■ 행(row) / 열(column) 조회하기

● 데이터 불러오기

```
df = pd.read_csv('data/emp.txt')
df
```

	empno	ename	job	sal	comm	deptno
0	7369	smith	clerk	800	NaN	20
1	7499	allen	salesman	1600	300.0	30
2	7521	ward	salesman	1250	500.0	30
3	7566	jones	manager	2975	NaN	20
4	7654	martin	salesman	1250	1400.0	30
5	7698	blake	manager	2850	NaN	30
6	7782	clark	manager	2450	NaN	10
7	7788	scott	analyst	3000	NaN	20
8	7839	king	president	5000	NaN	10
9	7844	turner	salesman	1500	0.0	30
10	7876	adams	clerk	1100	NaN	20
11	7900	james	clerk	950	NaN	30
12	7902	ford	analyst	3000	NaN	20
13	7934	miller	clerk	1300	NaN	10

■ 행(row) / 열(column) 조회하기

● 상위 데이터 가져오기

– 기본 5개

```
df.head()
```

	empno	ename	job	sal	comm	deptno
0	7369	smith	clerk	800	NaN	20
1	7499	allen	salesman	1600	300.0	30
2	7521	ward	salesman	1250	500.0	30
3	7566	jones	manager	2975	NaN	20
4	7654	martin	salesman	1250	1400.0	30

– 개수 지정

```
df.head(3)
```

	empno	ename	job	sal	comm	deptno
0	7369	smith	clerk	800	NaN	20
1	7499	allen	salesman	1600	300.0	30
2	7521	ward	salesman	1250	500.0	30

■ 행(row) / 열(column) 조회하기

● 하위 데이터 가져오기

– 기본 5개

```
df.tail()
```

	empno	ename	job	sal	comm	deptno
9	7844	turner	salesman	1500	0.0	30
10	7876	adams	clerk	1100	NaN	20
11	7900	james	clerk	950	NaN	30
12	7902	ford	analyst	3000	NaN	20
13	7934	miller	clerk	1300	NaN	10

– 개수 지정

```
df.tail(3)
```

	empno	ename	job	sal	comm	deptno
11	7900	james	clerk	950	NaN	30
12	7902	ford	analyst	3000	NaN	20
13	7934	miller	clerk	1300	NaN	10

■ 행(row) / 열(column) 조회하기

● 기본 인덱싱 (열)

– 단일 지정

```
df['empno'].head()
```

```
0    7369
1    7499
2    7521
3    7566
4    7654
Name: empno, dtype: int64
```

– 복수 지정

```
df[['empno']].head()
```

```
df[['empno', 'ename', 'job']].head()
```

	empno
0	7369
1	7499
2	7521
3	7566
4	7654

	empno	ename	job
0	7369	smith	clerk
1	7499	allen	salesman
2	7521	ward	salesman
3	7566	jones	manager
4	7654	martin	salesman

■ 행(row) / 열(column) 조회하기

● 범위 인덱싱 (행)

- 4 ~ 6 행

```
df[3:6]
```

	empno	ename	job	sal	comm	deptno
3	7566	jones	manager	2975	NaN	20
4	7654	martin	salesman	1250	1400.0	30
5	7698	blake	manager	2850	NaN	30

- 11 ~ 마지막 앞 데이터

```
df[10:-1]
```

	empno	ename	job	sal	comm	deptno
10	7876	adams	clerk	1100	NaN	20
11	7900	james	clerk	950	NaN	30
12	7902	ford	analyst	3000	NaN	20

■ 행(row) / 열(column) 조회하기

● 행 지정 조회 (loc)

– 4행

```
df.loc[3]

empno         7566
ename        jones
job        manager
sal           2975
comm           NaN
deptno          20
Name: 3, dtype: object
```

```
df.loc[[3]]
```

	empno	ename	job	sal	comm	deptno
3	7566	jones	manager	2975	NaN	20

– 4, 7, 10행

```
df.loc[[3, 6, 9]]
```

	empno	ename	job	sal	comm	deptno
3	7566	jones	manager	2975	NaN	20
6	7782	clark	manager	2450	NaN	10
9	7844	turner	salesman	1500	0.0	30

■ 행(row) / 열(column) 조회하기

● 행 / 열 범위 조회

■ 행(row) / 열(column) 조회하기

● 특정 열의 값으로 조회

– 급여(sal) 2500 초과

```python
df[df.sal > 2500]
```

	empno	ename	job	sal	comm	deptno
3	7566	jones	manager	2975	NaN	20
5	7698	blake	manager	2850	NaN	30
7	7788	scott	analyst	3000	NaN	20

– 직무(job) manager

```python
df[df.job == 'manager']
```

	empno	ename	job	sal	comm	deptno
3	7566	jones	manager	2975	NaN	20
5	7698	blake	manager	2850	NaN	30
6	7782	clark	manager	2450	NaN	10

■ 행(row) / 열(column) 조회하기

● 특정 열의 값으로 조회

- 급여(sal) 2500 초과이면서 직무(job)가 manager

```
df[(df.sal > 2500) & (df.job == 'manager')]
```

	empno	ename	job	sal	comm	deptno
3	7566	jones	manager	2975	NaN	20
5	7698	blake	manager	2850	NaN	30

- query 함수

```
df.query('sal > 2500 & job == "manager"')
```

	empno	ename	job	sal	comm	deptno
3	7566	jones	manager	2975	NaN	20
5	7698	blake	manager	2850	NaN	30
6	7782	clark	manager	2450	NaN	10

■ 행(row) / 열(column) 조회하기

● 중복 값 제거 후 조회

```
df.job.unique()
```

```
array(['clerk', 'salesman', 'manager', 'analyst', 'president'],
      dtype=object)
```

● 중복 요소와 개수 확인

```
df.job.value_counts()
```

```
clerk       4
salesman    4
manager     3
analyst     2
president   1
Name: job, dtype: int64
```

■ 행(row) / 열(column) 조회하기

● 중복 요소 확인

```
df.duplicated(['job'], keep='first')

0     False
1     False
2      True
3     False
4      True
5      True
6      True
7     False
8     False
9      True
10     True
11     True
12     True
13     True
dtype: bool
```

```
df.duplicated(['job'], keep='last')

0      True
1      True
2      True
3      True
4      True
5      True
6     False
7      True
8     False
9     False
10     True
11     True
12    False
13    False
dtype: bool
```

```
df.duplicated(['job'], keep=False)

0      True
1      True
2      True
3      True
4      True
5      True
6      True
7      True
8     False
9      True
10     True
11     True
12     True
13     True
dtype: bool
```

	empno	ename	job	sal	comm	deptno
0	7369	smith	clerk	800	NaN	20
1	7499	allen	salesman	1600	300.0	30
2	7521	ward	salesman	1250	500.0	30
3	7566	jones	manager	2975	NaN	20
4	7654	martin	salesman	1250	1400.0	30
5	7698	blake	manager	2850	NaN	30
6	7782	clark	manager	2450	NaN	10
7	7788	scott	analyst	3000	NaN	20
8	7839	king	president	5000	NaN	10
9	7844	turner	salesman	1500	0.0	30
10	7876	adams	clerk	1100	NaN	20
11	7900	james	clerk	950	NaN	30
12	7902	ford	analyst	3000	NaN	20
13	7934	miller	clerk	1300	NaN	10

■ 행(row) / 열(column) 조회하기

● 숫자가 아닌 요소 확인

```
df.comm.isna()
```

```
0       True
1      False
2      False
3       True
4      False
5       True
6       True
7       True
8       True
9      False
10      True
11      True
12      True
13      True
Name: comm, dtype: bool
```

● 값이 없는 요소 확인

```
df.comm.isnull()
```

```
0       True
1      False
2      False
3       True
4      False
5       True
6       True
7       True
8       True
9      False
10      True
11      True
12      True
13      True
Name: comm, dtype: bool
```

■ 행(row) / 열(column) 조회하기

　● 데이터 불러오기 (열 범위 인덱싱)

```
df2 = pd.read_csv('data/emp.txt', header=None)
df2
```

	0	1	2	3	4	5
0	empno	ename	job	sal	comm	deptno
1	7369	smith	clerk	800	NaN	20
2	7499	allen	salesman	1600	300	30
3	7521	ward	salesman	1250	500	30
4	7566	jones	manager	2975	NaN	20
5	7654	martin	salesman	1250	1400	30
6	7698	blake	manager	2850	NaN	30
7	7782	clark	manager	2450	NaN	10
8	7788	scott	analyst	3000	NaN	20
9	7839	king	president	5000	NaN	10
10	7844	turner	salesman	1500	0	30
11	7876	adams	clerk	1100	NaN	20
12	7900	james	clerk	950	NaN	30
13	7902	ford	analyst	3000	NaN	20
14	7934	miller	clerk	1300	NaN	10

■ 행(row) / 열(column) 조회하기

● 열 범위 인덱싱 (iloc)

– 2 ~ 4행, 1 ~ 4열

```
df2.iloc[1:4, 0:-2]
```

	0	1	2	3
1	7369	smith	clerk	800
2	7499	allen	salesman	1600
3	7521	ward	salesman	1250

– 6 ~ 8행, 모든 열

```
df2.iloc[5:8, :]
```

	0	1	2	3	4	5
5	7654	martin	salesman	1250	1400	30
6	7698	blake	manager	2850	NaN	30
7	7782	clark	manager	2450	NaN	10

■ 행(row) / 열(column) 조회하기

● filter 함수

```
df.filter(items=['empno', 'ename', 'job', 'sal'])
```

	empno	ename	job	sal
0	7369	smith	clerk	800
1	7499	allen	salesman	1600
2	7521	ward	salesman	1250
3	7566	jones	manager	2975
4	7654	martin	salesman	1250
5	7698	blake	manager	2850
6	7782	clark	manager	2450
7	7788	scott	analyst	3000
8	7839	king	president	5000
9	7844	turner	salesman	1500
10	7876	adams	clerk	1100
11	7900	james	clerk	950
12	7902	ford	analyst	3000
13	7934	miller	clerk	1300

■ 행(row) / 열(column) 조회하기

● filter 함수

– a 가 포함되어 있는 열을 대상으로 조회

```
df.filter(like='a', axis=1)[0:5]
```

	ename	sal
0	smith	800
1	allen	1600
2	ward	1250
3	jones	2975
4	martin	1250

■ 행(row) / 열(column) 조회하기

● filter 함수

– e 로 시작하는 열을 대상으로 조회

```
df.filter(regex='^e')[0:5]
```

	empno	ename
0	7369	smith
1	7499	allen
2	7521	ward
3	7566	jones
4	7654	martin

– e 로 끝나는 열을 대상으로 조회

```
df.filter(regex='e$')[0:5]
```

	ename
0	smith
1	allen
2	ward
3	jones
4	martin