

판다스는 데이터용 분석용 오픈소스 파이썬 라이브러리입니다.

데이터프레임과 시리즈라는 두가지 새로운 자료형을 제공하여, 스프레드 시트 형태의 데이터를 불러와 빠르게 조작, 정렬, 병합 할 수 있습니다. 한마디로 파이썬으로 다루는 엑셀 이라고 보시면 될꺼 같습니다.

데이터프레임은 전체스프레드시트 또는 직사각형의 형태의 데이터로 나타내고, 시리즈는 데이터 프레임을 한열을 나타냅니다. 시리즈를 여러개를 모은 딕셔너리나 컬렉션이 판다스의 데이터 프레임이라고 생각해도 좋습니다.

데이터를 다룰때에는 왜 파이썬같은 프로그래밍과 판다스 같은 도구를 사용해야될까요?

첫 번째는 여러 데이터셋에 같은 분석과정을 적용해야 할때에 일련의 작업을 자동화 할수 있기 때문입니다.

대부분 스프레드시트 프로그램은 마이크로 소프트 365의 VBA와 같은 고유한 매크로 프로그래밍 언어를 제공하지만 이를 활용하는 사람은 많지 않습니다.

매크로보다는 파이썬과 상관없이 작동하는 프로그램언어를 활용하는 것이 좋습니다.

두 번째는 데이터를 작업수행할때에 데이터에 적용한 모든 실행단계를 기록할수있다는 장점, 즉 재현성이 있기 때문입니다.

```
[10]: # 데이터셋불러오기
```

```
[5]: import pandas as pd # 판다스 라이브러리 불러오기
```

```
[7]: df=pd.read_csv("gapminder.tsv",sep='\t') # csv 파일 불러오기
```

```
[9]: print(df) # 출력
```

	country	continent	year	lifeExp	pop	gdpPercap
0	Afghanistan	Asia	1952	28.801	8425333	779.445314
1	Afghanistan	Asia	1957	30.332	9240934	820.853030
2	Afghanistan	Asia	1962	31.997	10267083	853.100710
3	Afghanistan	Asia	1967	34.020	11537966	836.197138
4	Afghanistan	Asia	1972	36.088	13079460	739.981106
...
1699	Zimbabwe	Africa	1987	62.351	9216418	706.157306
1700	Zimbabwe	Africa	1992	60.377	10704340	693.420786
1701	Zimbabwe	Africa	1997	46.809	11404948	792.449960
1702	Zimbabwe	Africa	2002	39.989	11926563	672.038623
1703	Zimbabwe	Africa	2007	43.487	12311143	469.709298

[1704 rows x 6 columns]

```
[11]: print(type(df)) # 자료형이 무엇인지
```

<class 'pandas.core.frame.DataFrame'>

```
[13]: print(df.shape) # 행과열의 개수 확인하기. 1704개의 행 6개의 열
```

(1704, 6)

```
[15]: print(df.columns) # 열의 이름
```

Index(['country', 'continent', 'year', 'lifeExp', 'pop', 'gdpPercap'], dtype='object')

```
[17]: print(df.dtypes) # 각열이 어떤자료형인지
```

country object
continent object
year int64
lifeExp float64
pop int64
gdpPercap float64
dtype: object

```
[ ]: # 데이터 추출하기
```

```
[19]: print(df.head) # 가장 앞 5개의 행을 확인할수있다.
```

```
<bound method NDFrame.head of
country continent year lifeExp pop gdpPercap
0 Afghanistan Asia 1952 28.801 8425333 779.445314
1 Afghanistan Asia 1957 30.332 9240934 820.853030
2 Afghanistan Asia 1962 31.997 10267083 853.100710
3 Afghanistan Asia 1967 34.020 11537966 836.197138
4 Afghanistan Asia 1972 36.088 13079460 739.981106
...
1699 Zimbabwe Africa 1987 62.351 9216418 706.157306
1700 Zimbabwe Africa 1992 60.377 10704340 693.420786
1701 Zimbabwe Africa 1997 46.809 11404948 792.449960
1702 Zimbabwe Africa 2002 39.989 11926563 672.038623
1703 Zimbabwe Africa 2007 43.487 12311143 469.709298
```

```
[1704 rows x 6 columns]>
```

```
[21]: country_df=df['country'] # df에서 country 열 데이터를 추출하고 그결과를country_df 저장
```

```
[23]: print(country_df.head()) # country의 가장 앞에 5개의 행 출력
```

```
0 Afghanistan
1 Afghanistan
2 Afghanistan
3 Afghanistan
4 Afghanistan
Name: country, dtype: object
```

```
[25]: print(country_df.tail()) # 가장 마지막 5개의 데이터 출력
```

```
1699 Zimbabwe
1700 Zimbabwe
1701 Zimbabwe
1702 Zimbabwe
1703 Zimbabwe
Name: country, dtype: object
```

```
[27]: subset=df[['country','continent','year']] # 3개의 열 데이터를 추출하고 그결과를 subset변수에 저장
```

```
[29]: print(subset)
```

```
country continent year
0 Afghanistan Asia 1952
1 Afghanistan Asia 1957
2 Afghanistan Asia 1962
3 Afghanistan Asia 1967
4 Afghanistan Asia 1972
...
1699 Zimbabwe Africa 1987
1700 Zimbabwe Africa 1992
1701 Zimbabwe Africa 1997
1702 Zimbabwe Africa 2002
1703 Zimbabwe Africa 2007
```

```
[1704 rows x 3 columns]
```

```
[31]: country_df=df['country']
```

```
print(type(country_df)) # 타입확인 시라즈 타입
<class 'pandas.core.series.Series'>
```

```
[33]: print(country_df)
```

```
0 Afghanistan
1 Afghanistan
2 Afghanistan
3 Afghanistan
4 Afghanistan
...
1699 Zimbabwe
1700 Zimbabwe
1701 Zimbabwe
1702 Zimbabwe
1703 Zimbabwe
Name: country, Length: 1704, dtype: object
```

```
[37]: country_df_list=df[['country']]
```

```
print(type(country_df_list)) # DataFrame 타입
<class 'pandas.core.frame.DataFrame'>
```

```
[39]: print(country_df_list) # 열데이터를 추출
```

```
country
0    Afghanistan
1    Afghanistan
2    Afghanistan
3    Afghanistan
4    Afghanistan
...
1699   Zimbabwe
1700   Zimbabwe
1701   Zimbabwe
1702   Zimbabwe
1703   Zimbabwe
```

```
[1704 rows x 1 columns]
```

```
[41]: print(df)
```

```
country continent year lifeExp pop gdpPercap
0    Afghanistan Asia 1952 28.001 8425333 779.445314
1    Afghanistan Asia 1957 30.332 9240934 820.853030
2    Afghanistan Asia 1962 31.997 10267083 853.100710
3    Afghanistan Asia 1967 34.020 11537966 836.197138
4    Afghanistan Asia 1972 36.088 13079460 739.981106
...
1699   Zimbabwe Africa 1987 62.351 9216418 706.157306
1700   Zimbabwe Africa 1992 60.377 10704340 693.420786
1701   Zimbabwe Africa 1997 46.809 11404948 792.449960
1702   Zimbabwe Africa 2002 39.989 11926563 672.038623
1703   Zimbabwe Africa 2007 43.487 12311143 469.709298
```

```
[1704 rows x 6 columns]
```

```
[43]: print(df.loc[0]) # 첫번째 행출력
```

```
country    Afghanistan
continent   Asia
year       1952
lifeExp    28.001
pop        8425333
gdpPercap  779.445314
Name: 0, dtype: object
```

```
[45]: print(df.loc[99]) # 100번째 행출력
```

```
country    Bangladesh
continent   Asia
year       1967
lifeExp    43.453
pop        62821884
gdpPercap  721.186086
Name: 99, dtype: object
```

```
[49]: number_of_row=df.shape[0] # 행개수 구하기
last_row_index=number_of_row-1 # 마지막행 인덱스 구하기
print(df.loc[last_row_index]) # 마지막행의 인덱스로 데이터 추출
```

```
country    Zimbabwe
continent   Africa
year       2007
lifeExp    43.487
pop        12311143
gdpPercap  469.709298
Name: 1703, dtype: object
```

```
[51]: print(df.tail(n=1)) # 마지막행 추출
```

```
country continent year lifeExp pop gdpPercap
1703   Zimbabwe Africa 2007 43.487 12311143 469.709298
```

```
[53]: print(df.loc[[0,99,999]]) # 첫번째 100번째 1000번째 데이터 추출
```

```
country continent year lifeExp pop gdpPercap
0    Afghanistan Asia 1952 28.001 8425333 779.445314
99   Bangladesh Asia 1967 43.453 62821884 721.186086
999  Mongolia Asia 1967 51.253 1149500 1226.041130
```

```
*[35]: print(df.iloc[-1]) # 마지막 행
```

```
country    Zimbabwe
continent   Africa
year       2007
lifeExp    43.487
pop        12311143
gdpPercap  469.709298
Name: 1703, dtype: object
```

```
[37]: subset=df.loc[:,['year','pop']]
print(subset) # :,[열]] 은 특정 열의 모든 행 출력
```

```
   year  pop
0  1952  8425333
1  1957  9240934
2  1962  10267083
3  1967  11537966
4  1972  13079460
...    ...
1699 1987  9216418
1700 1992  10704340
1701 1997  11404948
1702 2002  11926563
1703 2007  12311143
```

[1704 rows x 2 columns]

```
[39]: subset=df.iloc[:,[2,4,-1]] # 3,5번째와 마지막(-1) 열 데이터의 추출 합니다.
print(subset)
```

```
   year  pop  gdpPercap
0  1952  8425333  779.445314
1  1957  9240934  820.853030
2  1962  10267083  853.100710
3  1967  11537966  836.197138
4  1972  13079460  739.981106
...    ...
1699 1987  9216418  706.157306
1700 1992  10704340  693.420786
1701 1997  11404948  792.449960
1702 2002  11926563  672.038623
1703 2007  12311143  469.709298
```

[1704 rows x 3 columns]

```
[43]: small_range=list(range(5))
print(small_range)
```

[0, 1, 2, 3, 4]

```
[45]: subset=df.iloc[:,small_range]# 리스트를 사용해서 데이터프레임에서 열을 추출함
print(subset)
```

```
   country continent  year  lifeExp  pop
0  Afghanistan    Asia  1952    28.801  8425333
1  Afghanistan    Asia  1957    30.332  9240934
2  Afghanistan    Asia  1962    31.997  10267083
3  Afghanistan    Asia  1967    34.020  11537966
4  Afghanistan    Asia  1972    36.088  13079460
...    ...
1699  Zimbabwe    Africa  1987    62.351  9216418
1700  Zimbabwe    Africa  1992    60.377  10704340
1701  Zimbabwe    Africa  1997    46.809  11404948
1702  Zimbabwe    Africa  2002    39.989  11926563
1703  Zimbabwe    Africa  2007    43.487  12311143
```

[1704 rows x 5 columns]

```
[47]: small_range=list(range(3,6)) # 3이상 6미만
print(small_range)
```

[3, 4, 5]

```
[53]: subset=df.iloc[:,small_range]# 리스트를 사용해서 데이터프레임에서 열을 추출함
print(subset)
```

```
   country continent  year  lifeExp  pop
0  Afghanistan    Asia  1952    28.801  8425333
1  Afghanistan    Asia  1957    30.332  9240934
2  Afghanistan    Asia  1962    31.997  10267083
3  Afghanistan    Asia  1967    34.020  11537966
4  Afghanistan    Asia  1972    36.088  13079460
...    ...
1699  Zimbabwe    Africa  1987    62.351  9216418
1700  Zimbabwe    Africa  1992    60.377  10704340
1701  Zimbabwe    Africa  1997    46.809  11404948
1702  Zimbabwe    Africa  2002    39.989  11926563
1703  Zimbabwe    Africa  2007    43.487  12311143
```

[1704 rows x 5 columns]

```
[57]: small_range=list(range(0,6,2))
subset=df.iloc[:,small_range]
print(subset)
```

	country	year	pop
0	Afghanistan	1952	8425333
1	Afghanistan	1957	9240934
2	Afghanistan	1962	10267083
3	Afghanistan	1967	11537966
4	Afghanistan	1972	13079460
...
1699	Zimbabwe	1987	9216418
1700	Zimbabwe	1992	10704340
1701	Zimbabwe	1997	11404948
1702	Zimbabwe	2002	11926563
1703	Zimbabwe	2007	12311143

[1704 rows x 3 columns]

```
[59]: print(df.columns)

Index(['country', 'continent', 'year', 'lifeExp', 'pop', 'gdpPercap'], dtype='object')
```

```
[61]: subset=df.iloc[:, :3]
print(subset)
```

	country	continent	year
0	Afghanistan	Asia	1952
1	Afghanistan	Asia	1957
2	Afghanistan	Asia	1962
3	Afghanistan	Asia	1967
4	Afghanistan	Asia	1972
...
1699	Zimbabwe	Africa	1987
1700	Zimbabwe	Africa	1992
1701	Zimbabwe	Africa	1997
1702	Zimbabwe	Africa	2002
1703	Zimbabwe	Africa	2007

[1704 rows x 3 columns]

```
[63]: subset=df.iloc[:, 0:6:2]
print(subset)
```

	country	year	pop
0	Afghanistan	1952	8425333
1	Afghanistan	1957	9240934
2	Afghanistan	1962	10267083
3	Afghanistan	1967	11537966
4	Afghanistan	1972	13079460
...
1699	Zimbabwe	1987	9216418
1700	Zimbabwe	1992	10704340
1701	Zimbabwe	1997	11404948
1702	Zimbabwe	2002	11926563
1703	Zimbabwe	2007	12311143

[1704 rows x 3 columns]

```
[65]: print(df.loc[42,'country']) # country 열에서 행이름이 42인거를 추출 할수있음

Angola
```

```
[67]: print(df.iloc[42,0])

Angola
```

```
[69]: print(df.loc[[0,99,999],['country','lifeExp','gdpPercap']])
```

	country	lifeExp	gdpPercap
0	Afghanistan	28.801	779.445314
99	Bangladesh	43.453	721.186086
999	Mongolia	51.253	1226.041130

```
*[71]: print(df.groupby('year')['lifeExp'].mean()) # 연도별 그룹화 lifeExp의 선택 평균 계산
```

```
year
1952    49.057620
1957    51.507401
1962    53.609249
1967    55.678290
1972    57.647386
1977    59.570157
1982    61.533197
1987    63.212613
1992    64.160338
1997    65.014676
2002    65.694923
2007    67.007423
Name: lifeExp, dtype: float64
```

```
*[77]: multi_group_var=df.groupby(['year','continent'])[['lifeExp','gdpPercap']].mean() # 2 이상 그룹화
print(multi_group_var)
```

```
year continent lifeExp gdpPercap
1952 Africa 39.135500 1252.572466
Americas 53.279840 4079.062552
Asia 46.314394 5195.484004
Europe 64.408500 5661.057435
Oceania 69.255000 10298.085650
1957 Africa 41.266346 1385.236062
Americas 55.960280 4616.043733
Asia 49.318544 5787.732940
Europe 66.703067 6963.012816
Oceania 70.295000 11598.522455
1962 Africa 43.319442 1598.078825
Americas 58.398760 4901.541870
Asia 51.563223 5729.369625
Europe 68.539233 8365.486814
Oceania 71.085000 12696.452430
1967 Africa 45.334538 2050.363001
Americas 60.410920 5668.253496
Asia 54.663640 5971.179374
Europe 69.737600 10143.823757
Oceania 71.310000 14495.021790
1972 Africa 47.450942 2339.615674
Americas 62.394920 6491.334139
Asia 57.319269 8187.468699
Europe 70.775033 12479.575246
Oceania 71.910000 16417.333380
```

```
*[79]: print(df.groupby('continent')['country'].nunique()) # 그룹화한 데이터 개수 구하기
```

```
continent
Africa 52
Americas 25
Asia 33
Europe 30
Oceania 2
Name: country, dtype: int64
```

```
[83]: print(df.groupby('continent')['country'].value_counts()) # 지정된 열이나 행의 개수(빈도수) 구함
```

```
continent country
Africa Algeria 12
Angola 12
Benin 12
Botswana 12
Burkina Faso 12
..
Europe Switzerland 12
Turkey 12
United Kingdom 12
Oceania Australia 12
New Zealand 12
Name: count, Length: 142, dtype: int64
```

```
[89]: global_yearly_life_expendancy=df.groupby('year')['lifeExp'].mean()
print(global_yearly_life_expendancy)
```

```
year
1952    49.057620
1957    51.507401
1962    53.609249
1967    55.678290
1972    57.647386
1977    59.570157
1982    61.533197
1987    63.212613
1992    64.160338
1997    65.014676
2002    65.694923
2007    67.007423
Name: lifeExp, dtype: float64
```

```
[91]: global_yearly_life_expendancy.plot()
```

```
[91]: <Axes: xlabel='year'>
```

