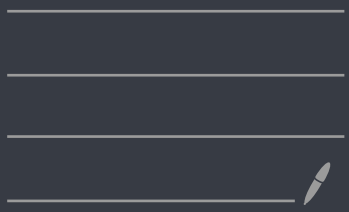


bind

Concat 메서드를 사용해서 위아래 (row), 좌우 (column)로 결합한다.

pd.concat([df1, df2], axis, ignore_index)

- ignore_index → default: False, True면 index가 새로 부여
- axis
  - 0 : (default) row bind 진행
  - 1 : column bind 진행

df1 :

	A	B	C
0	1	4	7
1	2	5	8
2	3	6	9

df2 :

	A	B	C
0	10	13	16
1	11	14	17
2	12	15	18

row bind

위아래로 합친다.

pd.concat([df1, df2], axis=0)

		A	B	C
	0	1	4	7
	1	2	5	8
	2	3	6	9
index가 겹침	0	10	13	16
	1	11	14	17
	2	12	15	18

pd.concat([df1, df2], axis=0, ignore_index=True)

	A	B	C
0	1	4	7
1	2	5	8
2	3	6	9
3	10	13	16
4	11	14	17
5	12	15	18

= pd.concat([df1, df2], axis=0).reset_index(drop=True)

Column bind

좌우로 결합한다.

pd.concat([df1, df2], axis = 1)

	A	B	C	A	B	C
0	1	4	7	10	13	16
1	2	5	8	11	14	17
2	3	6	9	12	15	18

pd.concat([df1, df2], axis = 1, ignore_index = True)

	0	1	2	3	4	5
0	1	4	7	10	13	16
1	2	5	8	11	14	17
2	3	6	9	12	15	18

→ index가 reset 되어 숫자로 할당됨

크기가 다른 bind

df1 :

	D	B	C
0	1	4	7
1	2	5	8
2	3	6	9
3	4	7	10

df2 :

	A	E	C	B
0	10	13	16	19
1	11	14	17	20
2	12	15	18	21

row bind

pd.concat([df1, df2], axis = 0, ignore_index = True)

	D	B	C	A	E
0	1	4	7	NaN	NaN
1	2	5	8	NaN	NaN
2	3	6	9	NaN	NaN
3	4	7	10	NaN	NaN
4	NaN	19	16	10	13
5	NaN	20	17	11	14
6	NaN	21	18	12	15

공통 column에 대해서는 모든값 존재
다른 column에 대해서는 NaN

column bind

```
pd.concat([df2, df1], axis = 1)
```

	A	E	C	B	D	B	C
0	10	13	16	19	1	4	7
1	11	14	17	20	2	5	8
2	12	15	18	21	3	6	9
3	NaN	NaN	NaN	NaN	4	7	10

→ 그냥 순서대로

df1, 2 의 최대 row 기준

point

bind는 무조건 DataFrame을 만들어준다.

중요한 것은 NaN이 가치를 지니는가이다.

type이 다른 bind

df1 :

	D	B	C
0	1	4	7
1	2	5	8
2	3	6	9
3	4	7	10

df2 :

	A	E	C	B
0	10	13	a	19
1	11	14	b	20
2	12	15	c	21

```
pd.concat([df1, df2], axis=0, ignore_index = True).info()
```

	D	B	C	A	E
0	1.0	4	7	NaN	NaN
1	2.0	5	8	NaN	NaN
2	3.0	6	9	NaN	NaN
3	4.0	7	10	NaN	NaN
4	NaN	19	a	10.0	13.0
5	NaN	20	b	11.0	14.0
6	NaN	21	c	12.0	15.0

float64 int64

☆ object로 결합된다.

⇒ 타입이 달라도 결합은 가능

중요한 기준은 변수명이다.

merge

특정 변수를 기준으로 병합한다.

pd.merge(df1, df2, on [, how])
↳ 기준이 되는 변수

df1 :

	name	age
0	철수	23
1	영이	34
2	John	19

df2 :

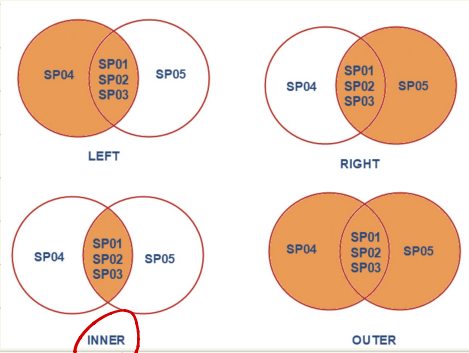
	name	sex
0	철수	M
1	영이	F
2	John	M

pd.merge(df1, df2, on = "name")
↳ name 변수의 값이 같은 것끼리 묶어라

	name	age	sex
0	철수	23	M
1	영이	34	F
2	John	19	M

df1.merge(df2, on = "name")도 같은 결과

merge의 종류



↳ default

df1 :

	name	age
0	철수	23
1	영이	34
2	John	19

df3 :

	name	weight
0	철수	55
1	영이	44

df2 :

	name	sex
0	철수	M
1	영이	F
2	John	M

df4 :

	name	height
0	철수	167
1	영이	175
2	홍민	183

inner

↳ default 옵션이다.

pd.merge(df1, df3, on="name")

	name	age	weight
0	철수	23	55
1	영이	34	44

left

↳ 첫 요소의 관측치는 다 가져옴, 두번째는 맞는것만 이어붙임

pd.merge(df1, df3, on="name", how="left")

	name	age	weight
0	철수	23	55
1	영이	34	44
2	John	19	NaN

right

↳ 두번째 요소의 관측치는 다 가져옴, 첫번째는 맞는 것만 이어붙임

pd.merge(df1, df4, on="name" how="right")

	name	age	height
0	철수	23	167
1	영이	34	175
2	홍민	NaN	183

right여도 변수순서 = 옵션순서 df1, df4

outer

↳ 있는 관측치 다 붙임

pd.merge(df1, df4, on="name" how="outer")

	name	age	height
0	철수	23.0	167.0
1	영이	34.0	175.0
2	John	19.0	NaN
3	흥민	NaN	183.0

변수의 이름이 다를때

ex)

df1 :

	이름	age
0	철수	23
1	영이	34

df2 :

	name	height
0	철수	167
1	영이	175

1. "이름" ⇒ "name"으로 변경 ⇒ df1.rename(colums={"이름":"name"}, inplace=True)
2. merge의 옵션 이용
   ⇒ pd.merge(df1, df2, left_on="이름", right_on="name")

merge 주의사항

1. Type이 같아야 한다.

bind priority : Type < column name

merge priority : Type > column name

ex) df5 = pd.DataFrame({
"name": ["철수", "영이", "흥민"],
"height": ["167", "175", 183]})
↳ 문자

pd.merge(df4, df5, on="height")

	name_x	height	name_y
0	흥민	183	흥민

2. 중복된 값이 존재하면 오히려 관측치 증가

ex)

df6 :

	name	age
0	철수	23
1	영이	34
2	John	19
3	철수	20
4	John	25

pd.merge(df1, df6, on = "name", how = "left")

↳ df1 기준 결합
But df5 row만큼 출력

	name	age_x	age_y
0	철수	23	23
1	철수	23	20
2	영이	34	34
3	John	19	19
4	John	19	25

3. 데이터가 조금이라도 틀리면 merge 안됨

ex) 특수기호, space 포함