



## 5. pandas 기본 사용법



데이터를 모다 쉽게 처리하고 데이터간의  
결합, 연산 동작을 손쉽게 할 수 있는 pandas  
라이브러리의 주요 기능을 학습



## I. pandas 란?

- 데이터 분석을 용이하게 할 수 있는 패키지
- Pandas에서 가장 많이 사용하는 자료구조 - **Series, DataFrame**

## why pandas?

이러한 구조의 데이터를 어떻게  
읽어들여야할까요?  
튜플?, 리스트?, 아니면, 사전?



이름	지역	나이
홍길동	서울	23
장길산	인천	30
임꺽정	서울	29
:	:	:

홍길동 – {지역 :서울 ,나이 -23}

장길산 – {지역 :인천 ,나이 -30}

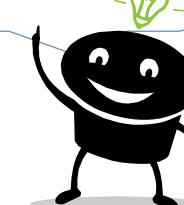
임꺽정 – {지역 :서울 ,나이 -29}

```
In [3]: mydata={'홍길동':{'지역':'서울','나이':23}, '임꺽정':{'지역':'인천','나이':30}, '장길산':{'지역':'서울','나이':29}}
mydata
Out[3]: {'임꺽정': {'나이': 30, '지역': '인천'}, '장길산': {'나이': 29, '지역': '서울'}, '홍길동': {'나이': 23, '지역': '서울'}}
```

지역별 거주인원, 평균 나이 등  
을 분석하려면 휴~ 프로그램 코  
딩이 길어지겠는데?



그렇게 복잡할 필요가 없어요  
^^pandas를 이용하면 아주 간단  
하게 분석할 수 있는걸요^^



## ◆ pandas 가볍게 사용하기

In [1]: `import pandas as pd`

### I. pandas 라이브러리 읽어오는 작업

In [2]: `df=pd.read_excel("http://qrc.depaul.edu/excel_files/presidents.xls")  
df`

Out [2]:

	President	Years in office	Year first inaugurated	Age at inauguration	State elected from	# of electoral votes	# of popular votes	National total votes	Total electoral votes	Rating points	Political Party	Occupation	College	% elec
0	George Washington	8	1789	57	Virginia	69	NA()	NA()	69	842.0	None	Planter	None	100.00
1	John Adams	4	1797	61	Massachusetts	132	NA()	NA()	139	598.0	Federalist	Lawyer	Harvard	94.96
2	Thomas Jefferson	8	1801	57	Virginia	73	NA()	NA()	137	711.0	Democratic-Republican	Planter, Lawyer	William and Mary	53.28
3	James Madison	8	1809	57	Virginia	122	NA()	NA()	176	567.0	Democratic-Republican	Lawyer	Princeton	69.31
4	James Monroe	8	1817	58	Virginia	183	NA()	NA()	221	602.0	Democratic-Republican	Lawyer	William and Mary	82.80
5	John Quincy Adams	4	1825	57	Massachusetts	84	NA()	NA()	261	564.0	Democratic-Republican	Lawyer	Harvard	32.18
6	Andrew Jackson	8	1829	61	Tennessee	178	642553	1148018	261	632.0	Democrat	Lawyer	None	68.19
7	Martin Van Buren	4	1837	54	New York	170	764176	1503534	294	429.0	Democrat	Lawyer	None	57.82

2. 분석할 파일이 엑셀파일이라면 pandas의 `read_excel`함수를 이용하면 이렇게 불러와서 데이터프레임형태의 구조로 변경을 한답니다.

3. 이 많은 데이터 항목 중 'political party' 열에 대한 빈도분석을 하고 싶어요. pandas가 제공하는 몇몇 메소드 만으로도 간단하게 원하는 결과를 얻을 수 있네요.

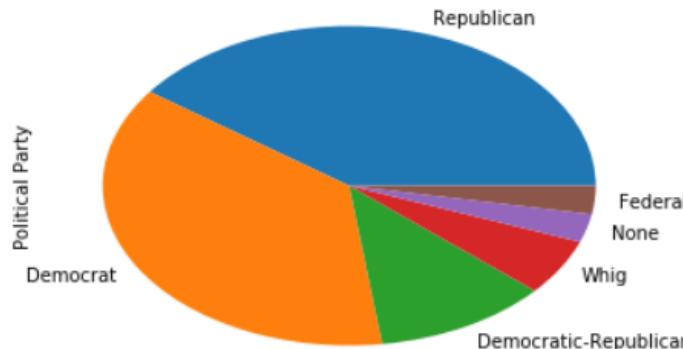
```
In [6]: df['Political Party'].value_counts()
```

```
Out[6]: Republican      14  
Democrat        13  
Democratic-Republican    4  
Whig            2  
None             1  
Federalist       1  
Name: Political Party, dtype: int64
```

4. 위 결과를 시각화 한다면 더 좋겠죠? matplotlib 라이브러리를 간단하게 결합하여 사용할 수 있어요.

```
In [7]: %matplotlib inline  
df['Political Party'].value_counts().plot(kind='pie')
```

```
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0xa1d3eb8>
```



pandas를 이용하면 정말 쉽게 원하는 분석을 할 수 있을거 같죠?



## ◆ Series

- ▶ 배열 구조와 같은 자료구조로써 '색인 - 값' 형태를 취함

### Series 생성 및 접근방법

```
In [1]: from pandas import Series, DataFrame
```

```
In [2]: age_in=Series([25,32,37,29,30])  
age_in
```

I. list형태의 데이터를 입력받아 Series 객체 생성

```
Out[2]: 0    25  
1    32  
2    37  
3    29  
4    30  
dtype: int64
```



인덱스 지정하지 않아도 자동으로 생성됨

```
In [3]: age_in.index  인덱스 정보는 index속성으로 확인
```

```
Out[3]: RangeIndex(start=0, stop=5, step=1)
```

```
In [5]: age_in.values  배열 값 정보는 values 속성으로 확인
```

```
Out[5]: array([25, 32, 37, 29, 30], dtype=int64)
```

```
In [6]: age_in=Series([25,32,37,29,30],index=['1번','2번','3번','4번','5번'])  
age_in
```

이렇게 index를 직접 지정할 수 있어요.

```
Out[6]: 1번    25  
2번    32  
3번    37  
4번    29  
5번    30  
dtype: int64
```

## ◆ DataFrame

- 엑셀문서처럼 여러 타입의 데이터를 저장할 수 있는 자료구조

### DataFrame생성 및 접근방법

```
In [1]: from pandas import Series,DataFrame
```

```
In [2]: mydata={'irum':['홍길동','장길산','임꺽정'],'addr':['서울','인천','서울'],'age':[23,30,29]}  
df=DataFrame(mydata)  
df
```

Out[2]:

	addr	age	irum
0	서울	23	홍길동
1	인천	30	장길산
2	서울	29	임꺽정

컬럼은 자동으로 정렬되어 표시

irum , age, addr 순서로  
표시할 순 없을까?

Series처럼 인덱스 자동으로 생성됨

```
In [3]: df1=DataFrame(mydata,columns=['irum','age','addr'],index=['no.1','no.2','no.3'])  
df1
```

Out[3]:

	irum	age	addr
no.1	홍길동	23	서울
no.2	장길산	30	인천
no.3	임꺽정	29	서울

컬럼 지정

인덱스 지정



In [4]: `df1['irum']`

컬럼명을 지정하여 해당 데이터에 접근

Out [4]:  
no.1 흥길동  
no.2 장길산  
no.3 임꺽정  
Name: irum, dtype: object

In [7]: `df1['point']=3`

컬럼을 추가하여 값을 대입할 수 있어요.

Out [7]:

	irum	age	addr	point
no.1	흥길동	23	서울	3
no.2	장길산	30	인천	3
no.3	임꺽정	29	서울	3

In [23]: `mydata=DataFrame({'홍길동':{'지역':'서울','age':23}, '임꺽정':{'지역':'인천','age':30}, '장길산':{'지역':'서울','age':29}})`

Out [23]:

컬럼

	홍길동	장길산	임꺽정
age	30	29	23
지역	서울	서울	인천

중첩된 딕셔너리를 이용한 데이터 프레임 생성



In [26]: `mydata.T`

행/열 구조 변경

Out [26]:

	age	지역
임꺽정	30	인천
장길산	29	서울
홍길동	23	서울