Chapter 04 엑셀 기본 함수 구현하기

4.1 파이썬으로 엑셀 파일 다루기

파이썬 패키지 설치

In [1]:

```
!pip list
```

```
Package                            Version
---------------------------------- --------------------
alabaster                          0.7.12
anaconda-client                    1.9.0
anaconda-navigator                 2.1.1
anaconda-project                   0.10.1
anyio                              2.2.0
appdirs                            1.4.4
argh                               0.26.2
argon2-cffi                        20.1.0
arrow                              0.13.1
asn1crypto                         1.4.0
astroid                            2.6.6
astropy                            4.3.1
async-generator                    1.10
atomicwrites                       1.4.0
attrs                              21.2.0
autopep8                           1.5.7
Babel                              2.9.1
backcall                           0.2.0
backports.functools-lru-cache      1.6.4
backports.shutil-get-terminal-size 1.0.0
backports.tempfile                 1.0
backports.weakref                  1.0.post1
bcrypt                             3.2.0
beautifulsoup4                     4.10.0
binaryornot                        0.4.4
bitarray                           2.3.0
bkcharts                           0.2
black                              19.10b0
bleach                             4.0.0
bokeh                              2.4.1
boto                               2.49.0
Bottleneck                         1.3.2
brotlipy                           0.7.0
bs4                                0.0.1
cached-property                    1.5.2
certifi                            2021.10.8
cffi                               1.14.6
chardet                            4.0.0
charset-normalizer                 2.0.4
click                              8.0.3
clipboard                          0.0.4
cloudpickle                        2.0.0
clyent                             1.2.2
colorama                           0.4.4
comtypes                           1.1.10
conda                              4.10.3
conda-build                        3.21.6
conda-content-trust                0+unknown
conda-pack                         0.6.0
conda-package-handling             1.7.3
conda-repo-cli                     1.0.4
conda-token                        0.3.0
conda-verify                       3.4.2
```

conda-verify	3.4.2
contextlib2	0.6.0.post1
cookiecutter	1.7.2
cryptography	3.4.8
cycler	0.10.0
Cython	0.29.24
cytoolz	0.11.0
daal4py	2021.3.0
dask	2021.10.0
debugpy	1.4.1
decorator	5.1.0
defusedxml	0.7.1
diff-match-patch	20200713
distributed	2021.10.0
docutils	0.17.1
entrypoints	0.3
et-xmlfile	1.1.0
fastcache	1.1.0
filelock	3.3.1
flake8	3.9.2
Flask	1.1.2
fonttools	4.25.0
fsspec	2021.10.1
future	0.18.2
gevent	21.8.0
glob2	0.7
greenlet	1.1.1
h11	0.13.0
h5py	3.2.1
HeapDict	1.0.1
html5lib	1.1
idna	3.2
imagecodecs	2021.8.26
imageio	2.9.0
imagesize	1.2.0
importlib-metadata	4.8.1
inflection	0.5.1
iniconfig	1.1.1
intervaltree	3.1.0
ipykernel	6.4.1
ipython	7.29.0
ipython-genutils	0.2.0
ipywidgets	7.6.5
isort	5.9.3
itsdangerous	2.0.1
jdcal	1.4.1
jedi	0.18.0
Jinja2	2.11.3
jinja2-time	0.2.0
joblib	1.1.0
json5	0.9.6
jsonschema	3.2.0
jupyter	1.0.0
jupyter-client	6.1.12
jupyter-console	6.4.0
jupyter-core	4.8.1
jupyter-server	1.4.1
jupyterlab	3.2.1
jupyterlab-pygments	0.1.2
jupyterlab-server	2.8.2
jupyterlab-widgets	1.0.0
keyring	23.1.0
kiwisolver	1.3.1
lazy-object-proxy	1.6.0
libarchive-c	2.9
llvmlite	0.37.0
locket	0.2.1
lxml	4.6.3
MarkupSafe	1.1.1
matplotlib	3.4.3
matplotlib-inline	0.1.2
mccabe	0.6.1
menuinst	1.4.18

menuinst	1.4.18
mistune	0.8.4
mkl-fft	1.3.1
mkl-random	1.2.2
mkl-service	2.4.0
mock	4.0.3
more-itertools	8.10.0
MouseInfo	0.1.3
mpmath	1.2.1
msgpack	1.0.2
multipledispatch	0.6.0
munkres	1.1.4
mypy-extensions	0.4.3
navigator-updater	0.2.1
nbclassic	0.2.6
nbclient	0.5.3
nbconvert	6.1.0
nbformat	5.1.3
nest-asyncio	1.5.1
networkx	2.6.3
nltk	3.6.5
nose	1.3.7
notebook	6.4.5
numba	0.54.1
numexpr	2.7.3
numpy	1.20.3
numpydoc	1.1.0
olefile	0.46
openpyxl	3.0.9
outcome	1.1.0
packaging	21.0
pandas	1.3.4
pandocfilters	1.4.3
paramiko	2.7.2
parso	0.8.2
partd	1.2.0
path	16.0.0
pathlib2	2.3.6
pathspec	0.7.0
patsy	0.5.2
pep8	1.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.4.0
pip	21.2.4
pkginfo	1.7.1
pluggy	0.13.1
ply	3.11
poyo	0.5.0
prometheus-client	0.11.0
prompt-toolkit	3.0.20
psutil	5.8.0
ptyprocess	0.7.0
py	1.10.0
PyAutoGUI	0.9.53
pycodestyle	2.7.0
pycosat	0.6.3
pycparser	2.20
pycurl	7.44.1
pydocstyle	6.1.1
pyerfa	2.0.0
pyflakes	2.3.1
PyGetWindow	0.0.9
Pygments	2.10.0
PyJWT	2.1.0
pylint	2.9.6
pyls-spyder	0.4.0
PyMsgBox	1.0.9
PyNaCl	1.4.0
pyodbc	4.0.0-unsupported
pyOpenSSL	21.0.0
pyparsing	3.0.4
pyperclip	1.8.2

pyperclip	1.8.2
pyreadline	2.1
PyRect	0.2.0
pyrsistent	0.18.0
PyScreeze	0.1.28
PySocks	1.7.1
pytest	6.2.4
python-dateutil	2.8.2
python-lsp-black	1.0.0
python-lsp-jsonrpc	1.0.0
python-lsp-server	1.2.4
python-pptx	0.6.21
python-slugify	5.0.2
pytweening	1.0.4
pytz	2021.3
PyWavelets	1.1.1
pywin32	228
pywin32-ctypes	0.2.0
pywinpty	0.5.7
PyYAML	6.0
pyzmq	22.2.1
QDarkStyle	3.0.2
qstylizer	0.1.10
QtAwesome	1.0.2
qtconsole	5.1.1
QtPy	1.10.0
regex	2021.8.3
requests	2.26.0
rope	0.19.0
Rtree	0.9.7
ruamel-yaml-conda	0.15.100
scikit-image	0.18.3
scikit-learn	0.24.2
scikit-learn-intelex	2021.20210714.120553
scipy	1.7.1
seaborn	0.11.2
selenium	4.1.3
Send2Trash	1.8.0
setuptools	58.0.4
simplegeneric	0.8.1
singledispatch	3.7.0
sip	4.19.13
six	1.16.0
sniffio	1.2.0
snowballstemmer	2.1.0
sortedcollections	2.1.0
sortedcontainers	2.4.0
soupsieve	2.2.1
Sphinx	4.2.0
sphinxcontrib-applehelp	1.0.2
sphinxcontrib-devhelp	1.0.2
sphinxcontrib-htmlhelp	2.0.0
sphinxcontrib-jsmath	1.0.1
sphinxcontrib-qthelp	1.0.3
sphinxcontrib-serializinghtml	1.1.5
sphinxcontrib-websupport	1.2.4
spyder	5.1.5
spyder-kernels	2.1.3
SQLAlchemy	1.4.22
statsmodels	0.12.2
sympy	1.9
tables	3.6.1
TBB	0.2
tblib	1.7.0
terminado	0.9.4
testpath	0.5.0
text-unidecode	1.3
textdistance	4.2.1
threadpoolctl	2.2.0
three-merge	0.1.1
tifffile	2021.7.2
tinycss	0.4
toml	0.10.2

```
toml                            0.10.2
toolz                           0.11.1
tornado                         6.1
tqdm                            4.62.3
traitlets                       5.1.0
trio                            0.20.0
trio-websocket                  0.9.2
typed-ast                       1.4.3
typing-extensions               3.10.0.2
ujson                           4.0.2
unicodecsv                      0.14.1
Unidecode                       1.2.0
urllib3                         1.26.7
watchdog                        2.1.3
wcwidth                         0.2.5
webencodings                    0.5.1
Werkzeug                        2.0.2
wheel                           0.37.0
whichcraft                      0.6.1
widgetsnbextension              3.5.1
win-inet-pton                   1.1.0
win-unicode-console             0.5
wincertstore                    0.2
wrapt                           1.12.1
wsproto                         1.1.0
xlrd                            2.0.1
XlsxWriter                      3.0.1
xlwings                         0.24.9
xlwt                            1.3.0
xmltodict                       0.12.0
yapf                            0.31.0
zict                            2.0.0
zipp                            3.6.0
zope.event                      4.5.0
zope.interface                  5.4.0
```

In [1]:

```
# pandas 패키지를 pd라는 별명으로 불러오기
import pandas as pd
```

데이터 프레임 생성하기

In [3]:

```
# 딕셔너리 유형으로 data를 만든 후 데이터 프레임 생성하기
import pandas as pd
data = {"이름" : ["홍길동", "이순신", "강감찬", "임꺽정", "이성계"],
        "출생년도" : [1980, 1986, 1990, 1985, 1988],
        "점수" : [1.5, 1.7, 3.6, 2.4, 2.9]}
df = pd.DataFrame(data)
df
```

Out[3]:

	이름	출생년도	점수
0	홍길동	**1980**	**1.5**
1	이순신	**1986**	**1.7**
2	강감찬	**1990**	**3.6**
3	임꺽정	**1985**	**2.4**
4	이성계	**1988**	**2.9**

In [4]:

```
# df의 열 이름 변경하기
df = df.rename({"출생년도":"출생"}, axis = "columns")
```

```
df
```

Out[4]:

	이름	출생	점수
0	홍길동	1980	1.5
1	이순신	1986	1.7
2	강감찬	1990	3.6
3	임꺽정	1985	2.4
4	이성계	1988	2.9

행과 열 추가 및 삭제하기

In [5]:

```
# df의 열 데이터 가져오기
df[["이름", "출생"]] #[이름]과 [출생] 열 데이터 가져오기
```

Out[5]:

	이름	출생
0	홍길동	1980
1	이순신	1986
2	강감찬	1990
3	임꺽정	1985
4	이성계	1988

In [6]:

```
# df의 행 데이터 가져오기
df[1:3]  # 1행부터 2행 데이터 가져오기
```

Out[6]:

	이름	출생	점수
1	이순신	1986	1.7
2	강감찬	1990	3.6

In [7]:

```
# df에서 [이름]과 [출생] 열의 0행과 1행을 가져오기
df.loc[0:1, ["이름", "출생"]]  # [0:1] 행 선택, [이름], [출생] 열 선택
```

Out[7]:

	이름	출생
0	홍길동	1980
1	이순신	1986

In [8]:

```
df # df 출력
```

Out[8]:

	이름	출생	점수
0	홍길동	1980	1.5

	이름	출생	점수
1	이순신	1986	1.7
2	강감찬	1990	3.6
3	임꺽정	1985	2.4
4	이성계	1988	2.9

In [9]:

```
df["보너스"] = df["점수"] * 5     # [보너스] 열 추가
df
```

Out[9]:

	이름	출생	점수	보너스
0	홍길동	1980	1.5	7.5
1	이순신	1986	1.7	8.5
2	강감찬	1990	3.6	18.0
3	임꺽정	1985	2.4	12.0
4	이성계	1988	2.9	14.5

In [10]:

```
df["지역"] = ["서울", "서울", "부산", "대구", "인천"] # [지역] 열 데이터 추가
df
```

Out[10]:

	이름	출생	점수	보너스	지역
0	홍길동	1980	1.5	7.5	서울
1	이순신	1986	1.7	8.5	서울
2	강감찬	1990	3.6	18.0	부산
3	임꺽정	1985	2.4	12.0	대구
4	이성계	1988	2.9	14.5	인천

In [11]:

```
del df["보너스"] # 보너스 열 삭제
df
```

Out[11]:

	이름	출생	점수	지역
0	홍길동	1980	1.5	서울
1	이순신	1986	1.7	서울
2	강감찬	1990	3.6	부산
3	임꺽정	1985	2.4	대구
4	이성계	1988	2.9	인천

In [12]:

```
df.loc[5] = ["김순신", 1980, 3.3, "광주"]  # 인덱스가 5인 행을 추가
df
```

Out[12]:

	이름	출생	점수	지역
0	홍길동	1980	1.5	서울
1	이순신	1986	1.7	서울

	이름	출생	점수	지역
2	강감찬	1990	3.6	부산
3	임꺽정	1985	2.4	대구
4	이성계	1988	2.9	인천
5	김순신	1980	3.3	광주

In [13]:

```
df.iloc[4, 1] = 1999    # 4행 1열의 데이터를 1999로 변경
df.drop(5, inplace = True)    # 5행을 삭제
df
```

Out[13]:

	이름	출생	점수	지역
0	홍길동	1980	1.5	서울
1	이순신	1986	1.7	서울
2	강감찬	1990	3.6	부산
3	임꺽정	1985	2.4	대구
4	이성계	1999	2.9	인천

In [14]:

```
df1 = df.copy()       # 원 데이터 보존을 위해 df를 df1로 복사
id = df1[df1["점수"] <= 2.0].index    # df1["점수"] <= 2.0인 행의 인덱스를 id에 저장
df1.drop(id, inplace = True)  # id에 저장된 인덱스로 행 삭제
df1
```

Out[14]:

	이름	출생	점수	지역
2	강감찬	1990	3.6	부산
3	임꺽정	1985	2.4	대구
4	이성계	1999	2.9	인천

엑셀 파일 읽고 쓰기

In [15]:

```
# 데이터 프레임 변수인 df를 현재 디렉토리에 "명단.xlsx"로 저장하기
df.to_excel("명단.xlsx")
```

In [16]:

```
# 현재 디렉토리의 "명단.xlsx" 파일을 불러와 df12에 저장하기
df12 = pd.read_excel("명단.xlsx")
# 첫 번째 열을 인덱스로 지정하는 코드; df12 = pd.read_excel("명단.xlsx", index_col = 0)
```

4.2 텍스트 함수

In [17]:

```
# pandas를 pd라는 이름으로 불러오기
import pandas as pd

# 텍스트 함수 실습을 위한 직원 정보 엑셀을 불러와 데이터 프레임 info에 저장하기
info = pd.read_excel(r"c:\works\chapter04\직원 정보.xlsx", sheet_name = "Sheet1")
info
```

Out[17]:

	순번	성	이름	영문명	사원번호	주소	전화번호
0	1	김	철수	kim cheolsu	2005-1478345	강서구 공항로 20455	010-1000-8735
1	2	박	종수	park jongsu	2010-1345972	강서구 대저중앙로 3009	010-1000-8736
2	3	김	하나	kim hana	2012-1487234	강서구 하덕로 1002	010-1000-8737
3	4	이	백만	lee baekman	2001-3747234	기장군 기장읍 연화100길	010-1000-8738
4	5	백	오십	baek osip	2002-4972944	기장군 기장읍 차성로	010-1000-8739
5	6	영웅	재준	youngwoong jaejun	2011-2382747	기장군 기장읍 기장해안로	010-1000-8740
6	7	현	빈	hyun bin	2000-2395634	남구 지게골로 10-2	010-1000-8741
7	8	장	하나	jang hana	2015-1626816	동래구 온천장로107-100	010-1000-8742
8	9	유	두울	yoo dooul	2016-4323930	동래구 동래로116	010-1000-8743
9	10	채	일	chae il	2012-3967845	북구 효열로 2502	010-1000-8744

여러 셀의 문자 합치기

In [18]:

```
info = pd.read_excel(r"c:\works\chapter04\직원 정보.xlsx", sheet_name = "Sheet1")
info["성명"] = info["성"] + info["이름"]    # 데이터 프레임 열 연산
info["성명1"] = info[["성", "이름"]].sum(1)   # 열 방향으로 각 행의 문자를 결합
info           # 데이터 프레임 info 출력
```

Out[18]:

	순번	성	이름	영문명	사원번호	주소	전화번호	성명	성명1
0	1	김	철수	kim cheolsu	2005-1478345	강서구 공항로 20455	010-1000-8735	김철수	김철수
1	2	박	종수	park jongsu	2010-1345972	강서구 대저중앙로 3009	010-1000-8736	박종수	박종수
2	3	김	하나	kim hana	2012-1487234	강서구 하덕로 1002	010-1000-8737	김하나	김하나
3	4	이	백만	lee baekman	2001-3747234	기장군 기장읍 연화100길	010-1000-8738	이백만	이백만
4	5	백	오십	baek osip	2002-4972944	기장군 기장읍 차성로	010-1000-8739	백오십	백오십
5	6	영웅	재준	youngwoong jaejun	2011-2382747	기장군 기장읍 기장해안로	010-1000-8740	영웅재준	영웅재준
6	7	현	빈	hyun bin	2000-2395634	남구 지게골로 10-2	010-1000-8741	현빈	현빈
7	8	장	하나	jang hana	2015-1626816	동래구 온천장로107-100	010-1000-8742	장하나	장하나
8	9	유	두울	yoo dooul	2016-4323930	동래구 동래로116	010-1000-8743	유두울	유두울
9	10	채	일	chae il	2012-3967845	북구 효열로 2502	010-1000-8744	채일	채일

몇 개의 문자만 추출하기

In [19]:

```
info = pd.read_excel(r"c:\works\chapter04\직원 정보.xlsx", sheet_name = "Sheet1")
info["사원번호 앞 4자리"] = info["사원번호"].str[0:4]  # 사원번호 앞에서 4자리 추출
info["전화번호 뒤 4자리"] = info["전화번호"].str[9:13] # 전화번호 뒤에서 4자리 추출
info
```

Out[19]:

	순번	성	이름	영문명	사원번호	주소	전화번호	사원번호 앞 4자리	전화번호 뒤 4자리
0	1	김	철수	kim cheolsu	2005-1478345	강서구 공항로 20455	010-1000-8735	2005	8735
1	2	박	종수	park jongsu	2010-1345972	강서구 대저중앙로 3009	010-1000-8736	2010	8736
					2012-		010-1000-		

	순번	성	이름	영문명	사원번호	주소	전화번호	사원번호 앞 4자리	전화번호 뒤 4자리
2	3	김	하나	kim hana	2012-1487234	강서구 하덕로 1002	010-1000-8737	2012	8737
3	4	이	백만	lee baekman	2001-3747234	기장군 기장읍 연화100길	010-1000-8738	2001	8738
4	5	백	오십	baek osip	2002-4972944	기장군 기장읍 차성로	010-1000-8739	2002	8739
5	6	영웅	재준	youngwoong jaejun	2011-2382747	기장군 기장읍 기장해안로	010-1000-8740	2011	8740
6	7	현	빈	hyun bin	2000-2395634	남구 지게골로 10-2	010-1000-8741	2000	8741
7	8	장	하나	jang hana	2015-1626816	동래구 온천장로107-100	010-1000-8742	2015	8742
8	9	유	두울	yoo dooul	2016-4323930	동래구 동래로116	010-1000-8743	2016	8743
9	10	채	일	chae il	2012-3967845	북구 효열로 2502	010-1000-8744	2012	8744

In [20]:

```
info = pd.read_excel(r"c:\works\chapter04\직원 정보.xlsx", sheet_name = "Sheet1")
info["구"] = info["주소"].str.split(" ").str[1]
info[["성", "이름", "전화번호", "구"]]
```

Out[20]:

	성	이름	전화번호	구
0	김	철수	010-1000-8735	강서구
1	박	종수	010-1000-8736	강서구
2	김	하나	010-1000-8737	강서구
3	이	백만	010-1000-8738	기장군
4	백	오십	010-1000-8739	기장군
5	영웅	재준	010-1000-8740	기장군
6	현	빈	010-1000-8741	남구
7	장	하나	010-1000-8742	동래구
8	유	두울	010-1000-8743	동래구
9	채	일	010-1000-8744	북구

영문 대소문자 바꾸기

In [21]:

```
info = pd.read_excel(r"c:\works\chapter04\직원 정보.xlsx", sheet_name = "Sheet1")
info["대문자"] = info["영문명"].str.upper()  # [영문명] 열을 대문자로 변환
info["첫 글자"] = info["영문명"].str.capitalize() # [영문명] 열의 첫 글자만 대문자로 변환
info[["순번", "성", "이름", "영문명", "전화번호", "대문자", "첫 글자"]]
```

Out[21]:

	순번	성	이름	영문명	전화번호	대문자	첫 글자
0	1	김	철수	kim cheolsu	010-1000-8735	KIM CHEOLSU	Kim cheolsu
1	2	박	종수	park jongsu	010-1000-8736	PARK JONGSU	Park jongsu
2	3	김	하나	kim hana	010-1000-8737	KIM HANA	Kim hana
3	4	이	백만	lee baekman	010-1000-8738	LEE BAEKMAN	Lee baekman
4	5	백	오십	baek osip	010-1000-8739	BAEK OSIP	Baek osip
5	6	영웅	재준	youngwoong jaejun	010-1000-8740	YOUNGWOONG JAEJUN	Youngwoong jaejun
6	7	현	빈	hyun bin	010-1000-8741	HYUN BIN	Hyun bin

	순번	성	이름	영문명	전화번호	대문자	첫 글자
7	8	장	하나	jang hana	010-1000-8742	JANG HANA	Jang hana
8	9	유	두울	yoo dooul	010-1000-8743	YOO DOOUL	Yoo dooul
9	10	채	일	chae il	010-1000-8744	CHAE IL	Chae il

In [22]:

```
str.swapcase("ABcde")
```

Out[22]:

```
'abCDE'
```

특정 문자 바꾸기

In [23]:

```
info = pd.read_excel(r"c:\works\chapter04\직원 정보.xlsx", sheet_name = "Sheet1")
info["phone"] = info["전화번호"].str.replace("-", "", 1)  # 첫 번째 하이픈(-) 제거
info["phone1"] = info["전화번호"].str.replace("-"," ")  # 모든 하이픈 (-)을 공백으로 대체
info[["순번", "성", "이름", "전화번호", "phone", "phone1"]]
```

Out[23]:

	순번	성	이름	전화번호	phone	phone1
0	1	김	철수	010-1000-8735	0101000-8735	010 1000 8735
1	2	박	종수	010-1000-8736	0101000-8736	010 1000 8736
2	3	김	하나	010-1000-8737	0101000-8737	010 1000 8737
3	4	이	백만	010-1000-8738	0101000-8738	010 1000 8738
4	5	백	오십	010-1000-8739	0101000-8739	010 1000 8739
5	6	영웅	재준	010-1000-8740	0101000-8740	010 1000 8740
6	7	현	빈	010-1000-8741	0101000-8741	010 1000 8741
7	8	장	하나	010-1000-8742	0101000-8742	010 1000 8742
8	9	유	두울	010-1000-8743	0101000-8743	010 1000 8743
9	10	채	일	010-1000-8744	0101000-8744	010 1000 8744

문자열 길이 구하기

In [24]:

```
info = pd.read_excel(r"c:\works\chapter04\직원 정보.xlsx", sheet_name = "Sheet1")
info["주소 길이"] = info["주소"].str.len()        # [주소] 열의 문자열 길이 반환
info[["순번", "성", "이름", "주소", "주소 길이" ]].head()
```

Out[24]:

	순번	성	이름	주소	주소 길이
0	1	김	철수	강서구 공항로 20455	14
1	2	박	종수	강서구 대저중앙로 3009	15
2	3	김	하나	강서구 하덕로 1002	13
3	4	이	백만	기장군 기장읍 연화100길	15
4	5	백	오십	기장군 기장읍 차성로	12

문자열 공백 삭제하기

In [25]:

```
info = pd.read_excel(r"c:\works\chapter04\직원 정보.xlsx", sheet_name = "Sheet1")
info["주소 길이"] = info["주소"].str.len()  # [주소] 열의 길이 구하기
info["공백 제거"] = info["주소"].str.strip()  # [주소] 열의 문자열 앞뒤 공백 제거
info["공백 제거 후 길이"] = info["공백 제거"].str.len() # 공백 제거 문자열의 길이 구하기
info[["순번", "주소", "주소 길이", "공백 제거", "공백 제거 후 길이"]].tail(6)
```

Out[25]:

	순번	주소	주소 길이	공백 제거	공백 제거 후 길이
4	5	기장군 기장읍 차성로	12	기장군 기장읍 차성로	11
5	6	기장군 기장읍 기장해안로	14	기장군 기장읍 기장해안로	13
6	7	남구 지게골로 10-2	13	남구 지게골로 10-2	12
7	8	동래구 온천장로107-100	16	동래구 온천장로107-100	15
8	9	동래구 동래로116	11	동래구 동래로116	10
9	10	북구 효열로 2502	12	북구 효열로 2502	11

4.3 수학 및 통계 함수

실습 데이터 불러오기

In [26]:

```
# pandas를 pd라는 이름으로 불러오기
import pandas as pd

# 수학 및 통계 함수 실습을 위해 "성적 처리.xlsx" 파일을 불러와 score에 저장하기
score = pd.read_excel(r"c:\works\chapter04\성적 처리.xlsx", sheet_name = "Sheet1")
score  # 'c:\works\chapter04' 디렉토리에 파일을 복사한 후 진행
```

Out[26]:

	반	성명	국어	영어	수학	사회	과학
0	1반	홍길동	93	80	94	73	64
1	2반	백일홍	93	63	76	84	92
2	3반	이삼상	94	74	86	90	70
3	1반	정말로	83	55	64	90	65
4	2반	한번도	87	95	66	75	60
5	3반	이철수	53	81	59	88	69
6	1반	김영자	71	71	51	84	57
7	2반	다니엘	87	54	95	71	97
8	3반	이미로	59	54	75	90	82
9	1반	신성삼	64	66	59	91	86
10	2반	케로로	56	76	52	64	65
11	3반	장발장	85	51	64	80	68

데이터 합계 구하기

** 합계 출력 코드 수정

기존 : `total_korean`
변경 : `print(total_korean)`

In [5]:

```
# score = pd.read_excel(r"c:\works\chapter04\성적 처리.xlsx", sheet_name = "Sheet1")
score = pd.read_excel("./성적 처리.xlsx", sheet_name = "Sheet1")
score = score[:12]
total_korean = score["국어"].sum(axis=0)  # 행 방향 [국어] 열의 데이터 합계 구하기
print(total_korean)
```

```
925
```

In [28]:

```
score["sum"] = score.iloc[:, 2:7].sum(1)  # 2~6열 각 행의 데이터 합계 구하기
score["sum1"] = score["국어"] + score["영어"] + score["수학"] + score["사회"] + score["과학"]
score.head()
```

Out[28]:

	반	성명	국어	영어	수학	사회	과학	sum	sum1
0	1반	홍길동	93	80	94	73	64	404	404
1	2반	백일홍	93	63	76	84	92	408	408
2	3반	이삼상	94	74	86	90	70	414	414
3	1반	정말로	83	55	64	90	65	357	357
4	2반	한번도	87	95	66	75	60	383	383

데이터 평균 구하기

** 평균 출력 코드 수정

기존 : korean_avg
변경 : print(korean_avg)

In [8]:

```
# score = pd.read_excel(r"c:\works\chapter04\성적 처리.xlsx", sheet_name = "Sheet1")
korean_avg = score["국어"].mean()  # [국어] 열의 행 방향 평균 구하기
print(korean_avg)
```

```
77.08333333333333
```

In [30]:

```
score["평균"] = score.iloc[:, 2:7].mean(1)  # 2~6열의 열 방향 각 행의 평균 구하기
score["평균1"] = (score["국어"] + score["영어"] + score["수학"] + score["사회"] + score["과학"])/5
score
```

Out[30]:

	반	성명	국어	영어	수학	사회	과학	평균	평균1
0	1반	홍길동	93	80	94	73	64	80.8	80.8
1	2반	백일홍	93	63	76	84	92	81.6	81.6
2	3반	이삼상	94	74	86	90	70	82.8	82.8
3	1반	정말로	83	55	64	90	65	71.4	71.4
4	2반	한번도	87	95	66	75	60	76.6	76.6
5	3반	이철수	53	81	59	88	69	70.0	70.0
6	1반	김영자	71	71	51	84	57	66.8	66.8
7	2반	다니엘	87	54	95	71	97	80.8	80.8
8	3반	이미로	59	54	75	90	82	72.0	72.0

	반	성명	국어	영어	수학	사회	과학	평균	평균1
9	1반	신성섭	64	66	59	91	86	73.2	73.2
10	2반	케로로	56	76	52	64	65	62.6	62.6
11	3반	장발장	85	51	64	80	68	69.6	69.6

조건에 따른 합계, 평균 구하기

**** score1 코드 수정**

기존 : score1 = score.groupby(["반"]).sum()
기존 : score1 = score[["반","국어","영어","수학","사회","과학"]].groupby(["반"]).sum()

**** score2 코드 수정**

기존 : score1 = score.groupby(["반"]).mean()
기존 : score1 = score[["반","국어","영어","수학","사회","과학"]].groupby(["반"]).mean()

In [19]:

```
# score = pd.read_excel(r"c:\works\chapter04\성적 처리.xlsx", sheet_name = "Sheet1")
score1 = score[["반","국어","영어","수학","사회","과학"]].groupby(["반"]).sum()          # 반, 다섯 과목의 점수 데이터를 추출하여 [반] 열을 기준으로 그룹화한 후 합계 계산
score1[["국어","영어","수학","사회","과학"]]       # 반 별 다섯 과목의 점수 합계 출력
```

Out[19]:

반	국어	영어	수학	사회	과학
1반	157	146	153	164	150
2반	144	105	139	170	150
3반	221	221	189	238	192
4반	136	136	123	178	134
5반	267	212	237	230	249

In [22]:

```
score2 = score[["반","국어","영어","수학","사회","과학"]].groupby(["반"]).mean()   # 반, 다섯 과목의 점수 데이터를 추출하여 [반] 열을 기준으로 그룹화한 후 평균 계산
score2       # score2 출력
```

Out[22]:

반	국어	영어	수학	사회	과학
1반	78.500000	73.000000	76.5	82.000000	75.0
2반	72.000000	52.500000	69.5	85.000000	75.0
3반	73.666667	73.666667	63.0	79.333333	64.0
4반	68.000000	68.000000	61.5	89.000000	67.0
5반	89.000000	70.666667	79.0	76.666667	83.0

순위 구하기

In [33]:

```
score = pd.read_excel(r"c:\works\chapter04\성적 처리.xlsx", sheet_name = "Sheet1")
```

```
score["평균"] = score.iloc[:, 2:7].mean(1)    # 2~6열의 열 방향 각 행의 평균 구하기
score["순위_오름"] = score["평균"].rank(ascending = True)  # 동점 시 평균 순위 부여, 오름차순
score["순위_내림"] = score["평균"].rank(ascending = False) # 동점 시 평균 순위 부여, 내림차순
score["순위_내림_min"] = score["평균"].rank(method = "min", ascending = False)  # 동점 시 최소 순위
score
```

Out[33]:

	반	성명	국어	영어	수학	사회	과학	평균	순위_오름	순위_내림	순위_내림_min
0	1반	홍길동	93	80	94	73	64	80.8	9.5	3.5	3.0
1	2반	백일홍	93	63	76	84	92	81.6	11.0	2.0	2.0
2	3반	이삼상	94	74	86	90	70	82.8	12.0	1.0	1.0
3	1반	정말로	83	55	64	90	65	71.4	5.0	8.0	8.0
4	2반	한번도	87	95	66	75	60	76.6	8.0	5.0	5.0
5	3반	이철수	53	81	59	88	69	70.0	4.0	9.0	9.0
6	1반	김영자	71	71	51	84	57	66.8	2.0	11.0	11.0
7	2반	다니엘	87	54	95	71	97	80.8	9.5	3.5	3.0
8	3반	이미로	59	54	75	90	82	72.0	6.0	7.0	7.0
9	1반	신성삼	64	66	59	91	86	73.2	7.0	6.0	6.0
10	2반	케로로	56	76	52	64	65	62.6	1.0	12.0	12.0
11	3반	장발장	85	51	64	80	68	69.6	3.0	10.0	10.0

최대값/최소값 구하기

In [34]:

```
score = pd.read_excel(r"c:\works\chapter04\성적 처리.xlsx", sheet_name = "Sheet1")
score_row = score.iloc[:, [2, 3, 4, 5, 6]]  # score에서 숫자열만 추출하여 score_row에 저장
score["MIN"] = score_row.min(1)  # score_row의 행 방향 최소값을 score["MIN"]에 저장
score["MAX"] = score_row.max(1)  # score_row의 행 방향 최대값을 score["MAX"]에 저장
score
```

Out[34]:

	반	성명	국어	영어	수학	사회	과학	MIN	MAX
0	1반	홍길동	93	80	94	73	64	64	94
1	2반	백일홍	93	63	76	84	92	63	93
2	3반	이삼상	94	74	86	90	70	70	94
3	1반	정말로	83	55	64	90	65	55	90
4	2반	한번도	87	95	66	75	60	60	95
5	3반	이철수	53	81	59	88	69	53	88
6	1반	김영자	71	71	51	84	57	51	84
7	2반	다니엘	87	54	95	71	97	54	97
8	3반	이미로	59	54	75	90	82	54	90
9	1반	신성삼	64	66	59	91	86	59	91
10	2반	케로로	56	76	52	64	65	52	76
11	3반	장발장	85	51	64	80	68	51	85

In [35]:

```
score = pd.read_excel(r"c:\works\chapter04\성적 처리.xlsx", sheet_name = "Sheet1")
score.describe()  # score 데이터 프레임의 기초 통계량 확인
```

Out[35]:

	국어	영어	수학	사회	과학
count	12.000000	12.000000	12.000000	12.000000	12.000000
mean	77.083333	68.333333	70.083333	81.666667	72.916667
std	15.512214	13.580289	15.174490	9.018500	13.041600
min	53.000000	51.000000	51.000000	64.000000	57.000000
25%	62.750000	54.750000	59.000000	74.500000	64.750000
50%	84.000000	68.500000	65.000000	84.000000	68.500000
75%	88.500000	77.000000	78.500000	90.000000	83.000000
max	94.000000	95.000000	95.000000	91.000000	97.000000