

欠損値の行列を抽出

<元DataFrame>

	name	age	state	point	other
0	Alice	24	NY	NaN	NaN
1	NaN	NaN	CA	NaN	NaN
2	Charlie	NaN	CA	NaN	NaN
3	Dave	68	TX	70	NaN
4	Ellen	48	CA	88	etc
5	Frank	30	FK	NaN	NaN

※①において、逆に、`isnull()`ではない行だけを抽出する場合は

`df [ df[ 左のSeries ] == False ]` または

`df [ ~ df[ 左のSeries ] ]`

①特定の列に欠損値が含まれる行を抽出

`df['age'].isnull()`

```

0    False
1     True
2     True
3    False
4    False
5    False
Name: age, dtype: bool

```

`df[ 左のSeries ]`

	name	age	state	point	other
0	Alice	24	NY	NaN	NaN
1	NaN	NaN	CA	NaN	NaN
2	Charlie	NaN	CA	NaN	NaN
3	Dave	68	TX	70	NaN
4	Ellen	48	CA	88	etc
5	Frank	30	FK	NaN	NaN

②特定の行に欠損値が含まれる列を抽出

`df.iloc[2].isnull()`

```

name    False
age      True
state    False
point    True
other    True
Name: 2, dtype: bool

```

`df.loc[:, 左のSeries]`

	name	age	state	point	other
0	Alice	24	NY	NaN	NaN
1	NaN	NaN	CA	NaN	NaN
2	Charlie	NaN	CA	NaN	NaN
3	Dave	68	TX	70	NaN
4	Ellen	48	CA	88	etc
5	Frank	30	FK	NaN	NaN

(再掲)列ごとに1つでも欠損値を含むかを判定

`df.isnull().any()`

```

name    True
age      True
state    False
point    True
other    True
dtype: bool

```

欠損値が1つでも含まれる列を抽出

`df.loc[:, 左のSeries]`

	name	age	state	point	other
0	Alice	24	NY	NaN	NaN
1	NaN	NaN	CA	NaN	NaN
2	Charlie	NaN	CA	NaN	NaN
3	Dave	68	TX	70	NaN
4	Ellen	48	CA	88	etc
5	Frank	30	FK	NaN	NaN

(再掲)行ごとに1つでも欠損値を含むかを判定

`df.isnull().any(axis=1)`

```

0     True
1     True
2     True
3     True
4    False
5     True

```

欠損値が1つでも含まれる行を抽出

`df.loc[左のSeries]`

	name	age	state	point	other
0	Alice	24	NY	NaN	NaN
1	NaN	NaN	CA	NaN	NaN
2	Charlie	NaN	CA	NaN	NaN
3	Dave	68	TX	70	NaN
4	Ellen	48	CA	88	etc
5	Frank	30	FK	NaN	NaN

列ごとに欠損値の個数をカウント

`df.isnull().sum()`

```

name      1
age       2
state      0
point     4
other     5
dtype: int64

```

行ごとに欠損値の個数をカウント

`df.isnull().sum(axis=1)`

```

0      2
1      4
2      3
3      1
4      0
5      2

```

dtype:int64

<元DataFrame>

	name	age	state	point	other
0	Alice	24	NY	NaN	NaN
1	NaN	NaN	CA	NaN	NaN
2	Charlie	NaN	CA	NaN	NaN
3	Dave	68	TX	70	NaN
4	Ellen	48	CA	88	etc
5	Frank	30	FK	NaN	NaN

欠損値の総数をカウント

df.isnull().values

```
array([[False, False, False,  True,  True],
       [ True,  True, False,  True,  True],
       [False,  True, False,  True,  True],
       [False, False, False, False,  True],
       [False, False, False, False, False],
       [False, False, False,  True,  True]])
```



df.isnull().values.sum()

DataFrameのsum()と異なり、nd_arrayのsum()は全体の合計を出せる

12

列ごとに欠損値でない要素をカウント

df.count()

```
name      5
age       4
state     6
point     2
other     1      18
dtype: int64
```

**欠損値でない要素の総数をカウント**

df.count().sum()

18

欠損値の穴埋め**欠損値を共通の値で一律に穴埋め**

df.fillna(■)

穴埋めする値の例

df.mean(), df.median(), df.mode() など

前の値で置き換え

df.fillna(method='ffill') <-- front

後の値で置き換え

df.fillna(method='bfill') <-- back

欠損値を列ごとに異なる値で穴埋め

df.fillna({'name': '---', 'age': 25})