# What is the method for specifying a format in pandas.to_datetime?

Authored by

**stats writer**

June 24, 2024

Pandas.to_datetime is a method used for converting a column or series of values into datetime format in pandas. This method allows for the specification of a desired format using the 'format' parameter, which follows the standard strftime directives for formatting dates and times. By specifying the format, the method can accurately convert the values into the desired datetime format, ensuring consistency and accuracy in data analysis.

## Specify Format in pandas.to_datetime

**You can use the pandas.to_datetime() function to convert a string column to a datetime column in a pandas DataFrame.**

**When using this function, you can use the format argument to specify the format that your date is in so that you avoid errors when converting it from string to datetime.**

**This function uses the following basic syntax:**

**df = pd.to_datetime(df, format='%m%d%Y %H:%M:%S'])**

**Here are the most common directives that you can provide to the format argument:**

**%m: Month as zero-padded number (01, 02, ... 12)%d: Day of the month as zero-padded number (01, 02, ... 31)%y: Year with century as number (2020, 2021, 2022,**

**...)%H: Hour (24-hour clock) as zero-padded number (00, 01, ... 23)%I: Hour (12-hour clock) as zero-padded number (01, 02, ... 12)%p: Either AM or PM%M: Minute as zero-padded number (00, 01, ... 59)%S: Second as zero-padded number (00, 01, ... 59)**

**For a complete list of directives, refer to .**

**The following example shows how to use the format argument within the to_datetime() function in different scenarios.**

### Example: Specify Format in pandas.to_datetime

**Suppose we have the following pandas DataFrame that contains information about total sales made on various dates at some retail store:**

```
import pandas as pd

#create DataFrame
df = pd.DataFrame({'date': ,
'sales': })

#view DataFrame
print(df)
```

**date sales**
**0 10012023 4:15:30 100**
**1 10042023 7:16:04 140**
**2 10062023 9:25:00 235**
**3 10142023 15:30:50 120**
**4 10152023 18:15:00 250**

**#view data type of each column in DataFrame**
**print(df.dtypes)**

**date object**
**sales int64**
**dtype: object**

**We can see that the date column is currently a string (i.e. object) column.**

**Suppose we attempt to use pandas.to_datetime() to convert this column to datetime:**

**#attempt to convert date column to datetime format**
**df = pd.to_datetime(df)**

**ParserError: month must be in 1..12: 10012023 4:15:30 present at position 0**

**We receive an error because the pandas.to_datetime() function doesn't recognize the date and time format that the date column is currently in.**

**We can also use the format argument to specify the format of the column:**

```
#convert date column to datetime format
df = pd.to_datetime(df, format='%m%d%Y %H:%M:%S')

#view DataFrame
print(df)

date sales
0 2023-10-01 04:15:30 100
1 2023-10-04 07:16:04 140
2 2023-10-06 09:25:00 235
3 2023-10-14 15:30:50 120
4 2023-10-15 18:15:00 250

#view updated type of each column
print(df.dtypes)

date datetime64
sales int64
```

**dtype: object**

**Note: You can find the complete documentation for the pandas to_datetime() function .**

**The following tutorials explain how to perform other common operations in pandas:**