

■ DataFrame 합치기

● 기본 데이터

```
product_list = [  
    {'name': 'MOUSE', 'price': 100, 'company': 'A'},  
    {'name': 'SSD', 'price': 200, 'company': 'B'},  
    {'name': 'CPU', 'price': 300, 'company': 'C'}  
]  
df = pd.DataFrame(product_list, columns=['name', 'price', 'company'])  
df
```

	name	price	company
0	MOUSE	100	A
1	SSD	200	B
2	CPU	300	C

● 합쳐질 두번째 데이터

```
df2 = pd.DataFrame([  
    ['LAN', 100, 'A']  
], columns=['name', 'price', 'company'])  
df2
```

	name	price	company
0	LAN	100	A

■ DataFrame 합치기

● concat()

– index 중복

```
pd.concat([df, df2])
```

	name	price	company
0	MOUSE	100	A
1	SSD	200	B
2	CPU	300	C
0	LAN	100	A

– index 정렬

```
pd.concat([df, df2], ignore_index=True)
```

	name	price	company
0	MOUSE	100	A
1	SSD	200	B
2	CPU	300	C
3	LAN	100	A

■ DataFrame 합치기

● append()

– index 중복

```
df.append(df2)
```

	name	price	company
0	MOUSE	100	A
1	SSD	200	B
2	CPU	300	C
0	LAN	100	A

– index 정렬

```
df.append(df2, ignore_index=True)
```

	name	price	company
0	MOUSE	100	A
1	SSD	200	B
2	CPU	300	C
3	LAN	100	A

■ DataFrame 합치기

● 기본 데이터

```
df1 = pd.DataFrame([
    {'name': 'John', 'age': 20},
    {'name': 'Liam', 'age': 21},
    {'name': 'Noah', 'age': 22}
], columns=['name', 'age'])
df2 = pd.DataFrame([
    {'name': 'Logan', 'age': 23},
    {'name': 'Lucas', 'age': 24},
    {'name': 'Mason', 'age': 25}
], columns=['name', 'age'])
df3 = pd.DataFrame([
    {'job': 'developer', 'area': 'seoul'},
    {'job': 'driver', 'area': 'jeju'},
    {'job': 'chef', 'area': 'busan'},
], columns=['job', 'area'])
df4 = pd.DataFrame([
    {'job': 'salesman', 'area': 'gwangju'},
    {'job': 'analyst', 'area': 'cheonnan'},
    {'job': 'broadcaster', 'area': 'daejeon'},
], columns=['job', 'area'])
```

■ DataFrame 합치기

● concat()

– index 중복

```
pd.concat([df1, df2])
```

	name	age
0	John	20
1	Liam	21
2	Noah	22
0	Logan	23
1	Lucas	24
2	Mason	25

– index 정렬

```
pd.concat([df1, df2], ignore_index=True)
```

	name	age
0	John	20
1	Liam	21
2	Noah	22
3	Logan	23
4	Lucas	24
5	Mason	25

■ DataFrame 합치기

● append()

– index 중복

```
df1.append(df2)
```

	name	age
0	John	20
1	Liam	21
2	Noah	22
0	Logan	23
1	Lucas	24
2	Mason	25

– index 정렬

```
df1.append(df2, ignore_index=True)
```

	name	age
0	John	20
1	Liam	21
2	Noah	22
3	Logan	23
4	Lucas	24
5	Mason	25

■ DataFrame 합치기

● 열 합치기

– axis 지정

```
pd.concat([df1, df3], axis=1)
```

	name	age	job	area
0	John	20	developer	seoul
1	Liam	21	driver	jeju
2	Noah	22	chef	busan

– axis 미지정

```
pd.concat([df1, df3])
```

	age	area	job	name
0	20.0	NaN	NaN	John
1	21.0	NaN	NaN	Liam
2	22.0	NaN	NaN	Noah
0	NaN	seoul	developer	NaN
1	NaN	jeju	driver	NaN
2	NaN	busan	chef	NaN

■ DataFrame 합치기 (merge)

● merge

```
df1 = pd.DataFrame({  
    '고객번호': [1001, 1002, 1003, 1004, 1005, 1006, 1007],  
    '이름': ['둘리', '도우너', '또치', '길동', '희동', '마이콜', '영희']  
}, columns=['고객번호', '이름'])
```

	고객번호	이름
0	1001	둘리
1	1002	도우너
2	1003	또치
3	1004	길동
4	1005	희동
5	1006	마이콜
6	1007	영희

```
df2 = pd.DataFrame({  
    '고객번호': [1001, 1001, 1005, 1006, 1008, 1001],  
    '금액': [10000, 20000, 15000, 5000, 100000, 30000]  
}, columns=['고객번호', '금액'])
```

	고객번호	금액
0	1001	10000
1	1001	20000
2	1005	15000
3	1006	5000
4	1008	100000
5	1001	30000

■ DataFrame 합치기 (merge)

● 기본사용

```
pd.merge(df1, df2)
```

	고객번호	이름	금액
0	1001	둘리	10000
1	1001	둘리	20000
2	1001	둘리	30000
3	1005	희동	15000
4	1006	마이콜	5000

df1

	고객번호	이름
0	1001	둘리
1	1002	도우너
2	1003	또치
3	1004	길동
4	1005	희동
5	1006	마이콜
6	1007	영희

df2

	고객번호	금액
0	1001	10000
1	1001	20000
2	1005	15000
3	1006	5000
4	1008	100000
5	1001	30000

■ DataFrame 합치기 (merge)

● outer

```
pd.merge(df1, df2, how='outer')
```

	고객번호	이름	금액
0	1001	둘리	10000.0
1	1001	둘리	20000.0
2	1001	둘리	30000.0
3	1002	도우너	NaN
4	1003	또치	NaN
5	1004	길동	NaN
6	1005	희동	15000.0
7	1006	마이클	5000.0
8	1007	영희	NaN
9	1008	NaN	100000.0

df1

	고객번호	이름
0	1001	둘리
1	1002	도우너
2	1003	또치
3	1004	길동
4	1005	희동
5	1006	마이클
6	1007	영희

df2

	고객번호	금액
0	1001	10000
1	1001	20000
2	1005	15000
3	1006	5000
4	1008	100000
5	1001	30000

■ DataFrame 합치기 (merge)

● left

```
pd.merge(df1, df2, how='left')
```

	고객번호	이름	금액
0	1001	둘리	10000.0
1	1001	둘리	20000.0
2	1001	둘리	30000.0
3	1002	도우너	NaN
4	1003	또치	NaN
5	1004	길동	NaN
6	1005	희동	15000.0
7	1006	마이콜	5000.0
8	1007	영희	NaN

● right

```
pd.merge(df1, df2, how='right')
```

	고객번호	이름	금액
0	1001	둘리	10000
1	1001	둘리	20000
2	1001	둘리	30000
3	1005	희동	15000
4	1006	마이콜	5000
5	1008	NaN	100000

■ DataFrame 합치기 (merge)

● 동일한 컬럼명이 2개 이상인 경우

```
df1 = pd.DataFrame({  
    '고객명': ['준향', '준향', '몽룡'],  
    '날짜': ['2018-01-01', '2018-01-02', '2018-01-01'],  
    '데이터': ['20000', '30000', '100000']})
```

	고객명	날짜	데이터
0	준향	2018-01-01	20000
1	준향	2018-01-02	30000
2	몽룡	2018-01-01	100000

```
df2 = pd.DataFrame({  
    '고객명': ['준향', '몽룡'],  
    '데이터': ['여자', '남자']})
```

	고객명	데이터
0	준향	여자
1	몽룡	남자



```
pd.merge(df1, df2, on='고객명')
```

	고객명	날짜	데이터_x	데이터_y
0	준향	2018-01-01	20000	여자
1	준향	2018-01-02	30000	여자
2	몽룡	2018-01-01	100000	남자

■ DataFrame 합치기 (merge)

● 동일한 컬럼명이 없는 경우

```
df1 = pd.DataFrame({  
    '이름': ['영희', '철수', '철수'],  
    '성적': [1, 2, 3]})  
df1
```

	이름	성적
0	영희	1
1	철수	2
2	철수	3

```
df2 = pd.DataFrame({  
    '성명': ['영희', '영희', '철수'],  
    '성적2': [4, 5, 6]})  
df2
```

	성명	성적2
0	영희	4
1	영희	5
2	철수	6



```
pd.merge(df1, df2, left_on='이름', right_on='성명')
```

	이름	성적	성명	성적2
0	영희	1	영희	4
1	영희	1	영희	5
2	철수	2	철수	6
3	철수	3	철수	6

■ 함수 적용 - apply

● 기본 데이터

```
list = [  
    {'join_date' : '2004-08-09'},  
    {'join_date' : '2009-11-19'},  
    {'join_date' : '2012-03-25'},  
    {'join_date' : '2013-09-01'},  
    {'join_date' : '2016-04-26'}  
]  
df = pd.DataFrame(list)  
df
```

	join_date
0	2004-08-09
1	2009-11-19
2	2012-03-25
3	2013-09-01
4	2016-04-26

■ 함수 적용 - apply

● 인자 1개

```
def get_year(date):  
    return date.split('-')[0]
```

```
df.join_date.apply(get_year)
```

0 2004

1 2009

2 2012

3 2013

4 2016

Name: join_date, dtype: object

```
df['year'] = df.join_date.apply(get_year)  
df
```

	join_date	year
0	2004-08-09	2004
1	2009-11-19	2009
2	2012-03-25	2012
3	2013-09-01	2013
4	2016-04-26	2016

■ 함수 적용 - apply

● 인자 2개

```
def get_continuous_year(year, now_year):  
    return now_year - int(year)  
  
df.year.apply(get_continuous_year, now_year=2018)
```

```
0    14  
1     9  
2     6  
3     5  
4     2  
Name: year, dtype: int64
```

```
df['continuous_year'] = df.year.apply(get_continuous_year, now_year=2018)  
df
```

	join_date	year	continuous_year
0	2004-08-09	2004	14
1	2009-11-19	2009	9
2	2012-03-25	2012	6
3	2013-09-01	2013	5
4	2016-04-26	2016	2

■ 함수 적용 - apply

● 기본 데이터

```
import pandas as pd
import numpy as np
```

```
arr = np.arange(1, 26).reshape(5, 5)
```

```
arr = arr + 0.4
```

```
arr[1] = arr[1] + 0.2
```

```
arr
```

```
array([[ 1.4,  2.4,  3.4,  4.4,  5.4],
       [ 6.6,  7.6,  8.6,  9.6, 10.6],
       [11.4, 12.4, 13.4, 14.4, 15.4],
       [16.4, 17.4, 18.4, 19.4, 20.4],
       [21.4, 22.4, 23.4, 24.4, 25.4]])
```

■ 함수 적용 - apply

● 외부 함수 적용 (Numpy around)

```
df = pd.DataFrame(arr)
df
```

	0	1	2	3	4
0	1.4	2.4	3.4	4.4	5.4
1	6.6	7.6	8.6	9.6	10.6
2	11.4	12.4	13.4	14.4	15.4
3	16.4	17.4	18.4	19.4	20.4
4	21.4	22.4	23.4	24.4	25.4

```
df.apply(np.around)
```

	0	1	2	3	4
0	1.0	2.0	3.0	4.0	5.0
1	7.0	8.0	9.0	10.0	11.0
2	11.0	12.0	13.0	14.0	15.0
3	16.0	17.0	18.0	19.0	20.0
4	21.0	22.0	23.0	24.0	25.0

■ 함수 적용 - map

● 기본 데이터

```
import pandas as pd

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

	name	role	salary
0	John	Director	9000.0
1	Liam	Manager	NaN
2	Noah	Staff	3300.0
3	Logan	Manager	5500.0
4	Lucas	Director	8500.0
5	Mason	Manager	6200.0
6	Oliver	Staff	NaN
7	Ethan	Staff	3800.0

■ 함수 적용 - map

● Dictionary 활용 데이터 변경 - 특정 값 변경

```
df['role'] = df.role.map({'Director': '부장', 'Manager': '과장', 'Staff': '사원'})  
df
```

	name	role	salary
0	John	부장	9000.0
1	Liam	과장	NaN
2	Noah	사원	3300.0
3	Logan	과장	5500.0
4	Lucas	부장	8500.0
5	Mason	과장	6200.0
6	Oliver	사원	NaN
7	Ethan	사원	3800.0