

Supercomputer, Stock Market data & MySQL

- ✓ How to use Supercomputer..?
- ✓ About Stock Market data
- ✓ Connecting MySQL @ Python



What we did last week is..



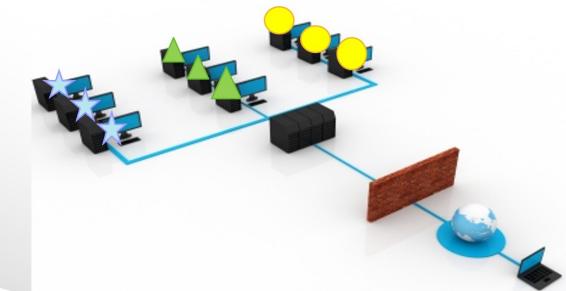
TOP500

- Computers in the **TOP500** (ranked 500)
 - ISC (International Supercomputing Conference)
 - SC (Supercomputing Conference)
- June 2021 ranking
 - 1st : Japan (**Fugaku** (442 Pflop/s), Fujitsu RIKEN Center for Computational Science)
 - 2nd : U.S. (**Summit** (148.8 Pflop/s), Oak Ridge National Laboratory)
 - 3rd : U.S. (**Sierra** (94.6 Pflop/s), Lawrence Livermore National Laboratory)
 - 4th : China (**TaihuLight** (93.0 Pflop/s), National Supercomputing Center in Wuxi)

Supercomputer manufacturing process



Why is parallel computing so fast?



What is Metaverse...?

- Metavers = Meta + Universe
- The metaverse is derived from the 1992 novel '**Snow Crash**' by Neal Stephenson.
 - The characters in the novel must borrow a virtual body called 'Avatar' to enter the 'Metaverse', a virtual world.
- In the virtual reality called metaverse, there is no time or physical limitation.



Major Metaverse Platforms: ZEPETO

- ZEPETO is a 3D avatar creation application launched by SNOW, a subsidiary of NAVER.
- Global fashion brands such as **Gucci**, **Dior**, and **Nike** create virtual booths to promote their brands and sell paid items.



CONTENTS



- 01 How to use Supercomputer..?**
- 02 About Stock Market data**
- 03 How to connect MySQL @ Python..?**
- 04 Make your First Database using COVID-19 data**



01

How to use Supercomputer..?



Nurion @ KISTI



Main System



Storage

Create an account @ Nurion



- Researchers who have been approved to use the Nurion system apply for an account through the [KISTI website \(\)](https://www.ksc.re.kr) web service.
 - [Free account](#): Nurion system innovation support program, novice user of Unit 5
 - [Paid account](#): Unit 5 general user, Unit 5 student user

『초보사용자』

대상 : 슈퍼컴퓨터 시스템을 처음 사용하는 사용자

- 사용료 : 무료
- 할당시간 : 10 SRU시간(일반사용자의 1/10)
- 사용기간 : 3 개월
- 연장여부 : 불가

※ 최초 할당 받은 SRU 시간 초과 사용 또는 사용 기간 만료시 까지 사용 가능

※ 코드 테스트를 위한 계정입니다. 연구 목적용 계산을 할 수 없습니다.

※ 정식 계약시 새로운 ID(초보사용자 ID는 사용 불가)로만 사용 가능합니다.

[신청하기](#)

『학생사용자』

대상 : 학교에 재학 중인 학생(학부/석사/박사과정)

- 사용료 : 100,000원/구좌당
 - 할당시간 : 10 SRU 시간 (일반사용자의 1/10)
 - 사용기간 : 1 년
 - 연장여부 : 가능(사용료 추가 납부)
- ※ 구입한 SRU 시간 초과 사용 또는 사용 기간 만료시 까지 사용 가능
※ 신규/연장시 학생증 사본을 E-mail(account@ksc.re.kr)로 전송해야 합니다.

[신청하기](#)

『일반사용자』

대상 : 대학(교수/직원), 국가기관, 연구소, 산업체의 모든 사용자

- 사용료 : 1,000,000원/구좌당
- 할당시간 : 100 SRU 시간(요금 안내 참조)
- 사용기간 : 1년 단위
- 연장여부 : 가능 (사용료 추가 납부)

※ 구입한 SRU 시간 초과 사용 또는 사용 기간 만료시 까지 사용 가능

[신청하기](#)

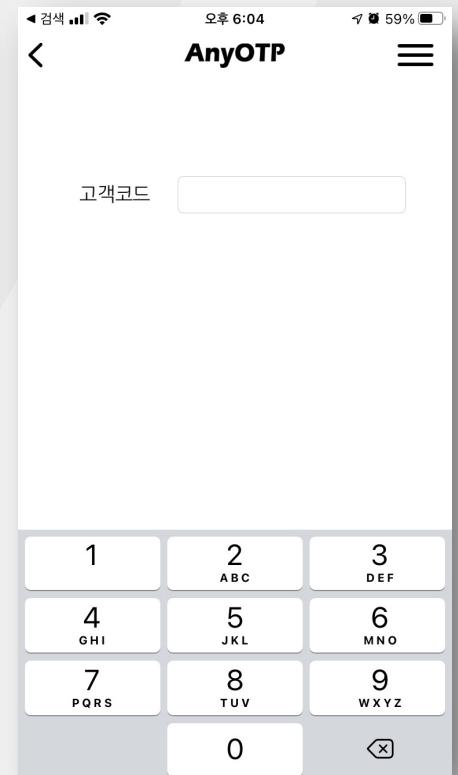
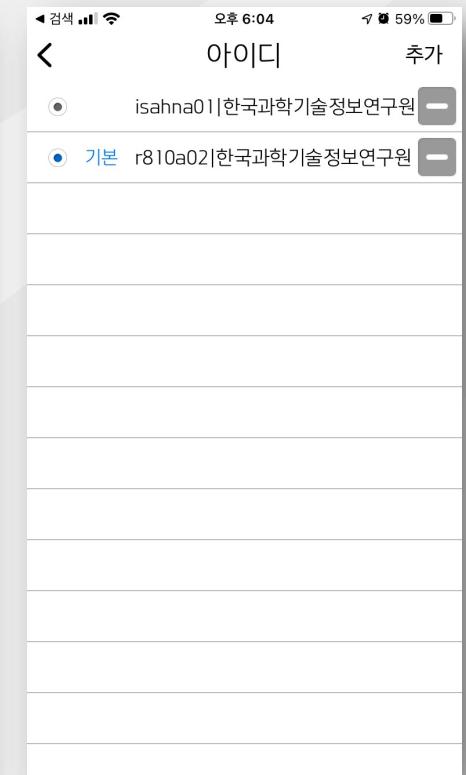
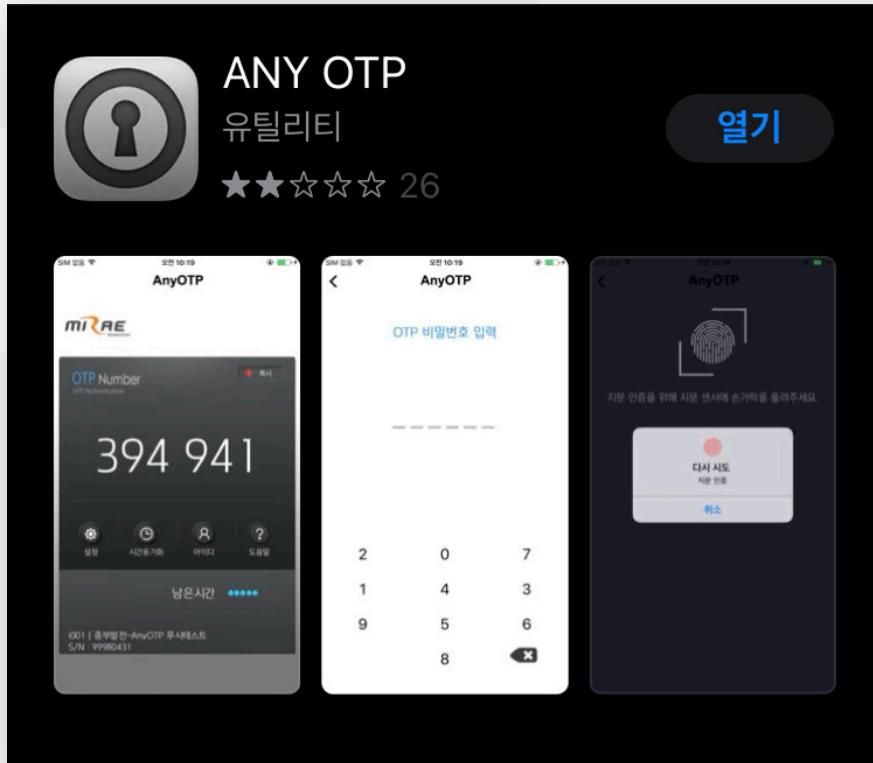
OTP (One Time Password) authentication code

- Referring to the received account information e-mail, fill out the form below and receive an authentication code through account@ksc.re.kr.
 - E-mail title: “OTP 인증코드 발송 요청 – User ID”
 - Address: account@ksc.re.kr
 - Mail content (example)
 - Login ID: x123abc
 - Phone: 010-1234-5678
 - Name: Tom & Jerry
 - Mobile carrier: SKT or KT or LG U+

Install the OTP app in your phone



- Install Smartphone app by searching for “Any OTP” in the Android App Store (Google Play) or the iPhone App Store (App Store)



Login to Nurion

- Nurion system login node : nurion.ksc.re.kr
- Unix or Linux environment: using Terminal

```
ssh -l r810a02 nurion.ksc.re.kr -p 22
```

- Windows environment: using putty

Download PuTTY: latest release (0.76)

[Home](#) | [FAQ](#) | [Feedback](#) | [Licence](#) | [Updates](#) | [Mirrors](#) | [Keys](#) | [Links](#) | [Team](#)
Download: [Stable](#) · [Snapshot](#) | [Docs](#) | [Changes](#) | [Wishlist](#)

This page contains download links for the latest released version of PuTTY. Currently this is 0.76, released on 2021-07-17.

When new releases come out, this page will update to contain the latest, so this is a good page to bookmark or link to. Alternatively, here is a [permanent link to the 0.76 release](#).

Release versions of PuTTY are versions we think are reasonably likely to work well. However, they are often not the most up-to-date version of the code available. If you have a problem with this release, then it might be worth trying out the [development snapshots](#), to see if the problem has already been fixed in those versions.

Package files

You probably want one of these. They include versions of all the PuTTY utilities.

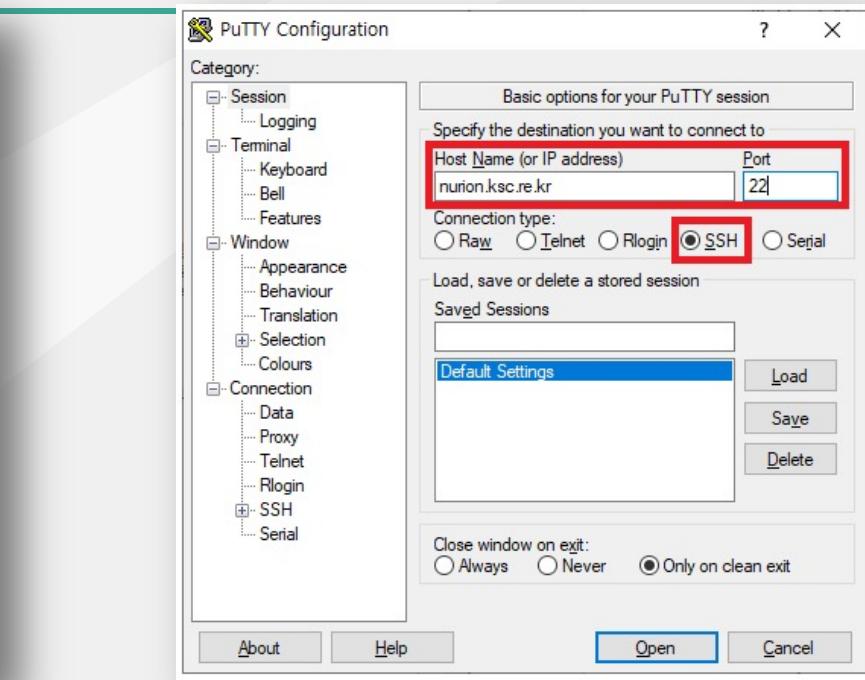
(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

MSI ('Windows Installer')

64-bit x86:	putty-64bit-0.76-installer.msi	(or by FTP)	(signature)
64-bit Arm:	putty-arm64-0.76-installer.msi	(or by FTP)	(signature)
32-bit x86:	putty-0.76-installer.msi	(or by FTP)	(signature)

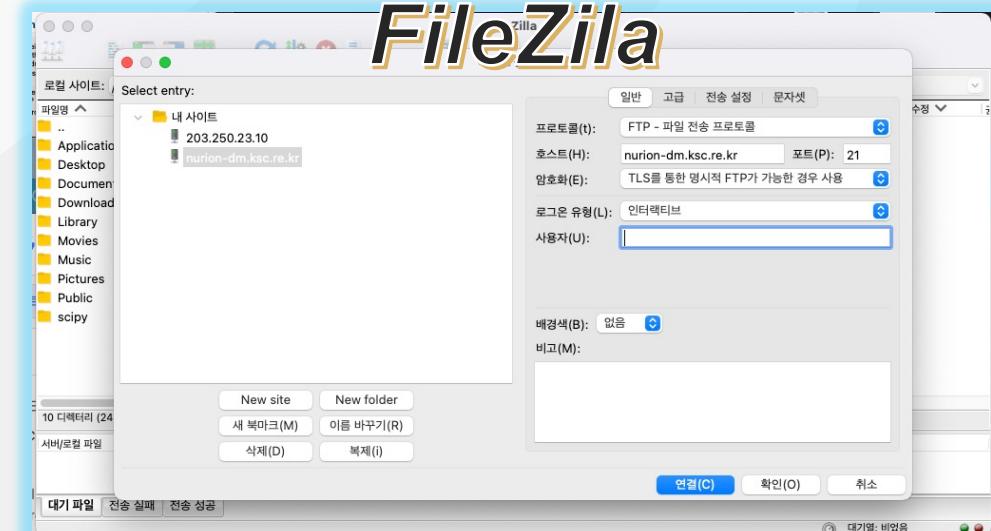
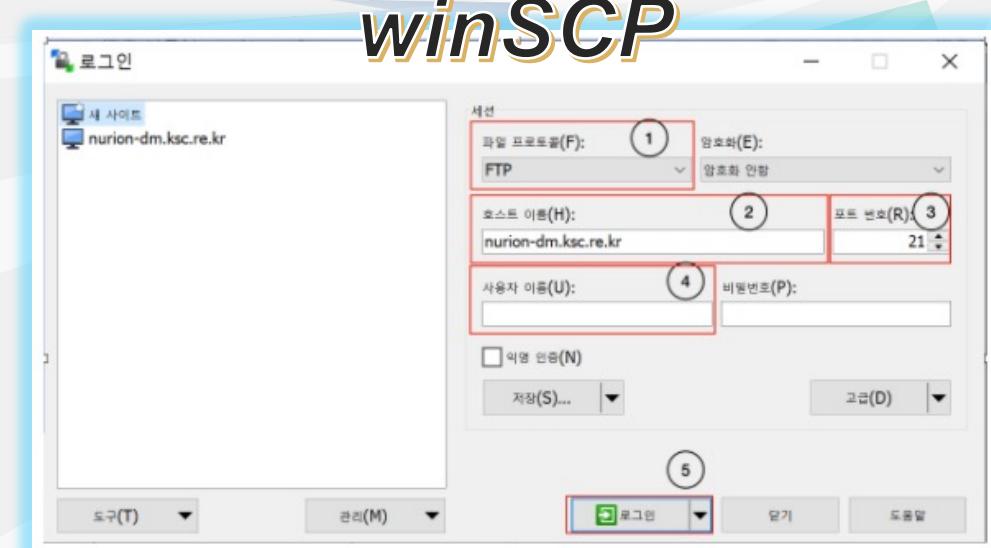
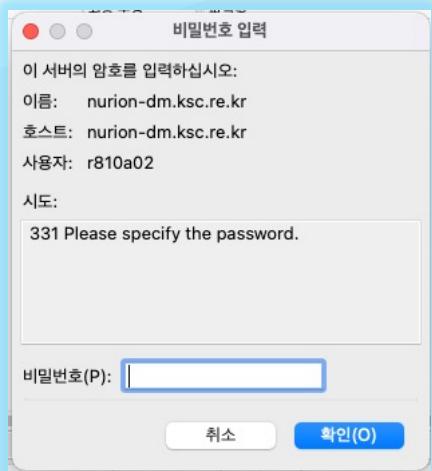
Unix source archive

.tar.gz:	putty-0.76.tar.gz	(or by FTP)	(signature)
----------	-----------------------------------	------------------------------	-------------------------------



Sending and receiving files

- Datamover node: nurion-dm.ksc.re.kr
- Unix or Linux environment: using
- Windows environment: using WinSCP
- Windows environment: using FileZilla



Working Directory and Quota Policy



Password(OTP):

Password:

===== KISTI 5th NURION System =====

* Any unauthorized attempts to use/access the system can be investigated and prosecuted by the related Act
(ACT ON THE PROTECTION OF INFORMATION AND COMMUNICATIONS INFRASTRUCTURE)

* Queue Policies

Queue	Wall-Clock Limit	Max Running Jobs	Max Active Jobs (running+waiting)
- normal	48h	180	200
- long	48h~120h	12	16
- flat	48h	25	30
- debug	12h	2	2
- commercial	48h	2	6
- norm_skl	48h	25	30

(Use the #showq & #pbs_status commands for more queue info.)

* Mandatory PBS Application Name option (#PBS -A App_Name)

- Allowed App_Name: ansys lsdyna nastran gaussian openfoam wrf cesm mpas roms grims vasp gromacs charmm qchem amber lammmps namd qe qmc bwa inhouse tf caffe pytorch siesta ramses cp2k gamess etc

* From May 25th, 2021 Ansys Software services will be suspended, due to license issue on using nurion system

- Ansys Fluent , Ansys CFX , Ansys HPC

* Preventive Maintenance

- 2021-09-08 09:00 ~ 2021-09-08 18:00 (9H)

* Failures of Scheduler(PBS)

- 2021-09-01 17:00 ~ 18:00

* More details can be found on <https://www.ksc.re.kr>

===== Account Information =====

* Account : na0926
* Due Date : 2021/11/02
* Allocated SRU Time : 3,744,000 [sec]
> Used SRU Time : 189,565 [sec]

* Available SRU Time : 3,554,435 [sec] [94.94%]
@ Available KNL CORE Time = Available SRU Time x 4,352
@ Available SKL CORE Time = Available SRU Time x 1,280

** more information : <https://www.ksc.re.kr>

Filesystem	KBytes	Quota	Files	Quota
/home01	277.7M	64G	1274	200000
/scratch	1.678G	100T	54563	1000000

Setting compiler environment



- You can add modules such as compilers and libraries you want to use.
 - [module load python/3.7](#)

```
[r810a02@login01 ~]$ python3
-bash: python3: command not found
[r810a02@login01 ~]$ module load python/3.7
[r810a02@login01 ~]$ python3
Python 3.7.0 (default, Jun 28 2018, 13:15:42)
[GCC 7.2.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
[>>> print('hello Nurion!')
hello Nurion!
```

Execution of tasks via scheduler (PBS)



- The job scheduler of Nurion System is PBS (the Portable Batch System).

Queue name	characteristic
exclusive	Dedicated resource queue to support large-scale challenging research and group research among R&D innovation support programs
normal	General resource queue for free service users (creative research field, national strategy field, innovation support field) and paid service users
long	A general resource queue that can be used for up to 120 hours (5 days) as a queue that requires long-term operation.
flat	Memory mode is Flat, and MCDRAM (16GB) and DDR4 can be specified and used, and a general resource queue that can use up to 102GB of memory

PBS Task Scheduler Required Options



Required option	Explanation
#PBS -V	Keep current environment variables
#PBS -N	Set task name
#PBS -q	queue to run the job
#PBS -l	Set up the resources to use for the job
#PBS -A	Information about programs used

■ Serial 프로그램 작업 스크립트 작성 예제(serial.sh)

```
#!/bin/sh  
  
#PBS -N serial_job  
  
#PBS -V  
  
#PBS -q normal  
  
#PBS -A {PBS 옵션 이름} # Application별 PBS 옵션 이름표 참고  
#PBS -l select=1:ncpus=1:mpiprocs=1:ompthreads=1  
#PBS -l walltime=04:00:00  
#PBS -m abe # 작업 이메일 알림 옵션  
#PBS -M abc@def.com # 수신할 메일 주소  
  
cd $PBS_O_WORKDIR  
  
module purge  
module load craype-mic-knl  
  
./test.exe
```



02

About Stock Market data



Basic terms in stocks



- Stock price:
 - **Market Price:** The price of a stock immediately after the market opens.
 - **Closing Price:** The price of a stock when the market closes.
 - **High and Low:** The price at which the stock was at its highest/lowest during the day.

Basic terms in stocks



- **Trading Volume:**
 - The total amount of shares traded on the stock market.
 - A trading volume of **1** means that each of the buy and sell shares were **made one share each**.
 - Trading volume is an **important indicator of the market's interest** in the stock and investor sentiment.

Major World Market Indices



- **United States:** New York Stock Exchange (NYSE) & NASDAQ.
 - **The New York Stock Exchange:** mainly lists large stocks
 - **The NASDAQ includes:** many IT or future industries, technology, and startups such as Google and Apple.
- **The three major U.S. indices:** the Nasdaq, Dow Jones Industrial Composite, and S&P 500.

Major World Market Indices



1. Nasdaq Composite Index

- It is the market capitalization-weighted index of over 2,500 common equities listed on the **Nasdaq stock exchange**.
- Most of the NASDAQ listed companies are **companies in the high-tech, IT, and future fields**
- Apple Inc., Amazon.com, Inc., Facebook, Inc., and Alphabet Inc.

Major World Market Indices

2. S&P 500

- The S&P 500 is a ‘Standard & Poor's 500 Stock Index’ that includes stocks of 500 large companies.
- The number of stocks included in the index calculation is 500, far more than the 30 of the Dow.
- Tesla Inc., Microsoft Corp., Nvidia Corp., Pfizer Inc., JPMorgan Chase & Co.

Major World Market Indices



3. Dow Jones

- It is a stock index created by Charles Dow.
- Today, the Dow is comprised of stocks of 30 blue-chip companies listed on the US stock exchange.
- 3M Company, AT&T, General Electric Company, Intel Corp., Wal-Mart Stores Inc., Walt Disney Company

Major World Market Indices



- **Shanghai Composite (China)**
 - It is a comprehensive stock index consisting of 1492 (as of January 1, 2021) companies listed on the [Shanghai Stock Exchange](#), China.
- **KOSPI Index (Korea)**
 - Korea Composite Stock Price Index

using Yahoo Finance @ python



- install libraries using pip3
 - pandas
 - pandas_datareader
 - datetime
 - matplotlib

using Yahoo Finance @ python



- **pandas_datareader.DataReader**

```
import pandas_datareader as data
```

```
import datetime
```

```
start_date = datetime.datetime(2020,1,1)
```

```
end_date = datetime.datetime(2021,9,20)
```

```
google_data = data.DataReader('GOOGL','yahoo', start_date,  
end_date)
```

```
print(google_data.head(9))
```

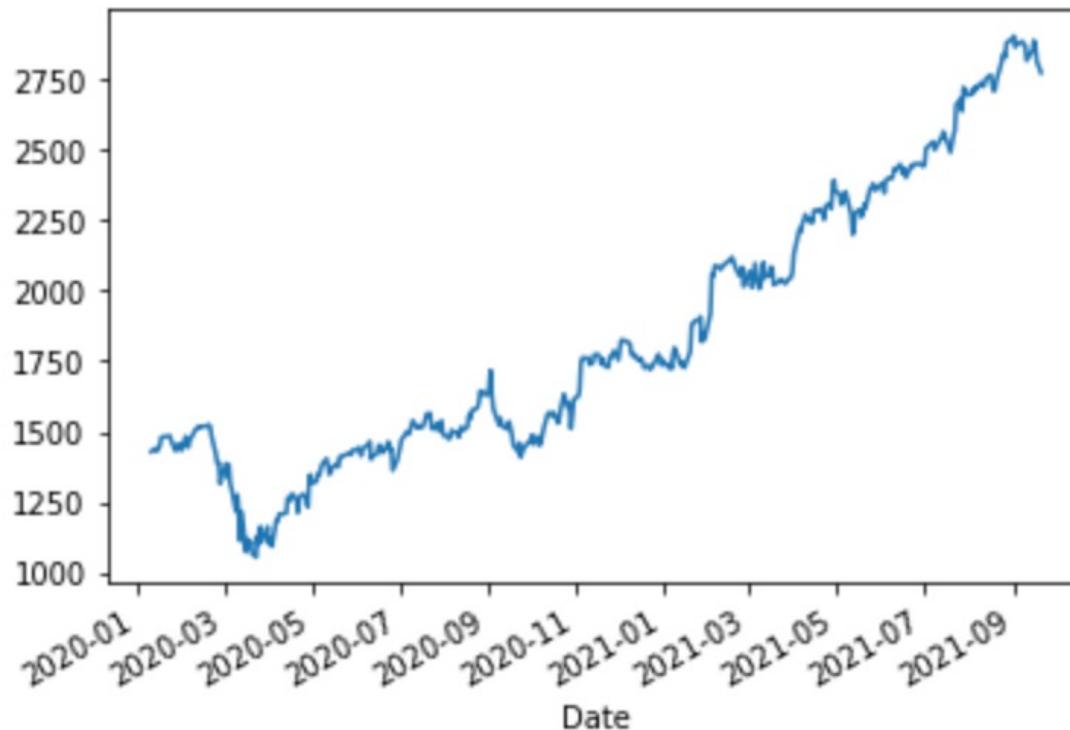
using Yahoo Finance @ python



Date	High	Low	Open	Close	Volume	Adj Close
2020-01-10	1434.939941	1419.599976	1429.469971	1428.959961	1312900	1428.959961
2020-01-13	1441.479980	1425.369995	1435.250000	1440.030029	1536500	1440.030029
2020-01-14	1442.630005	1427.770020	1440.000000	1430.589966	1303800	1430.589966
2020-01-15	1440.780029	1431.660034	1433.020020	1439.199951	1077500	1439.199951
2020-01-16	1450.699951	1440.000000	1445.449951	1450.160034	1304000	1450.160034
2020-01-17	1480.550049	1456.550049	1462.540039	1479.520020	2621200	1479.520020
2020-01-21	1489.880005	1470.209961	1479.000000	1482.250000	2446500	1482.250000
2020-01-22	1500.579956	1482.660034	1489.729980	1483.869995	1422900	1483.869995
2020-01-23	1493.500000	1480.319946	1487.550049	1484.689941	1332500	1484.689941

using Yahoo Finance @ python

```
google _data['Close'].plot()
```



1. Comparison before and after COVID-19 in the same company
2. Compare different companies in the same time period.
3. Compare changes in stock indexes in each country.
4. Analyze the correlation between the occurrence of COVID-19 and stock factors by country.

Kosdaq Composite Index (^KQ11)



yahoo! finance Search for news, symbols or companies

Kosdaq Composite Index (^KQ11)
KOSDAQ - KOSDAQ Delayed Price. Currency in KRW

1,046.12 +6.69 (+0.64%)
At close: September 17 6:03PM KST

[Summary](#) [Chart](#) [Conversations](#) [Historical Data](#) [Options](#) [Components](#)

1D 5D 1M 6M YTD 1Y 5Y Max

09 AM 11 AM 01 PM 03 PM

Previous Close **1,039.43** Day's Range **1,033.69 - 1,046.13**

Open **1,040.73** 52 Week Range **766.96 - 1,062.03**

Volume **1,160,928** Avg. Volume **59,452**

yahoo! finance Search for news, symbols or companies

Kosdaq Composite Index (^KQ11)
KOSDAQ - KOSDAQ Delayed Price. Currency in KRW

1,046.12 +6.69 (+0.64%)
At close: September 17 6:03PM KST

[Summary](#) [Chart](#) [Conversations](#) [Historical Data](#) [Options](#) [Components](#)

Worksport
WKSP | Worksport Ltd

Time Period: Sep 22, 2020 - Sep 22, 2021 Show: Historical Prices

Currency in KRW

Date	Open	High	Low	Close*	Adj Close**	Volume
Sep 17, 2021	1,040.73	1,046.13	1,033.69	1,046.12	1,046.12	1,200
Sep 16, 2021	1,047.85	1,048.37	1,037.72	1,039.43	1,039.43	1,100
Sep 15, 2021	1,039.57	1,044.26	1,038.80	1,042.79	1,042.79	1,400
Sep 14, 2021	1,029.57	1,038.94	1,028.40	1,037.74	1,037.74	1,200

Kosdaq Composite Index (^KQ11)

Time Period: Jan 01, 2019 - Sep 22, 2021 ▾

Frequency: Daily ▾

Show: Historical Prices ▾

Apply

Currency in KRW

Download

Date	Open	High	Low	Close*	Adj Close**	Volume
------	------	------	-----	--------	-------------	--------

^KQ11						
Date	Open	High	Low	Close	Adj Close	Volume
2019-01-02	682.159973	683.090027	667.710022	669.369995	669.369995	539500
2019-01-03	671.979980	673.609985	656.849976	657.020020	657.020020	656300
2019-01-04	655.619995	664.489990	648.950012	664.489990	664.489990	554500
2019-01-07	672.760010	675.309998	669.070007	672.840027	672.840027	584300
2019-01-08	674.530029	675.049988	667.010010	668.489990	668.489990	623500
2019-01-09	672.380005	681.119995	671.950012	679.739990	679.739990	622000
2019-01-10	680.609985	685.650024	678.229980	683.340027	683.340027	735100
2019-01-11	686.090027	688.010010	683.590027	686.330017	686.330017	701500
2019-01-14	687.419983	687.419983	682.539978	683.090027	683.090027	613300
2019-01-15	684.969971	691.559998	684.320007	690.390015	690.390015	639000
2019-01-16	692.719971	693.500000	687.429993	693.380005	693.380005	788300
2019-01-17	694.940002	696.210022	686.270020	686.349976	686.349976	862700
2019-01-18	690.830017	696.340027	689.799988	696.340027	696.340027	820000
2019-01-21	698.739990	698.739990	692.950012	695.619995	695.619995	794300
2019-01-22	698.000000	698.299988	692.450012	694.549988	694.549988	794400
2019-01-23	691.500000	697.059998	690.909973	695.630005	695.630005	683200
2019-01-24	696.969971	704.780029	696.799988	704.409973	704.409973	605800

How to download the KOSPI List

The screenshot shows the homepage of the KIND (KRX Information Disclosure Network) website. At the top, there is a search bar with a magnifying glass icon and a "SEARCH" button. Below the search bar, there is a menu with options like "KOREAN", "NOTICE", "ZOOM", and zoom control icons. On the left side, there is a sidebar with three main sections: "Latest Disclosures" (with a person icon), "Market Alert" (with an exclamation mark icon), and "New Listings" (with a building icon). Below these, there are links for "KOSPI Market", "KOSDAQ Market", and "KONEX Market". At the bottom of the sidebar, there is a link for "KRX Index" followed by "KOSPI 200" and its current value "410.99". At the very bottom, there are several navigation links: "Disclosure" (with a microphone icon), "Listed Companies" (which is highlighted with a red box), "Corporate Governance" (with a book icon), "IPOs" (with a globe icon), and "IR Information" (with a document icon).

The screenshot shows the "Company List" page of the KIND website. At the top, there is a breadcrumb navigation: "Home" → "Listed Companies" → "Company List". Below this, there is a section titled "Company List" with a search form. The search form includes fields for "Company Name" (with a placeholder), "Industry" (set to "ALL"), and "Market" (radio buttons for "ALL", "KOSPI Market" (selected), "KOSDAQ Market", and "KONEX Market"). There are also letter filters from A to Z and a numeric filter "0~9", along with "Search" and "EXCEL" buttons. Below the search form, there is a table of company results. The table has columns for "Company Name" (sorted by ascending name), "Telephone", "Industry", and "Main Business". The first few rows of the table are:

Company Name	Telephone	Industry	Main Business
K 0TO7	02-740-3188	Wholesale of Household Goods	
N 21store	033-761-8361	Manufacture of Other Non-metal..	
K 3S KOREA	02-896-9474	Manufacture of Electronic Comp..	
S A & P	032-676-9700	Manufacture of Electronic Comp..	
N A ONE ALFORM	031-8017-5305	Renting of Industrial Machiner..	
S A Plus Asset		Activities Auxiliary to Insura..	

'<https://kind.krx.co.kr/corpgeneral/corpList.do?method=download&marketType=stockMkt>

How to download the KOSPI List



```
kospicode =  
pd.read_html('https://kind.krx.co.kr/corpgeneral/corpList.do?method=download&m  
arketType=stockMkt', header=0)[0]  
print(kospicode.head(9))
```

	회사명	종목코드	업종	...	대표자명	홈페이지	지역
0	DRB동일	4840	고무제품 제조업	...	류영식	http://drbwold.com	부산광역시
1	DSR	155660	1차 비철금속 제조업	...	홍석빈	http://www.dsrl.com	부산광역시
2	GS	78930	기타 금융업	...	허태수, 홍순기 (각자 대표이사)	NaN	서울특별시
3	GS글로벌	1250	상품 종합 도매업	...	김태형	http://www.gsgcorp.com	서울특별시
4	HDC현대산업개발	294870	건물 건설업	...	권순호, 정경구	http://www.hdc-dvp.com	서울특별시
5	HMM	11200	해상 운송업	...	배재훈	http://www.hmm21.com	서울특별시
6	KEC	92220	반도체 제조업	...	박남규 김학남	http://www.kec.co.kr	서울특별시
7	KG동부제철	16380	1차 철강 제조업	...	박성희	http://www.kgdongbusteel.co.kr	서울특별시
8	KG케미칼	1390	기초 화학물질 제조업	...	곽정현, 김재익	http://www.kgchem.co.kr	울산광역시
9	KTis	58860	기타 정보 서비스업	...	윤경근	http://www.ktis.co.kr	서울특별시

[10 rows x 9 columns]

How to download the KOSPI List

- The company name and industry information → English
- There are many different ways to translate in python. Try to find it through Googling.

```
DRB is the same.          4840    Rubber product manufacturing.  
DSR      155660  The first non-ferrous metal manufacturing industry.  
GS       78930   Other financial businesses.  
GS Global.        1250   Product wholesale business.  
HDC Hyundai Development Company.    294870 Building construction industry.  
HMM      11200   Ocean transportation business.  
KEC      92220   Semiconductor manufacturing.  
KG Dongbu Steel.        16380  The first steel manufacturing industry.  
KG Chemical.        1390   Manufacturing of basic chemicals.  
KTis     58860   Other information service industries.
```



03

Connect your script to MySQL



How to use MySQL in Python..?



1. **Install** pymysql package
2. **Setup** the connection parameters of MySQL
3. **Create** a connection & Execute the MySQL query
4. **Receive** target data from MySQL Table

1. Install pymysql package

- **pymysql** : makes it easy to use MySQL in python.
- **installation**: pip3 install pymysql
- pymysql handles MySQL using cursor objects

2. Setup the connection parameters

```
• import pymysql  
• connection = pymysql.connect(host='localhost', user='_your_ID',  
password='_your_password', db='bigData', charset='utf8', autocommit=True,  
cursorclass = pymysql.cursors.DictCursor)
```

3. Create a connection & Execute the MySQL query



- `cursor = connection.cursor()`
- `sql = 'CREATE TABLE ' + tableName + '(_your_condition_here);'`
- `cursor.execute(sql)`
- `connection.commit()`
- `connection.close()`

4-1. Receive target data from MySQL Table



- `sql = 'SELECT _your_condition_here from ' + tableName + '
_your_condition_here '`
- `cursor.execute(sql)`
- `result = cursor.fetchall()`
- `df = pd.DataFrame(result)`
- `connection.close()`

4-2. Receive target data from MySQL Table



```
handle = open(filename, 'w')
handle.write('return_year,return_month,country,count\n')

for kk in range(len(df)):
    month = ''
    country = ""
    count = ""

    month = str(str(df.return_month.values[kk]))
    country = str(str(df.country.values[kk]))
    count = str(str(df['sum(count)'].values[kk]))

    handle.write(targetYear + ',' + month + ',' + country + ',' + count + '\n')

handle.close()
```



04

Make your own Database



1. Create a connection & MySQL Table



```
• import pymysql  
• connection = pymysql.connect(host='localhost', user='your_ID', password='your_password',  
    db='bigData', charset='utf8', autocommit=True, cursorclass = pymysql.cursors.DictCursor)  
• cursor = connection.cursor()  
• sql = 'CREATE TABLE ' + tableName + '(id int, country text, year int, month int, day int,  
    confirmed int, deaths int, population int, PRIMARY KEY (id));'  
• cursor.execute(sql)  
• connection.close()
```

2-3. Extract target information from Crawled data



	Date	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	...
0	Lat	33.939110	41.1533	28.0339	42.5063	-11.2027	17.0608	-3.841610e+01	40.0691	-35.4735	...
1	Long	67.709953	20.1683	1.6596	1.5218	17.8739	-61.7964	-6.361670e+01	45.0382	149.0124	...
2	1/22/20	0.000000	0.0000	0.0000	0.0000	0.0000	0.0000	0.000000e+00	0.0000	0.0000	...
3	1/23/20	0.000000	0.0000	0.0000	0.0000	0.0000	0.0000	0.000000e+00	0.0000	0.0000	...
4	1/24/20	0.000000	0.0000	0.0000	0.0000	0.0000	0.0000	0.000000e+00	0.0000	0.0000	...
...
491	5/25/21	66903.000000	132229.0000	127361.0000	13664.0000	32933.0000	1258.0000	3.586736e+06	222139.0000	124.0000	...
492	5/26/21	67743.000000	132244.0000	127646.0000	13671.0000	33338.0000	1258.0000	3.622135e+06	222269.0000	124.0000	...
493	5/27/21	68366.000000	132264.0000	127926.0000	13682.0000	33607.0000	1258.0000	3.663215e+06	222409.0000	124.0000	...
494	5/28/21	69130.000000	132285.0000	128198.0000	13693.0000	33944.0000	1259.0000	3.702422e+06	222513.0000	124.0000	...
495	5/29/21	70111.000000	132297.0000	128456.0000	13693.0000	34180.0000	1259.0000	3.732263e+06	222555.0000	124.0000	...

3. Send extracted data into MySQL Table



- `connection = pymysql.connect(host='localhost', user='your_ID', password='your_password', db='bigData', charset='utf8', autocommit=True, cursorclass = pymysql.cursors.DictCursor)`
- `cursor = connection.cursor()`
- `sql = "INSERT INTO " + tableName + " VALUES (" + idNum + ", \" + country + '\', ' + year + ', ' + month + ', ' + day + ', ' + str(eachConfirmed) + ',' + str(deaths) + ',' + population + ");"`
- `cursor.execute(sql)`
- `connection.close()`



Assignments





1. Review how to use a supercomputer using PBS.
2. Try applying for a supercomputer account for beginners.
3. Download stock data from Yahoo Finance.
4. Find a list of stock symbols of companies in your home country that you think have a high correlation with COVID-19.

THANK YOU

