



8장 판다스 자료형

08-1 자료형 다루기

자료형 변환 — astype 메서드



카테고리 자료형인 tips['sex']를 astype 메서드를 이용하여 **문자열**로 변환

```
tips['sex_str'] = tips['sex'].astype(str)
```

```
print(tips.dtypes)
```

total_bill	float64
tip	float64
sex	category
smoker	category
day	category
time	category
size	int64
sex_str	object

dtype: object

잘못 입력한 데이터 처리하기



missing 처리된 1, 3, 5, 7행의 데이터 처리하기

```
tips_sub_miss = tips.head(10)
tips_sub_miss.loc[[1, 3, 5, 7], 'total_bill'] = 'missing'
print(tips_sub_miss)
```

	total_bill	tip	sex	smoker	day	time	size	sex_str
0	16.99	1.01	Female	No	Sun	Dinner	2	Female
1	missing	1.66	Male	No	Sun	Dinner	3	Male
2	21.01	3.50	Male	No	Sun	Dinner	3	Male

잘못 입력한 데이터



missing으로 인하여 total_bill(float)의 값이 문자열로 인식

```
print(tips_sub_miss.dtypes)
```

total_bill	object
tip	float64
sex	category
smoker	category
day	category
time	category
size	int64
sex_str	object
dtype: object	

잘못 입력한 데이터 처리하기



total_bill column을 float type으로 바꾸어보자.

```
1 | tips_sub_miss['total_bill'].astype(float)
```

잘못 입력한 데이터 처리하기 — to_numeric 메서드



errors 인자에 설정할 수 있는 값

- raise : 숫자로 변환할 수 없는 값이 있으면 오류 발생
- coerce : 숫자로 변환할 수 없는 값을 누락값으로 지정
- ignore : 아무 작업도 하지 않음

```
1 tips_sub_miss['total_bill'] = pd.to_numeric(  
2     tips_sub_miss['total_bill'],  
3     errors='ignore')  
4  
5 print(tips_sub_miss.dtypes)
```

total_bill	object
tip	float64
sex	category
smoker	category
day	category

잘못 입력한 데이터 처리하기 — to_numeric 메서드



errors = 'coerce' 숫자로 변환할 수 없는 값을 누락값으로 지정

```
1 tips_sub_miss['total_bill'] = pd.to_numeric(  
2     tips_sub_miss['total_bill'],  
3     errors='coerce')  
4  
5 print(tips_sub_miss.dtypes)
```

total_bill	float64
tip	float64

downcast float 64 → float32

```
1 tips_sub_miss['total_bill'] = pd.to_numeric( tips_sub_miss['total_bill'],  
2                                             errors='coerce',  
3                                             downcast='float')  
4  
5 print(tips_sub_miss.dtypes)
```

total_bill	float32
tip	float64

08-2 카테고리 자료형

카테고리 자료형?



‘카테고리’ : 유한한 범위의 값만 가질 수 특수한 자료형

- 용량과 속도 면에서 매우 효율적
- 주로 동일한 문자열이 반복되어 데이터를 구성할 때 사용

```
1 tips['sex'] = tips['sex'].astype('str')
2 print(tips.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 8 columns):
total_bill    244 non-null float64
tip           244 non-null float64
sex           244 non-null object
smoker        244 non-null category
day           244 non-null category
time          244 non-null category
size          244 non-null int64
sex_str       244 non-null object
dtypes: category(3), float64(2), int64(1), object(2)
memory usage: 10.7+ KB
None
```

```
1 tips['sex'] = tips['sex'].astype('category')
2 print(tips.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 8 columns):
total_bill    244 non-null float64
tip           244 non-null float64
sex           244 non-null category
smoker        244 non-null category
day           244 non-null category
time          244 non-null category
size          244 non-null int64
sex_str       244 non-null object
dtypes: category(4), float64(2), int64(1), object(1)
memory usage: 9.1+ KB
None
```