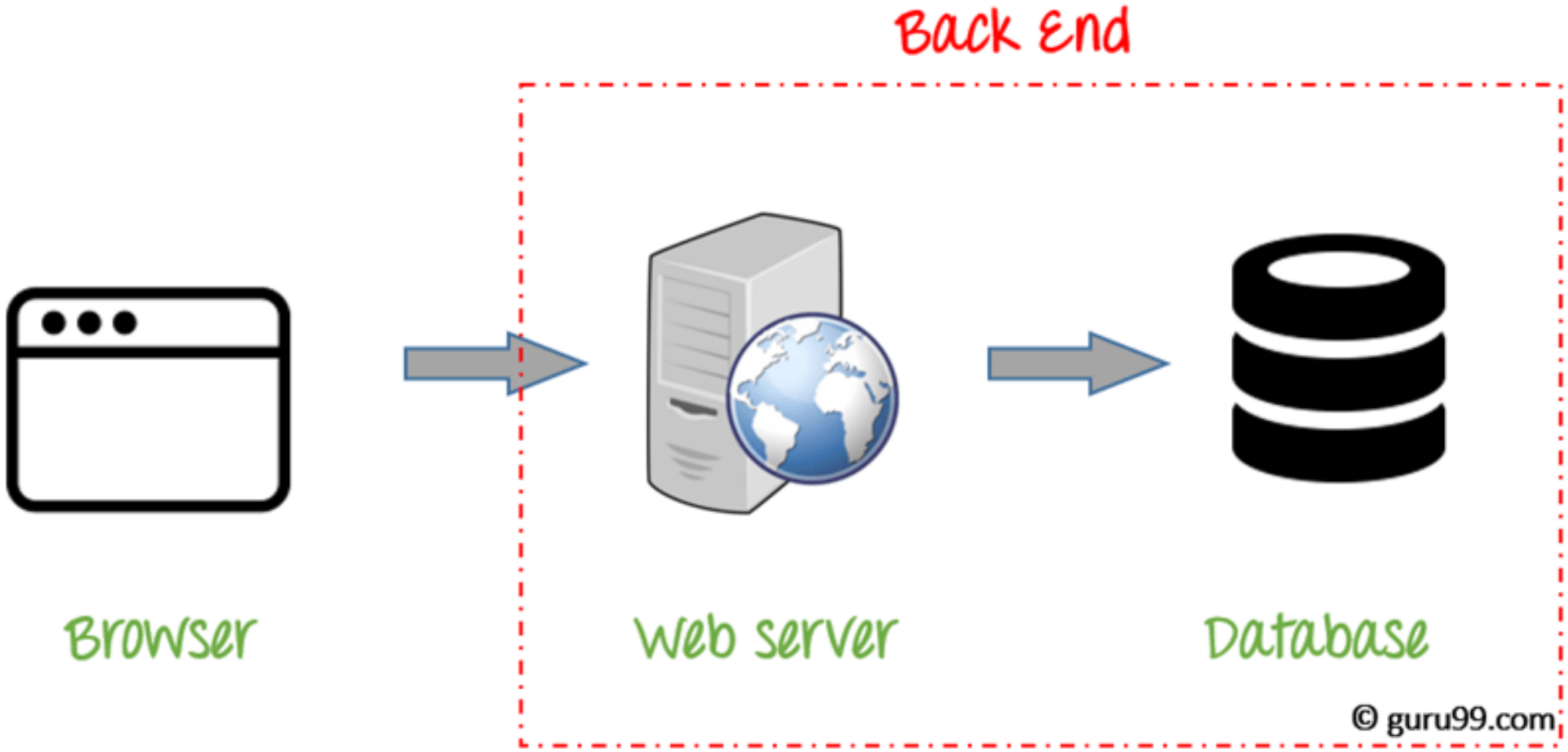




Data Science : Machine Learning

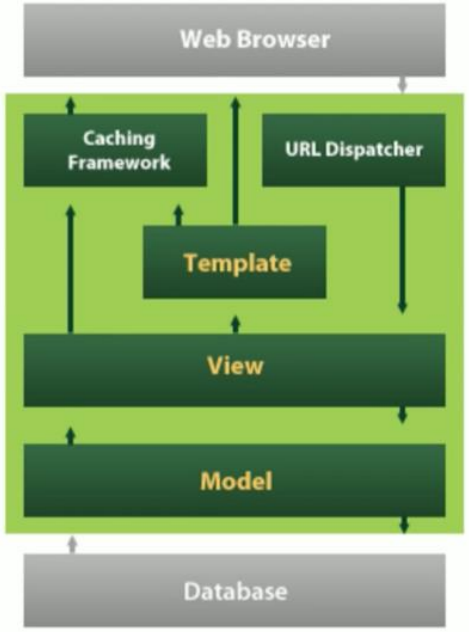
Web Programming

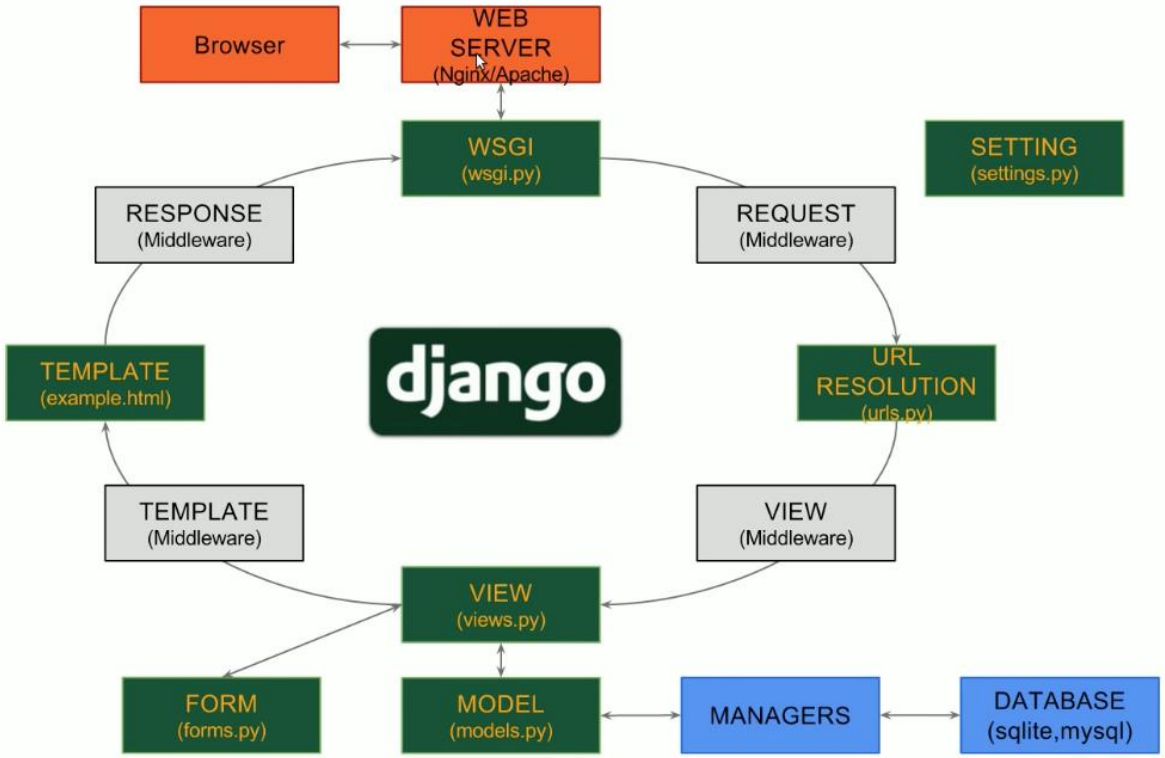
Web Programming : Frontend , Backend



Web Programming : Django

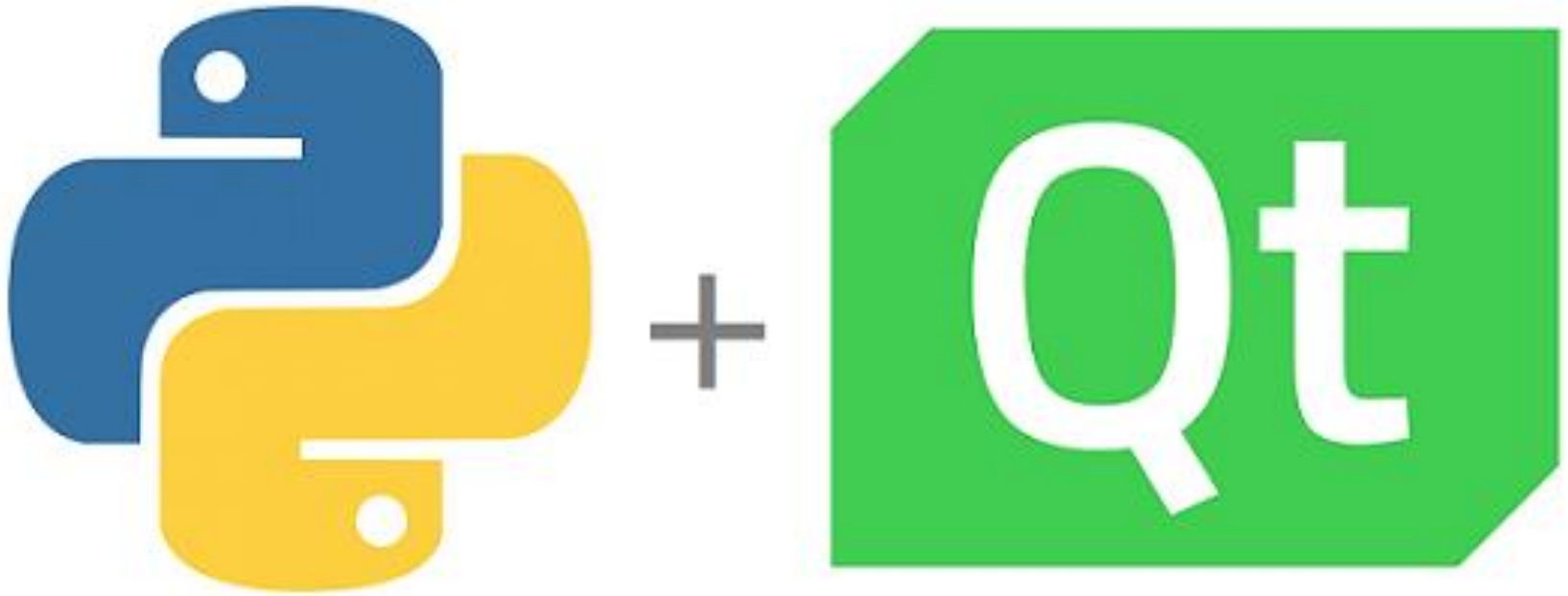
Django 개념





출처: https://mytardis.readthedocs.org/en/latest/architecture.html

GUI : PyQT5



GUI : PyQT5

C:₩Anaconda3₩Lib₩site-packages₩PyQt5₩uic₩main_window.py - Notepad++

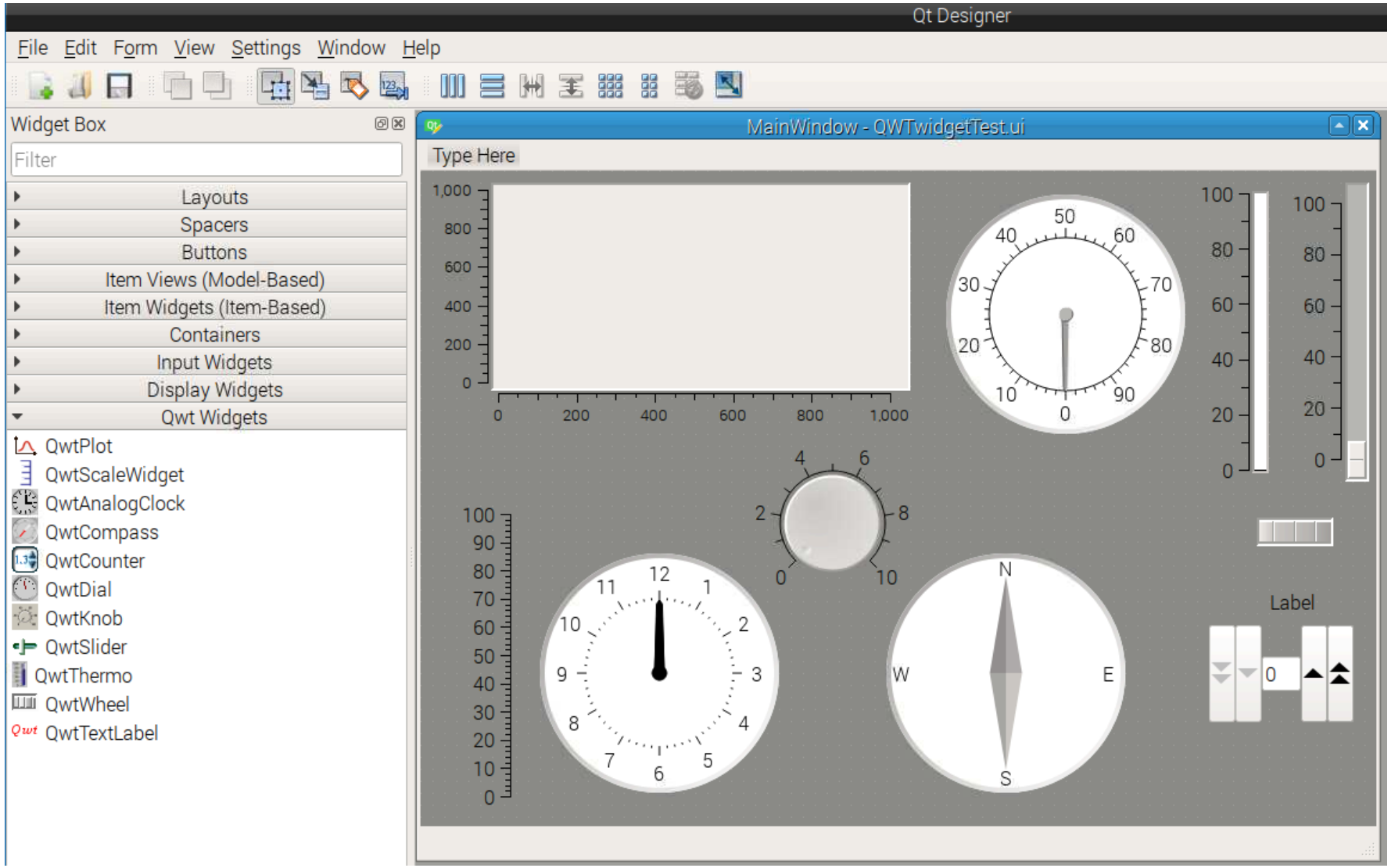
파일(F) 편집(E) 찾기(S) 보기(V) 인코딩(N) 언어(L) 설정(T) 매크로 실행 플러그인 창 관리 ? X

main_window.py

```
# -*- coding: utf-8 -*-

# Form implementation generated from reading ui file 'main_window.ui'
#
# Created by: PyQt5 UI code generator 5.6
#
# WARNING! All changes made in this file will be lost!

from PyQt5 import QtCore, QtGui, QtWidgets

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(361, 159)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.pushButton = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton.setGeometry(QtCore.QRect(280, 20, 75, 23))
        self.pushButton.setObjectName("pushButton")
        self.label = QtWidgets.QLabel(self.centralwidget)
        self.label.setGeometry(QtCore.QRect(30, 30, 56, 12))
        self.label.setText("")
```

length : 1918 lines Ln : 7 Col : 55 Sel : 0 | 0 Dos₩Windows UTF-8 INS

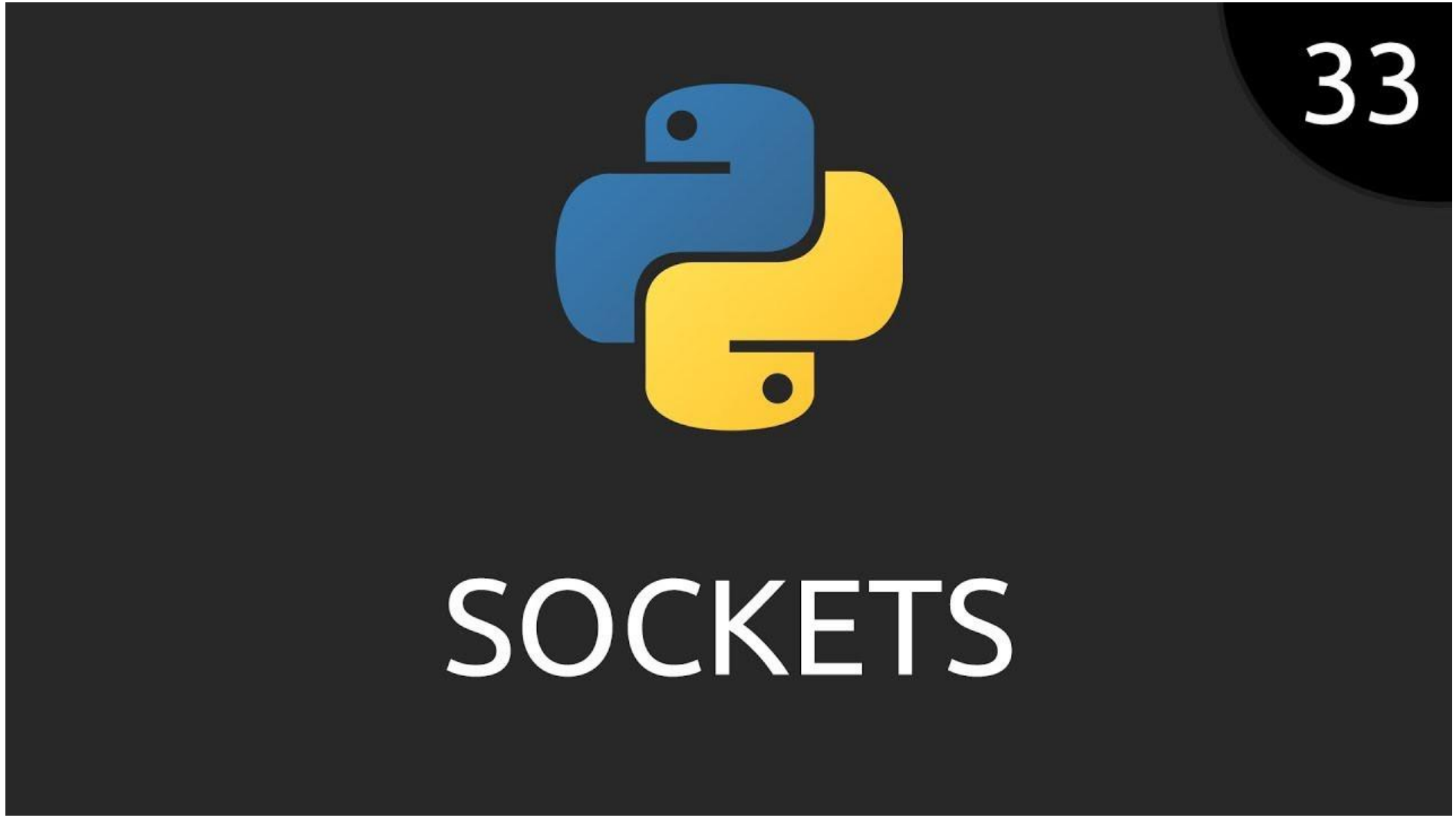
GUI : PyQT5


Qt Designer
File Edit Form View Settings Window Help
Widget Box
Filter
Layouts
Spacers
Buttons
Item Views (Model-Based)
Item Widgets (Item-Based)
Containers
Input Widgets
Display Widgets
Qwt Widgets
QwtPlot
QwtScaleWidget
QwtAnalogClock
QwtCompass
QwtCounter
QwtDial
QwtKnob
QwtSlider
QwtThermo
QwtWheel
QwtTextLabel
MainWindow - QWTwidgetTest.ui
Type Here
Label

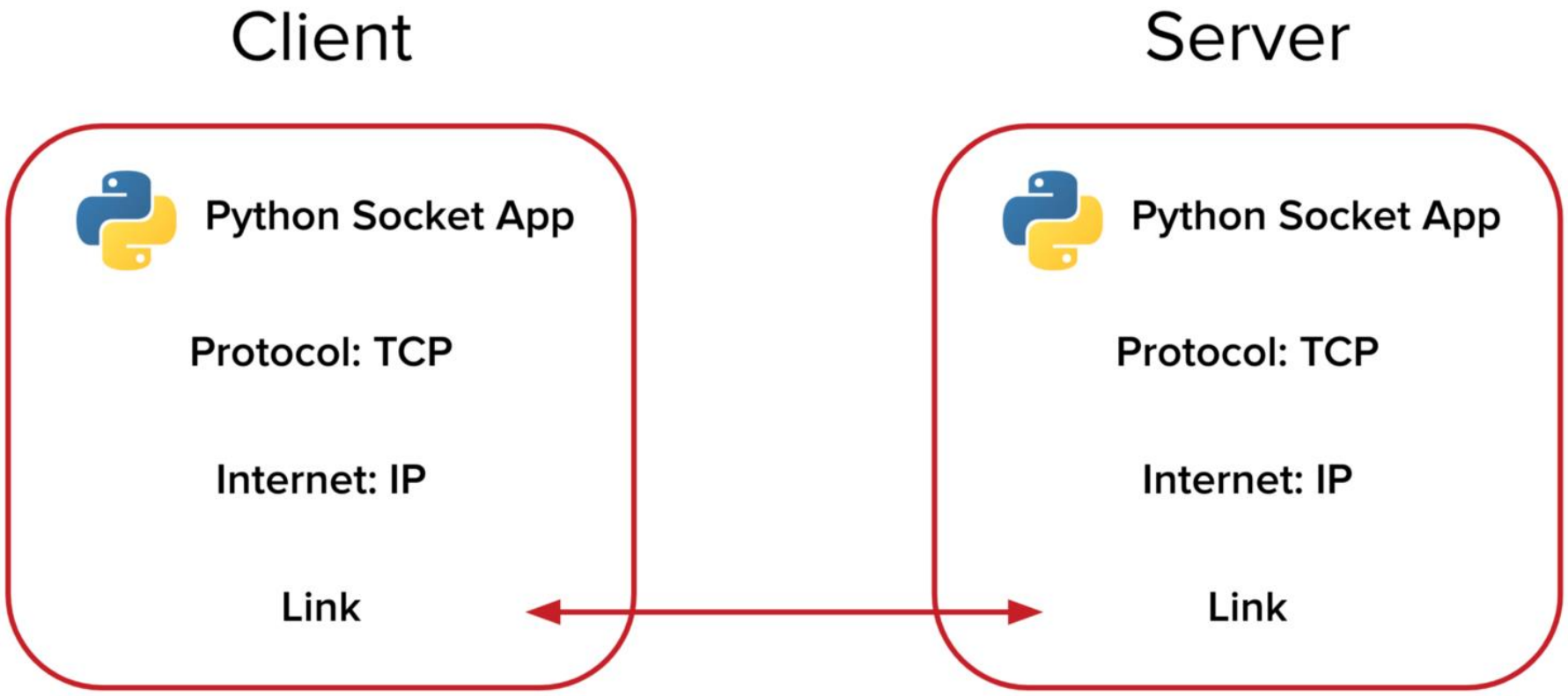
Game : PyGame

Game : PyGame

Network



Network


Client
Server
Python Socket App
Protocol: TCP
Internet: IP
Link
Python Socket App
Protocol: TCP
Internet: IP
Link

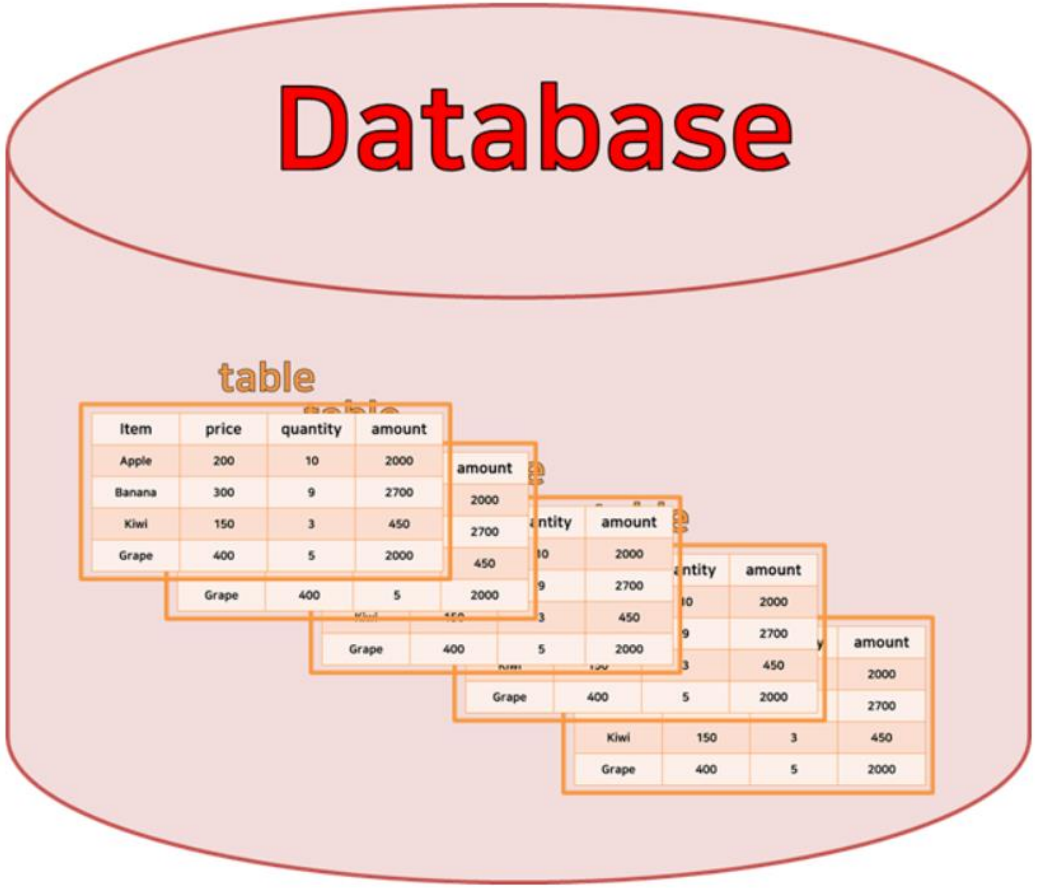
Database

Database : table

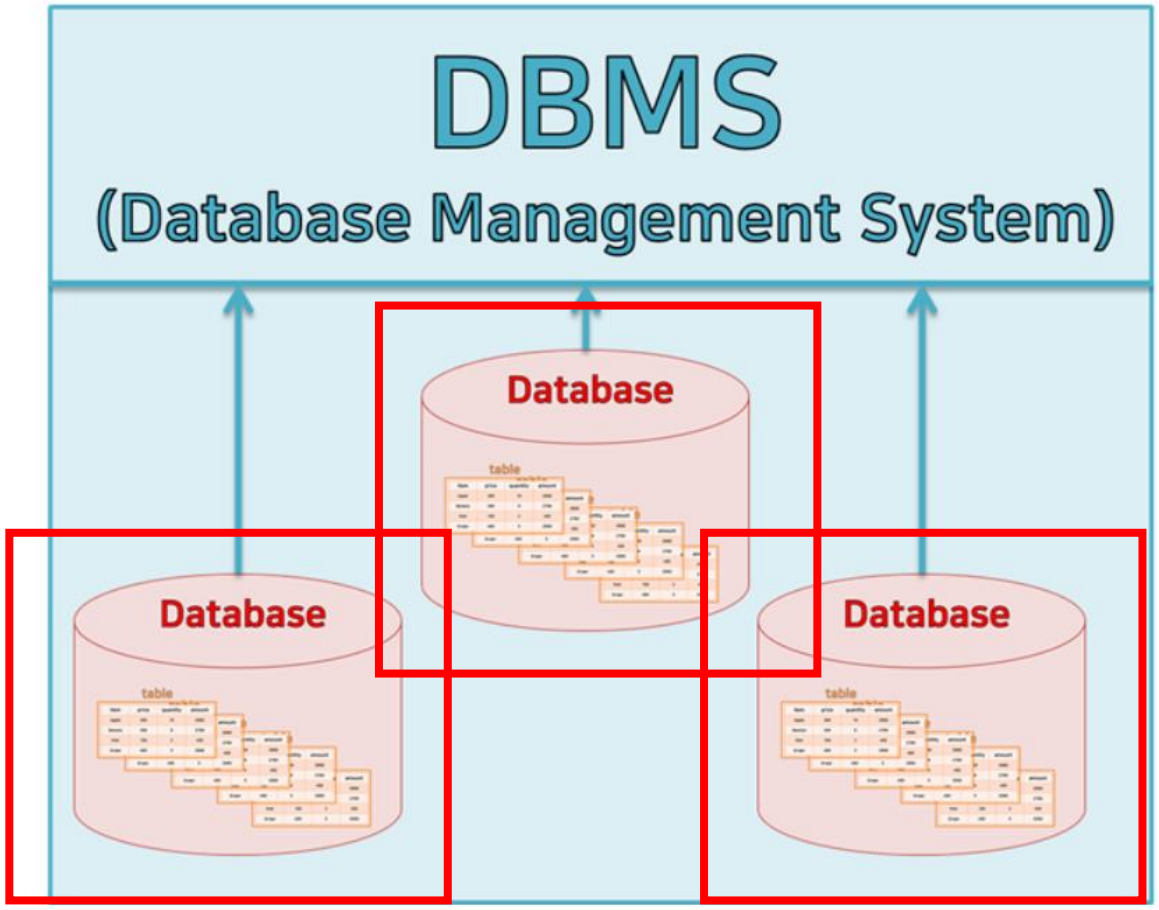
table

Item	price	quantity	amount
Apple	200	10	2000
Banana	300	9	2700
Kiwi	150	3	450
Grape	400	5	2000

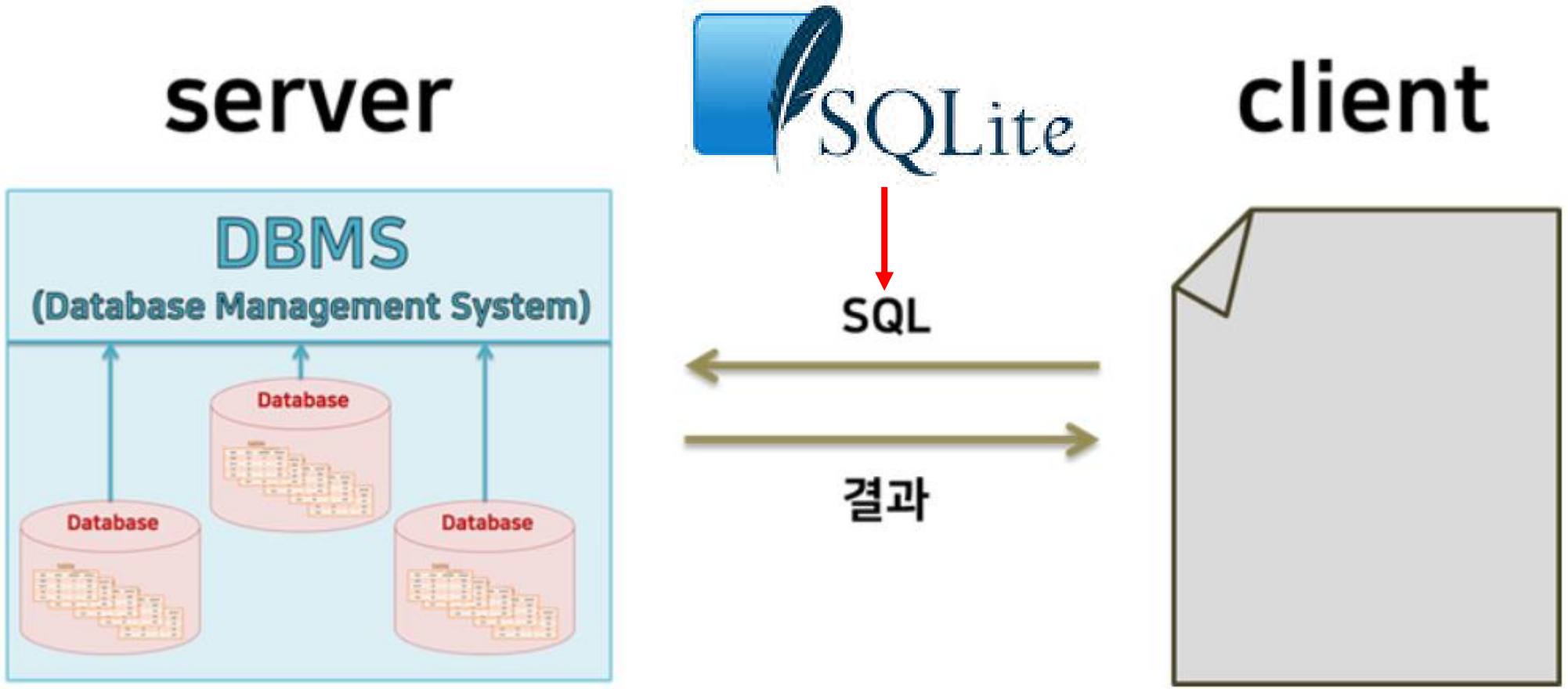
Database : DataBase



Database : DBMS



Database : Architecture



Database : command

```
sqlite> CREATE TABLE person(
   ...> ID INT PRIMARY KEY NOT NULL,
   ...> NAME TEXT NOT NULL,
   ...> AGE INT NOT NULL );
sqlite> .tables
person
```

```
sqlite> SELECT * FROM person;
1|LEE|28
2|CHO|29
3|WANG|24
4|PARL|24
5|CHOI|22
```

```
sqlite> INSERT INTO person VALUES ( 1, 'LEE', 28 );
sqlite> INSERT INTO person VALUES ( 2, 'CHO', 29 );
sqlite> INSERT INTO person VALUES ( 3, 'WANG', 24 );
sqlite> INSERT INTO person VALUES ( 4, 'PARL', 24 );
sqlite> INSERT INTO person VALUES ( 5, 'CHOI', 22 );
```

```
sqlite> DROP TABLE person;
sqlite> .table
sqlite>
```

Database : Collaborate with Python

```
conn1 = sqlite3.connect('../crawling_history/database/History_all_users_title_token.db')

c1 = conn1.cursor()

# 비교할 username의 최근 History를 모든 user들의 history와 비교하기 위해 임시로 attach
c1.execute("ATTACH '" + "../crawling_history/database/" + user_name + "/History' as History_" + user_name + "_temp;")
c1.execute("ATTACH '" + "../crawling_history/database/History_all_users.db' as History_all_users_temp;")

# username의 history가 있다면 이를 제거한 모든 user들의 history data들의 결과
result = []
# token마다 진행
for token in final_token_list:
    c1.execute("CREATE TABLE IF NOT EXISTS extract_urls(title text, url text, user_count INTEGER, visit_count INTEGER)")
    c1.execute("SELECT title, url, user_count, visit_count from History_all_users_temp.urls WHERE url in (SELECT url from sorted_urls where title_token = '"+token+"' EXC
    result.extend(c1.fetchall())

# 결과를 extract_urls table에 저장
c1.executemany("INSERT INTO extract_urls(title, url, user_count ,visit_count) VALUES (?,?,?,?)", result)
# 우선 token이 두 개 이상 겹치는 단어를 보여준다.
c1.execute("select title, url , user_count, visit_count FROM extract_urls GROUP BY url having count(url) = " + str(len(final_token_list)) + " ORDER BY user_count asc")
# c1.execute("select * from extract_urls order by url")
# token이 모두 겹친 것을 보여준다.
result_token_all = c1.fetchall()
```

Database : SQLite GUI