

# 빅데이터 실습

13주차 2차시

시계열 데이터 기초 [2]

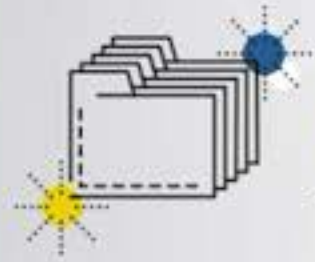
# 시계열 데이터



주요 함수 소개 및 활용 실습



# 시계열 데이터



## 학습개요

### 1/ datetime의 주요 함수 [1]

strptime() : 문자열을 datetime으로 변환

### 2/ datetime의 주요 함수 [2]

strftime() : datetime을 문자열로 변환

01

# datetime의 주요 함수 [ 1 ]





datetime의 주요 함수 [1]

**strptime()**

**문자열을  
datetime으로 변환**





datetime의 주요 함수 [1]

➔ **21/08/11**은 며칠일까?



2021년 8월 11일?

2011년 8월 21일?

11월 8일 21시?

8월 11일 21시?





datetime의 주요 함수 [1]

## ➤ 21/08/11은 며칠일까?



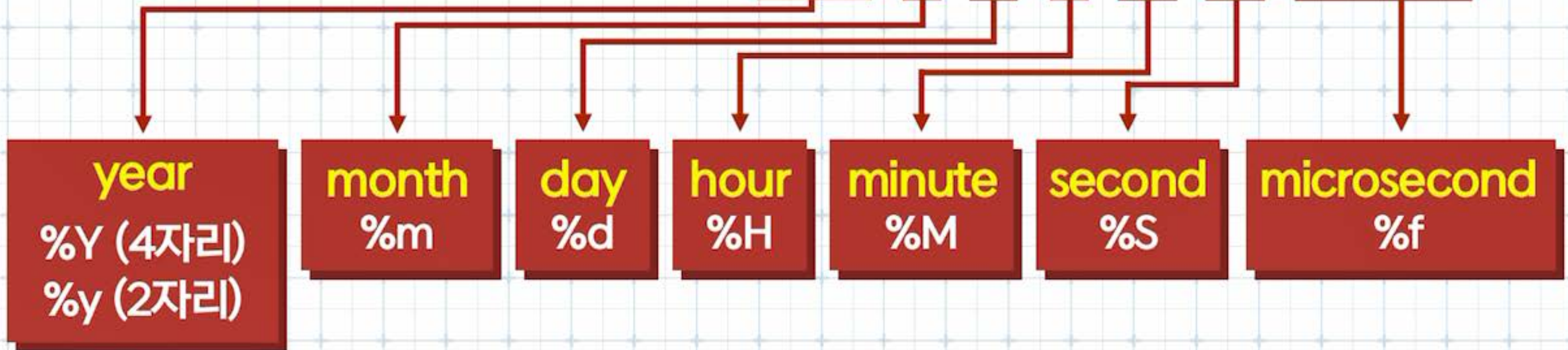




datetime의 주요 함수 [1]

## ➤ strftime (문자열, format)

Out [20]: datetime.datetime(2021, 8, 11, 0, 24, 30, 912787)





# strptime()으로 문자열을 날짜 타입으로 변환하기

## ◎ 문자열 선언



### 1-4.strptime()으로 문자열을 날짜 타입으로 변환하기

```
In [55]: 문자열 = '21/08/11'
```

```
In [56]: 문자열
```

```
Out[56]: '21/08/11'
```

```
In [20]: # 21년 8월 11일로 변환
```

```
In [21]: # 11년 8월 21일로 변환
```

```
In [22]: # 8월 11일 21시로 변환
```

날짜 정보로 보이지만, 실제 데이터 타입은 문자열로 되어 있습니다.

[실습] 아래 데이터에서 2020년 7월 데이터만 선택하기

# strptime()으로 문자열을 날짜 타입으로 변환하기

## ◎ 문자열을 datetime 객체로 변환

✓ 21년 8월 11일로 변환하기

### 1-4.strptime()으로 문자열을 날짜 타입으로 변환하기

In [55]: 문자열 = '21/08/11'

In [56]: 문자열

Out [56]: '21/08/11'

**strptime(문자열, format)**

In [20]: *# 21년 8월 11일로 변환*  
dt.datetime.strptime(문자열, '|')

In [21]: *# 11년 8월 21일로 변환*

In [22]: *# 8월 11일 21시로 변환*

[실습] 아래 df에서 2020년 7월 데이터만 선택하기



# strptime()으로 문자열을 날짜 타입으로 변환하기

## ◎ 문자열을 datetime 객체로 변환

✓ 11년 8월 21일로 변환하기

### 1-4.strptime()으로 문자열을 날짜 타입으로 변환하기

```
In [55]: 문자열 = '21/08/11'
```

```
In [56]: 문자열
```

```
Out[56]: '21/08/11'
```

```
In [57]: # 21년 8월 11일로 변환  
dt.datetime.strptime(문자열, '%y/%m/%d')
```

```
Out[57]: datetime.datetime(2021, 8, 11, 0, 0)
```

```
In [21]: # 11년 8월 21일로 변환  
dt.datetime.strptime(문자열, '%d/%m/%y')
```

```
In [22]: # 8월 11일 21시로 변환
```

# strptime( )으로 문자열을 날짜 타입으로 변환하기

## ◎ 문자열을 datetime 객체로 변환

✓ 8월 11일 21시로 변환하기

In [56]: 문자열

Out[56]: '21/08/11'

In [57]: # 21년 8월 11일로 변환

```
dt.datetime.strptime(문자열, '%y/%m/%d')
```

Out[57]: datetime.datetime(2021, 8, 11, 0, 0)

In [58]: # 11년 8월 21일로 변환

```
dt.datetime.strptime(문자열, '%d/%m/%y')
```

Out[58]: datetime.datetime(2011, 8, 21, 0, 0)

In [59]: # 8월 11일 21시로 변환

```
dt.datetime.strptime(문자열, '%H/%m/%d')
```

Out[59]: datetime.datetime(1900, 8, 11, 21, 0)

In [ ]:



# strptime( )으로 문자열을 날짜 타입으로 변환하기

## ◎ 문자열을 datetime 객체로 변환

✓ year을 2021로 업데이트

```
Out[58]: datetime.datetime(2011, 8, 21, 0, 0)
```

```
In [60]: # 8월 11일 21시로 변환  
dt1 = dt.datetime.strptime(문자열, '%H/%m/%d')
```

```
In [62]: # dt1의 year를 2021로 업데이트  
dt1.year = 2021
```

```
AttributeError                                Traceback (most recent call last)  
<ipython-input-62-018a4d0365b5> in <module>  
      1 # dt1의 year를 2021로 업데이트  
----> 2 dt1.year = 2021
```

```
AttributeError: attribute 'year' of 'datetime.date' objects is not writable
```

**Error 발생**

```
In [ ]:
```

# strptime( )으로 문자열을 날짜 타입으로 변환하기

## ◎ 문자열을 datetime 객체로 변환

✓ year을 2021로 업데이트

```
time(문자열, '%d/%m/%y')
```

```
Out[58]: datetime.datetime(2011, 8, 21, 0, 0)
```

```
In [60]: # 8월 11일 21시로 변환  
dt1 = dt.datetime.strptime(문자열, '%H/%m/%d')
```

```
In [62]: # dt1의 year를 2021로 업데이트  
#dt1.year = 2021  
dt1.replace(year = 2)
```

replace 실행

AttributeError Traceback (most recent call last)

<ipython-input-62-018a4d0365b5> in <module>

1 # dt1의 year를 2021로 업데이트

----> 2 dt1.year = 2021

AttributeError: attribute 'year' of 'datetime.date' objects is not writable

In [ ]:



## ◎ 문자열을 datetime 객체로 변환

time(문자열, '%d/%m/%y')

```
In [60]: # 8월 11일 21시로 변환
dt1 = dt.datetime.strptime(문자열, '%H/%m/%d')
```

```
Out[63]: datetime.datetime(2021, 8, 11, 21, 0)
```

In [ ]:

[실습] 아래 df에서 2020년 7월 데이터만 선택하기

```

10000 07 04| 10000 07 02| 10000 07 05| 10000 07 10| 10000 07

```

**datetime에서는 replace 함수를 통해서만 업데이트 가능합니다.**

} )

# strptime( )으로 문자열을 날짜 타입으로 변환하기

◎ df에서 2020년 7월 데이터만 선택하기



```
6 2020-07-28    69
7 2020-08-05    47
8 2020-08-10    62
9 2020-08-23    26
```

```
In [66]: df.set_index('date')['2020-07']
```

Out[66]:

	value
date	
2020-07-01	25
2020-07-02	20
2020-07-05	24
2020-07-10	33
2020-07-23	4
2020-07-24	45
2020-07-28	69
2020-08-05	47



# strftime( )으로 문자열을 날짜 타입으로 변환하기

© df에서 2020년 7월 데이터만 선택하기



```
KeyError                                Traceback (most recent call last)
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
    2894         try:
-> 2895             return self._engine.get_loc(casted_key)
    2896         except KeyError as err:
```

```
pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()
```

```
pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()
```

```
pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()
```

```
pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.get_item()
```

```
KeyError: '2020-07'
```

The above exception was the direct cause of the following exception:

```
KeyError                                Traceback (most recent call last)
<ipython-input-67-91a79c0cb655> in <module>
----> 1 df.set_index('date')['2020-07']
```

**Error 발생**

# strptime( )으로 문자열을 날짜 타입으로 변환하기

◎ df에서 2020년 7월 데이터만 선택하기

✓ 에러가 발생하는 이유

8	69
7	47
8	62
9	26

```
In [67]: df.set_index('date')['2020-07']
```

...

```
In [70]: # 위에서 배운대로 해도 정상적으로 동작하지 않음
sample.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 2 columns):
#   Column  Non-Null Count  Dtype  
---  -
0   date    500 non-null    datetime64[ns]
1   count   500 non-null    int64   
dtypes: datetime64[ns](1), int64(1)
memory usage: 7.9 KB
```



# strptime( )으로 문자열을 날짜 타입으로 변환하기

◎ df에서 2020년 7월 데이터만 선택하기

✓ 에러가 발생하는 이유

8	69	
7	2020-08-05	47
8	2020-08-10	62
9	2020-08-23	26

주의하기

✓ date 컬럼의 데이터 타입이 datetime이 아니라 문자열이므로 에러 발생

In [72]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 2 columns):
#   Column  Non-Null Count  Dtype  
---  -
0    date    10 non-null     object  
1    value    10 non-null     int64   
dtypes: int64(1), object(1)
memory usage: 288.0+ bytes
```

In [67]: df.set\_index('date')[['2020-07']]

In [70]: # 위에서 배운대로 해도 정상적으로 동작하지 않음

# strptime( )으로 문자열을 날짜 타입으로 변환하기

◎ df에서 2020년 7월 데이터만 선택하기

✓ **해결책**) date 컬럼의 데이터 타입을 datetime으로 변환

```
Data columns (total 2 columns):  
#   Column  Non-Null Count  Dtype  
---  ---  
0   date     500 non-null     datetime64[ns]  
1   count     500 non-null     int64  
dtypes: datetime64[ns](1), int64(1)  
memory usage: 7.9 KB
```

```
In [76]: # 해결책. df의 date컬럼의 타입을 datetime으로 변환  
df.date.apply(lambda X: dt.datetime.strptime(X, '%Y-%m-%d')).info()
```

```
Out[76]: 0    2020-07-01  
1    2020-07-02  
2    2020-07-05  
3    2020-07-10  
4    2020-07-23  
5    2020-07-24  
6    2020-07-28  
7    2020-08-05  
8    2020-08-10  
9    2020-08-23  
Name: date, dtype: datetime64[ns]
```



# strptime()으로 문자열을 날짜 타입으로 변환하기

◎ df에서 2020년 7월 데이터만 선택하기

✓ pandas.to\_datetime() 함수 사용

```
In [79]: # 해결책. df의 date컬럼의 타입을 datetime으로 변환  
# 1..strptime() 활용  
df['date2'] = df.date.apply(lambda X: dt.datetime.strptime(X, '%Y-%m-%d'))
```

```
In [81]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 10 entries, 0 to 9  
Data columns (total 3 columns):  
#   Column  Non-Null Count  Dtype  
---  ---  
0    date    10 non-null      object  
1    value    10 non-null      int64  
2    date2     10 non-null      datetime64[ns]  
dtypes: int64(1), object(1), datetime64[ns](1)  
memory usage: 1.0 KB
```

pd.to\_datetime 실행

```
In [ ]: pd.to_datetime(df.date, format = '%Y-%m-%d')
```

```
In [24]: # 7월 데이터만 조회
```

# strptime()으로 문자열을 날짜 타입으로 변환하기

◎ df에서 2020년 7월 데이터만 선택하기

✓ 7월 데이터만 조회

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 10 entries, 0 to 9  
Data columns (total 4 columns):  
#   Column  Non-Null Count  Dtype  
---  ---  
0   date    10 non-null     object  
1   value   10 non-null     int64  
2   date2    10 non-null     datetime64[ns]  
3   date3    10 non-null     datetime64[ns]  
dtypes: datetime64[ns](2), int64(1), object(1)  
memory usage: 448.0+ bytes
```

```
In [85]: # 7월 데이터만 조회  
df.set_index('date2')['2020']
```

Out[85]:

	date	value	date3
date2			
2020-07-01	2020-07-01	25	2020-07-01
2020-07-02	2020-07-02	20	2020-07-02



# strftime( )으로 문자열을 날짜 타입으로 변환하기

◎ df에서 2020년 7월 데이터만 선택하기

✓ 7월 데이터만 조회

```
df.set_index('date')['2020-7']
```

Out[86]:

	date	value	date3
date2			
2020-07-01	2020-07-01	25	2020-07-01
2020-07-02	2020-07-02	20	2020-07-02
2020-07-05	2020-07-05	24	2020-07-05
2020-07-10	2020-07-10	33	2020-07-10
2020-07-23	2020-07-23	4	2020-07-23
2020-07-24	2020-07-24	45	2020-07-24
2020-07-28	2020-07-28	69	2020-07-28

1.5 strftime( )으로 날짜 타입의 변수를 문자열로 변환하기

datetime 타입의 컬럼을 인덱스로 변환(set\_index( ))해야지만 편리한 색인이 가능합니다.

02

## datetime의 주요 함수 [ 2 ]







datetime의 주요 함수 [2]

**strftime()**

**datetime을  
문자열로 변환**

## 2 datetime의 주요 함수 [2]

### ➤ strftime

```
datetime.strftime(data, format)
```

or

↪ datetime 자료형 변수

```
data.strftime(format)
```

↪ datetime 자료형 변수



## 2 datetime의 주요 함수 [2]

### ➤ strftime





# strftime( )으로 날짜 타입의 변수를 문자열로 변환하기

File

Edit

View

Insert

Cell

Kernel

Widgets

Help

Not Trusted



Python 3



Run



Code



## strftime( )으로 날짜 타입의 변수를 문자열로 변환하기

```
In [25]: # now 변수를 4자리년도-월-일 형식으로 출력  
I
```

```
In [26]: # 실습 - now 변수를 월-일 시:분 형식으로 출력
```

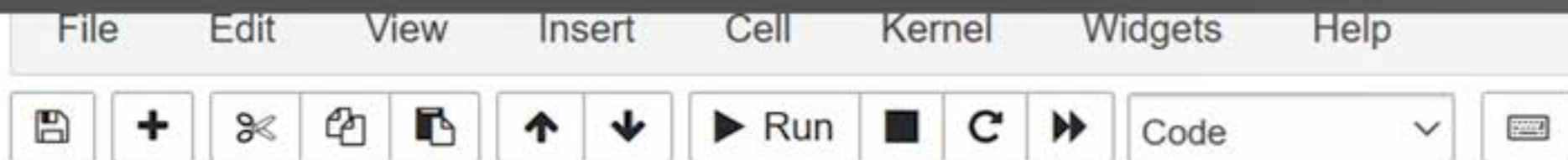
- Tip. Week와 요일 정보 얻기(strftime)

```
In [133]: # 심화 - Week와 요일 정보 얻기  
# 참고 사이트 : https://docs.python.org/3/library/datetime.html#strftime-and-strptime-t  
  
# %U : week number (sunday to monday) (e.g. 00, 01, ...53)  
# %V : ISO 8601 week number (monday to sunday) (e.g. 01, 02, ... 53)  
...  
# %a : weekday (Sun, Mon, ...)  
# %w : weekday (0, 1, 2, 3, ...) (0 = Sunday)
```



# strftime( )으로 날짜 타입의 변수를 문자열로 변환하기

◎ now 변수를 4자리 년도-월-일 형식으로 출력



## 1.5 strftime()으로 날짜 타입의 변수를 문자열로 변환하기

```
In [91]: # now 변수를 4자리년도-월-일 형식으로 출력  
now.strftime('%Y-%m-%d')  
dt.datetime.strftime(now, '%Y-%m-%d')
```

```
Out[91]: '2021-02-26'
```

```
In [26]: # 실습 - now 변수를 월-일 시:분 형식으로 출력
```

- Tip. Week와 요일 정보 얻기(strftime)

**dt.datetime.strftime(now, format)으로 지정해도 값은 동일합니다.**

# strftime( )으로 날짜 타입의 변수를 문자열로 변환하기

© now 변수를 월-일 시:분 형식으로 출력



## 1.5 strftime()으로 날짜 타입의 변수를 문자열로 변환하기

```
In [92]: # now 변수를 4자리년도-월-일 형식으로 출력
#now.strftime('%Y-%m-%d')
dt.datetime.strftime(now, '%Y-%m-%d')
```

Out[92]: '2021-02-26'

```
In [94]: # 실습 - now 변수를 월-일 시:분 형식으로 출력
now.strftime('%m-%d %H:%M')
```

Out[94]: '02-26 16:41'

- Tip. Week와 요일 정보 얻기(strftime)

```
In [133]: # 심화 - Week와 요일 정보 얻기
# 참고 사이트 : https://docs.python.org/3/library/datetime.html#strftime-and-strptime-t
```



# strftime( )으로 날짜 타임의 변수를 문자열로 변환하기

◎ now 변수를 월-일 시:분 형식으로 출력



```
dt.datetime.strftime(now, '%Y-%m-%d')
```

Out[92]: '2021-02-26'

```
In [94]: # 실습 - now 변수를 월-일 시:분 형식으로 출력  
now.strftime('%m-%d %H:%M')
```

Out[94]: '02-26 18:41'

- Tip. Week와 요일 정보 얻기(strftime)

```
In [133]: # 심화 - Week와 요일 정보 얻기  
# 참고 사이트 : https://docs.python.org/3/library/datetime.html#strftime-and-strptime-t  
  
# %U : week number (sunday to monday) (e.g. 00, 01, ...53)  
# %V : ISO 8601 week number (monday to sunday) (e.g. 01, 02, ... 53)  
...  
# %a : weekday (Sun, Mon, ...)  
# %w : weekday (0, 1, 2, 3, ...) (0 = Sunday)
```

strftime( )는 요일 등 날짜와 관련된 새로운 정보를 취득할 때 유용합니다.

# strftime( )으로 날짜 타임의 변수를 문자열로 변환하기

## ◎ week와 요일 정보 얻기



```
In [94]: # 실습 - now 변수를 월-일 시:분 형식으로 출력  
now.strftime('%m-%d %H:%M')
```

```
Out[94]: '02-26 18:41'
```

- Tip. Week와 요일 정보 얻기(strftime)

```
In [95]: now
```

```
Out[95]: datetime.datetime(2021, 2, 26, 18, 41, 54, 468656)
```

```
In [133]: 참고 - Week와 요일 정보 얻기
```

```
참고 사이트 : https://docs.python.org/3/library/datetime.html#strftime-and-strptime-behavior
```

```
: week number (sunday to monday) (e.g. 00, 01, ... 53)
```

```
: ISO 8601 week number (monday to sunday) (e.g. 01, 02, ... 53)
```

```
: weekday (Sun, Mon, ...)
```

```
: weekday (0, 1, 2, 3, ...) (0 = Sunday)
```

시간과 관련된 다양한 정보들을 알 수 있습니다.



# strftime( )으로 날짜 타입의 변수를 문자열로 변환하기

## © week와 요일 정보 얻기

### strftime() and strptime() Format Codes

The following is a  
with a standard C

**%a**

요일 정보 세 글자로 표시

requires, and these work on all platforms

Directive	Meaning	Example	Notes
%a	Weekday as locale's abbreviated name.	Sun, Mon, ..., Sat (en_US); So, Mo, ..., Sa (de_DE)	(1)
%A	Weekday as locale's full name.	Sunday, Monday, ..., Saturday (en_US); Sonntag, Montag, ..., Samstag (de_DE)	(1)
%W	Weekday as a decimal number, where 0 is Sunday and 6 is Saturday.	0, 1, ..., 6	
%d	Day of the month as a zero-padded decimal number.	01, 02, ..., 31	(9)
%b	Month as locale's abbreviated name.	Jan, Feb, ..., Dec (en_US); Jan, Feb, ..., Dez (de_DE)	(1)

# strftime( )으로 날짜 타입의 변수를 문자열로 변환하기

## © week와 요일 정보 얻기

The following is a list of all the format codes that the 1989 C standard requires, and these work on all platforms with a standard C implementation.

Directive	Meaning	Example	Notes
%a	Weekday as locale's abbreviated name.	Sun, Mon, ..., Sat (en_US); So, Mo, ..., Sa (de_DE)	(1)
%A	Weekday as locale's full name.	Sunday, Monday, ..., Saturday (en_US); Sonntag, Montag, ..., Samstag (de_DE)	(1)
%w	Weekday as a decimal number, where 0 is Sunday and 6 is Saturday.	0, 1, ..., 6	
%d	Day of the month as a zero-padded decimal number.	01, 02, ..., 31	(9)
%b	Month as locale's abbreviated name.	Jan, Feb, ..., Dec (en_US); Jan, Feb, ..., Dez (de_DE)	(1)



# strftime( )으로 날짜 타입의 변수를 문자열로 변환하기

## © week와 요일 정보 얻기

%a	Weekday as locale's abbreviated name.	Sun, Mon, ..., Sat (en_US); So, Mo, ..., Sa (de_DE)	(1)
%A	Weekday as locale's full name.	Sunday, Monday, ..., Saturday (en_US); Sonntag, Montag, ..., Samstag (de_DE)	(1)
%w	Weekday as a decimal number, where 0 is Sunday and 6 is Saturday.	0, 1, ..., 6	
%d	Day of the month as a zero-padded decimal number.	01, 02, ..., 31	(9)
%b	Month as locale's abbreviated name.	Jan, Feb, ..., Dec (en_US); Jan, Feb, ..., Dez (de_DE)	(1)
%B	Month as locale's full name.	January, February, ..., December (en_US); Januar, Februar, ..., Dezember (de_DE)	(1)
%m	Month as a zero-padded decimal number.	01, 02, ..., 12	(9)

**%b**

월 정보 앞 세글자로 표시

# strftime( )으로 날짜 타임의 변수를 문자열로 변환하기

## ◎ week와 요일 정보 얻기



```
In [94]: # 실습 - now 변수를 월-일 시:분 형식으로 출력  
now.strftime('%m-%d %H:%M')
```

```
Out[94]: '02-26 18:41'
```

- Tip. Week와 요일 정보 얻기(strftime)

```
In [95]: now
```

```
Out[95]: datetime.datetime(2021, 2, 26, 18, 41, 54, 468656)
```

```
In [133]: # 심화 - Week와 요일 정보 얻기  
# 참고 사이트 : https://docs.python.org/3/library/datetime.html#strftime-and-strptime-t  
  
# %U : week number (sunday to monday) (e.g. 00, 01, ...53)  
# %V : ISO 8601 week number (monday to sunday) (e.g. 01, 02, ... 53)  
...  
# %a : weekday (Sun, Mon, ...)  
# %w : weekday (0, 1, 2, 3, ...) (0 = Sunday)
```

주로 주차 정보나 요일 정보 등을 많이 사용합니다.



# strftime( )으로 날짜 타입의 변수를 문자열로 변환하기

## ◎ week와 요일 정보 얻기



In [95]: now

Out[95]: datetime.datetime(2021, 2, 26, 18, 41, 54, 468656)

```
In [133]: # 심화 - Week와 요일 정보 얻기
# 참고 사이트 : https://docs.python.org/3/library/datetime.html#strftime-and-strptime-t

# %U : week number (sunday to monday) (e.g. 00, 01, ...53)
# %V : ISO 8601 week number (monday to sunday) (e.g. 01, 02, ... 53)
...
# %a : weekday (Sun, Mon, ...)
# %w : weekday (0,1,2,3,...) (0 = Sunday)
```

Out[133]: Ellipsis

[실습 #2] df에 week number(%U), weekday(%a) 컬럼 추가

format을 활용해 원하는 형태로 변환할 수 있습니다.

# strftime( )으로 날짜 타입의 변수를 문자열로 변환하기

## ◎ 주차별/요일별 value column의 합 구하기

✓ drop 실행

In [98]: df

```
In [97]: df.drop(['date2', 'date3'], axis = 1, inplace = True)
         abels, axis, index, columns, level, inplace, errors)
         3885         for axis, labels in axes.items():
         3886             if labels is not None:
-> 3887                 obj = obj._drop_axis(labels, axis, level=level, errors=errors)
         3888
         3889             if inplace:

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py in _drop_axis(self, labels, axis, level, errors)
         3919         new_axis = axis.drop(labels, level=level, errors=errors)
         3920         else:
-> 3921         new_axis = axis.drop(labels, errors=errors)
```

column이기 때문에 drop할 때 axis=1로 설정합니다.



# strftime( )으로 날짜 타입의 변수를 문자열로 변환하기

## ◎ 주차별/요일별 value column의 합 구하기

✓ strftime( ) 활용해서 주차/요일 정보 추가하기

```
dtypes: datetime64[ns](1), int64(1)
memory usage: 288.0 bytes
```

[실습 #2] df에 week number(%U), weekday(%a) 컬럼 추가

```
In [27]: #df에 week number(%V), weekday(%w) 컬럼 추가
```

I

[실습 #3-1] 주차별 value 컬럼의 합 구하기

```
In [104]: df
```

Out [104]:

	date	value
0	2020-07-01	25
1	2020-07-02	20
2	2020-07-05	24

# strftime( )으로 날짜 타입의 변수를 문자열로 변환하기

## ◎ 주차별/요일별 value column의 합 구하기

✓ weekNumber(주차), weekday(요일) 컬럼 추가

```
In [106]: now.strftime('%V')
```

```
Out[106]: datetime.datetime(2021, 2, 26, 18, 41, 54, 468656)
```

```
In [105]: #df에 week number(%V), weekday(%w) 컬럼 추가  
df
```

```
Out[105]:
```

	date	value
0	2020-07-01	25
1	2020-07-02	20
2	2020-07-05	24
3	2020-07-10	33
4	2020-07-23	4
5	2020-07-24	45
6	2020-07-28	69



# strftime( )으로 날짜 타입의 변수를 문자열로 변환하기

## ◎ 주차별/요일별 value column의 합 구하기



[실습 #2] df에 week number(%U), weekday(%a) 컬럼 추가

In [ ]:

In [108]: now.strftime('%V')

Out[108]: '08'

In [109]: now.strftime('%W')

Out[109]: '5'

In [105]: *#df에 week number(%V), weekday(%w) 컬럼 추가*  
df

Out[105]:

	date	value
0	2020-07-01	25

strftime을 DataFrame의 date column에 각각 적용해서 관련 column을 추가합니다.

# strftime( )으로 날짜 타입의 변수를 문자열로 변환하기

## ◎ 주차별/요일별 value column의 합 구하기



[실습 #2] df에 week number(%U), weekday(%a) 컬럼 추가

In [ ]:

In [108]: now.strftime('%V')

Out[108]: '08'

In [109]: now.strftime('%W')

Out[109]: '5'

In [110]: *#df에 week number(%V), weekday(%w) 컬럼 추가*  
df.date.apply(lambda X: X.strftime('|'))

Out[110]:

0	2020-07-01
1	2020-07-02
2	2020-07-05
3	2020-07-10

df의 date 컬럼에 익명함수(lambda)를 활용하여 strftime() 함수를 적용합니다.

6 2020-07-26  
7 2020-08-05



# strftime( )으로 날짜 타입의 변수를 문자열로 변환하기

## ◎ 주차별/요일별 value column의 합 구하기



Out [109]: '5'

```
In [115]: #df에 week number(%V), weekday(%w) 컬럼 추가
df['weekNumber'] = df.date.apply(lambda X: X.strftime('%V'))
df['weekDay'] = df.date.apply(lambda X: X.strftime('%a'))
```

```
In [116]: df
```

...

**pivot\_table 실행**

합 구하기

```
In [104]: df.pivot_table(index = 'weekNumber', aggfunc = 'sum', value)
```

Out [104]:

	date	value
0	2020-07-01	25
1	2020-07-02	20
2	2020-07-05	24
3	2020-07-10	33

# strftime( )으로 날짜 타입의 변수를 문자열로 변환하기

## ◎ 요일별 value column의 합 구하기



27	69
28	33
30	49
31	69
32	47
33	62
34	26

[실습 #3-2] 요일별 value 컬럼의 평균 구하기

```
In [ ]: df.pivot_table(index = 'weekNumber', aggfunc = 'sum', values = 'value')
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```



# 시계열 데이터



## 학습완료

### 1/ datetime의 주요 함수 [1]

strptime() : 문자열을 datetime으로 변환

### 2/ datetime의 주요 함수 [2]

strftime() : datetime을 문자열로 변환



수고하셨습니다!