5.3-Working with Date and Time (Optional)

KeytoDataScience.com

Table of Contents

💡💡💡💡💡 **Dates and Calendars** 💡💡💡💡💡

Hurricanes (also known as cyclones or typhoons) hit the U.S. state of Florida several times per year. In this section, we'll learn how to work with date objects in Python, starting with the dates of every hurricane to hit Florida since 1950. We'll learn how Python handles dates, common date operations, and the right way to format dates to avoid confusion.

Dates in Python

A date in Python is not a data type of its own, but we can import a module named datetime to work with dates as date objects.

In [1]:
```
import datetime

x = datetime.datetime.now()
print(x)
print(type(x))
```

```
2022-01-07 02:45:27.856690
<class 'datetime.datetime'>
```

Which day of the week?

Hurricane Andrew, which hit Florida on August 24, 1992, was one of the costliest and deadliest hurricanes in US history. Which day of the week did it make landfall?

Let's walk through all of the steps to figure this out.

In [2]:
```
# Import date from datetime
from datetime import date

# Create a date object
hurricane_andrew = date(1992,8,24)

# Which day of the week is the date?
# remember that Python counts days of the week starting from Monday as 0, Tuesday as 1,
print(hurricane_andrew.weekday())
```

```
0
```

How many hurricanes come early?

In this execise, you will work with a list of the hurricanes that made landfall in Florida from 1950 to 2017. There were 235 in total. Check out the variable florida_hurricane_dates, which has all of these dates.

Atlantic hurricane season officially begins on June 1. How many hurricanes since 1950 have made landfall in Florida before the official start of hurricane season?

In [3]:
```
florida_hurricane_dates = [datetime.date(1950, 8, 31), datetime.date(1950, 9, 5), datet
```

In [4]:
```
# Counter for how many before June 1
early_hurricanes = 0

# We loop over the dates
for hurricane in florida_hurricane_dates:
  # Check if the month is before June (month number 6)
  if hurricane.month < 6:
    early_hurricanes = early_hurricanes + 1

print(early_hurricanes)
```

```
10
```

↥ back to top

Math with Dates

Subtracting dates

Python date objects let us treat calendar dates as something similar to numbers: we can compare them, sort them, add, and even subtract them. This lets us do math with dates in a way that would be a pain to do by hand.

The 2007 Florida hurricane season was one of the busiest on record, with 8 hurricanes in one year. The first one hit on May 9th, 2007, and the last one hit on December 13th, 2007. How many days elapsed between the first and last hurricane in 2007?

In [5]:
```
# Import date
from datetime import date

# Create a date object for May 9th, 2007
start = date(2007, 5, 9)

# Create a date object for December 13th, 2007
end = date(2007, 12, 13)

# Subtract the two dates and print the number of days
print((end - start).days)
```

```
218
```

Counting events per calendar month

Hurricanes can make landfall in Florida throughout the year. As we've already discussed, some months are more hurricane-prone than others.

Using florida_hurricane_dates, let's see how hurricanes in Florida were distributed across months throughout the year.

Create a dictionary called hurricanes_each_month to hold your counts and set the initial counts to zero. You will loop over the list of hurricanes, incrementing the correct month in hurricanes_each_month as you go, and then print the result.

In [6]:
```
# A dictionary to count hurricanes per calendar month
hurricanes_each_month = {1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6:0, 7: 0, 8:0, 9:0, 10:0, 11:0,

# Loop over all hurricanes
for hurricane in florida_hurricane_dates:
    # Pull out the month
    month = hurricane.month
    # Increment the count in your dictionary by one
    hurricanes_each_month[month] += 1
```

```
print(hurricanes_each_month)
```

```
{1: 0, 2: 1, 3: 0, 4: 1, 5: 8, 6: 32, 7: 21, 8: 49, 9: 70, 10: 43, 11: 9, 12: 1}
```

Putting a list of dates in order

Much like numbers and strings, date objects in Python can be put in order. Earlier dates come before later ones, and so we can sort a list of date objects from earliest to latest.

What if our Florida hurricane dates had been scrambled? Your job is to put them back in chronological order, and then print the first and last dates from this sorted list.

In [7]:
```
dates_scrambled = [datetime.date(1988, 8, 4), datetime.date(1990, 10, 12), datetime.dat
```

In [8]:
```
# Print the first and last scrambled dates
print(dates_scrambled[0])
print(dates_scrambled[-1])
```

```
1988-08-04
2011-07-18
```

In [9]:
```
# Put the dates in order
dates_ordered = sorted(dates_scrambled)

# Print the first and last ordered dates
print(dates_ordered[0])
print(dates_ordered[-1])
```

```
1950-08-31
2017-10-29
```

In [10]:
```
print(min(dates_scrambled))
print(max(dates_scrambled))
```

```
1950-08-31
2017-10-29
```

In [11]:
```
# import date
from datetime import date

# create our dates
d1 = date(2017,11,5)
d2 = date(2017,12,4)

delta = d2-d1

print(delta.days)
```

```
29
```

In [12]:
```
# import timedelta
from datetime import timedelta
```

```
# create a 29 day timedelta
td = timedelta(days=29)
print(d1+td)
```

```
2017-12-04
```

↥ back to top

Turning dates into strings

We want to convert date into strings when:

- we want to print our results in a desired format
- if we want to put the dates into the file name

In [13]:
```
# import date
from datetime import date

# create our dates
d1 = date(2017,11,5)

# default ISO format: YYYY-MM-DD
print(d1)
```

```
2017-11-05
```

In [14]:
```
d1.isoformat()
```

Out[14]:
```
'2017-11-05'
```

In [15]:
```
str(d1)
```

Out[15]:
```
'2017-11-05'
```

Printing dates in a friendly format

Because people may want to see dates in many different formats, Python comes with very flexible functions for turning date objects into strings.

Let's see what event was recorded first in the Florida hurricane data set. In this exercise, you will format the earliest date in the florida_hurriance_dates list in two ways so you can decide which one you want to use: either the ISO standard or the typical US style.

In [16]:
```
# Assign the earliest date to first_date
first_date = min(florida_hurricane_dates)

# Convert to ISO and US formats
iso = "Our earliest hurricane date: " + first_date.isoformat()
us = "Our earliest hurricane date: " + first_date.strftime("%m/%d/%Y")
```

```
print("ISO: " + iso)
print("US: " + us)
```

```
ISO: Our earliest hurricane date: 1950-08-31
US: Our earliest hurricane date: 08/31/1950
```

A reference of all the legal format codes: https://www.w3schools.com/python/python_datetime.asp

Representing dates in different ways

date objects in Python have a great number of ways they can be printed out as strings. In some cases, you want to know the date in a clear, language-agnostic format. In other cases, you want something which can fit into a paragraph and flow naturally.

Let's try printing out the same date, August 26, 1992 (the day that Hurricane Andrew made landfall in Florida), in a number of different ways, to practice using the .strftime() method.

In [17]:
```
# Import date
from datetime import date

# Create a date object
d = date(1992, 8, 26)

# Print the date in the format 'YYYY-MM'
print(d.strftime("%Y-%m"))
```

```
1992-08
```

In [18]:
```
# Print the date in the format 'MONTH (YYYY)'
print(d.strftime("%B (%Y)"))
```

```
August (1992)
```

In [19]:
```
# Print the date in the format 'YYYY-DDD'
print(d.strftime("%Y-%j"))
```

```
1992-239
```

↥ back to top

💡💡💡💡💡 **Combining Dates and Times** 💡💡💡💡💡

Bike sharing programs have swept through cities around the world. Working with all of the comings and goings of one bike in Washington, D.C., we'll practice working with dates and times together. We'll parse dates and times from text, analyze peak trip times, calculate ride durations, and more.

Dates and times

Creating datetimes by hand

Often you create datetime objects based on outside data. Sometimes though, you want to create a datetime object from scratch.

We're going to create a few different datetime objects from scratch to get the hang of that process. These come from the bikeshare data set that we'll use throughout the rest of the section.

In [20]:
```
# Import datetime
from datetime import datetime

# Create a datetime object for October 1, 2017 at 15:26:26.
dt = datetime(2017, 10, 1, 15, 26, 26)

# Print the date
print(dt)
```

```
2017-10-01 15:26:26
```

In [21]:
```
# Replace the year with 1917
dt_old = dt.replace(year=1917)

# Print the new date
print(dt_old)
```

```
1917-10-01 15:26:26
```

Counting events before and after noon

In this section, we'll be working with a list of all bike trips for one Capital Bikeshare bike, W20529, from October 1, 2017 to December 31, 2017. This list has been loaded as onebike_datetimes.

Each element of the list is a dictionary with two entries: start is a datetime object corresponding to the start of a trip (when a bike is removed from the dock) and end is a datetime object corresponding to the end of a trip (when a bike is put back into a dock).

We can use this data set to understand better how this bike was used. Did more trips start before noon or after noon?

In [22]:
```
import datetime

onebike_datetimes = [{'end': datetime.datetime(2017, 10, 1, 15, 26, 26),  'start': date
```

In [23]:
```
# Create dictionary to hold results
trip_counts = {'AM': 0, 'PM': 0}

# Loop over all trips
for trip in onebike_datetimes:
  # Check to see if the trip starts before noon
    if trip['start'].hour < 12:
        # Increment the counter for before noon
        trip_counts['AM'] += 1
    else:
        # Increment the counter for after noon
```

```
        trip_counts['PM'] += 1

print(trip_counts)
```

```
{'AM': 94, 'PM': 196}
```

↥ back to top

Printing and parsing datetimes

In [24]:
```
dt = datetime.datetime(2017, 10, 1, 15, 26, 26)

#default format
print(dt, type(dt))
print(dt.strftime("%Y-%m-%d"), type(dt.strftime("%Y-%m-%d")))
print(dt.strftime("%Y-%m-%d %H-%M-%S"),type(dt.strftime("%Y-%m-%d %H-%M-%S")))
```

```
2017-10-01 15:26:26 <class 'datetime.datetime'>
2017-10-01 <class 'str'>
2017-10-01 15-26-26 <class 'str'>
```

Turning strings into datetimes

When you download data from the Internet, dates and times usually come to you as strings. Often the first step is to turn those strings into datetime objects.

In this exercise, we will practice this transformation.

In [25]:
```
# Import the datetime class
from datetime import datetime

# Starting string, in YYYY-MM-DD HH:MM:SS format
s = '2017-02-03 00:00:01'

# Write a format string to parse s
fmt = '%Y-%m-%d %H:%M:%S'

# Create a datetime object d
d = datetime.strptime(s, fmt)

# Print s, d
print(s, type(s))
print(d, type(d))
```

```
2017-02-03 00:00:01 <class 'str'>
2017-02-03 00:00:01 <class 'datetime.datetime'>
```

In [26]:
```
# Import the datetime class
from datetime import datetime

# Starting string, in YYYY-MM-DD format
s = '2030-10-15'

# Write a format string to parse s
fmt = '%Y-%m-%d'
```

```
# Create a datetime object d
d = datetime.strptime(s, fmt)

# Print s, d
print(s, type(s))
print(d, type(d))
```

```
2030-10-15 <class 'str'>
2030-10-15 00:00:00 <class 'datetime.datetime'>
```

In [27]:

```
# Import the datetime class
from datetime import datetime

# Starting string, in MM/DD/YYYY HH:MM:SS format
s = '12/15/1986 08:00:00'

# Write a format string to parse s
fmt = '%m/%d/%Y %H:%M:%S'

# Create a datetime object d
d = datetime.strptime(s, fmt)

# Print s, d
print(s, type(s))
print(d, type(d))
```

```
12/15/1986 08:00:00 <class 'str'>
1986-12-15 08:00:00 <class 'datetime.datetime'>
```

Parsing pairs of strings as datetimes

Up until now, we've been working with a pre-processed list of datetimes for W20529's trips. For this exercise, we're going to go one step back in the data cleaning pipeline and work with the strings that the data started as.

Explore onebike_datetime_strings in the IPython shell to determine the correct format.

In [28]:

```
onebike_datetime_strings = [('2017-10-01 15:23:25', '2017-10-01 15:26:26'), ('2017-10-0
```

In [29]:

```
# Write down the format string
fmt = "%Y-%m-%d %H:%M:%S"

# Initialize a list for holding the pairs of datetime objects
onebike_datetimes = []

# Loop over all trips
for start, end in onebike_datetime_strings:
    trip = {'start': datetime.strptime(start, fmt),
            'end': datetime.strptime(end, fmt)}

    # Append the trip
    onebike_datetimes.append(trip)

print(onebike_datetimes)
```

[{'start': datetime.datetime(2017, 10, 1, 15, 23, 25), 'end': datetime.datetime(2017, 10, 1, 15, 26, 26)}, {'start': datetime.datetime(2017, 10, 1, 15, 42, 57), 'end': datetime.datetime(2017, 10, 1, 17, 49, 59)}, {'start': datetime.datetime(2017, 10, 2, 6, 37, 10), 'end': datetime.datetime(2017, 10, 2, 6, 42, 53)}, {'start': datetime.datetime(2017, 10, 2, 8, 56, 45), 'end': datetime.datetime(2017, 10, 2, 9, 18, 3)}, {'start': datetime.datetime(2017, 10, 2, 18, 23, 48), 'end': datetime.datetime(2017, 10, 2, 18, 45, 5)}, {'start': datetime.datetime(2017, 10, 2, 18, 48, 8), 'end': datetime.datetime(2017, 10, 2, 19, 10, 54)}, {'start': datetime.datetime(2017, 10, 2, 19, 18, 10), 'end': datetime.datetime(2017, 10, 2, 19, 31, 45)}, {'start': datetime.datetime(2017, 10, 2, 19, 37, 32), 'end': datetime.datetime(2017, 10, 2, 19, 46, 37)}, {'start': datetime.datetime(2017, 10, 3, 8, 24, 16), 'end': datetime.datetime(2017, 10, 3, 8, 32, 27)}, {'start': datetime.datetime(2017, 10, 3, 18, 17, 7), 'end': datetime.datetime(2017, 10, 3, 18, 27, 46)}, {'start': datetime.datetime(2017, 10, 3, 19, 24, 10), 'end': datetime.datetime(2017, 10, 3, 19, 52, 8)}, {'start': datetime.datetime(2017, 10, 3, 20, 17, 6), 'end': datetime.datetime(2017, 10, 3, 20, 23, 52)}, {'start': datetime.datetime(2017, 10, 3, 20, 45, 21), 'end': datetime.datetime(2017, 10, 3, 20, 57, 10)}, {'start': datetime.datetime(2017, 10, 4, 7, 4, 57), 'end': datetime.datetime(2017, 10, 4, 7, 13, 31)}, {'start': datetime.datetime(2017, 10, 4, 7, 13, 42), 'end': datetime.datetime(2017, 10, 4, 7, 21, 54)}, {'start': datetime.datetime(2017, 10, 4, 14, 22, 12), 'end': datetime.datetime(2017, 10, 4, 14, 50)}, {'start': datetime.datetime(2017, 10, 4, 15, 7, 27), 'end': datetime.datetime(2017, 10, 4, 15, 44, 49)}, {'start': datetime.datetime(2017, 10, 4, 15, 46, 41), 'end': datetime.datetime(2017, 10, 4, 16, 32, 33)}, {'start': datetime.datetime(2017, 10, 4, 16, 34, 44), 'end': datetime.datetime(2017, 10, 4, 16, 46, 59)}, {'start': datetime.datetime(2017, 10, 4, 17, 26, 6), 'end': datetime.datetime(2017, 10, 4, 17, 31, 36)}, {'start': datetime.datetime(2017, 10, 4, 17, 42, 3), 'end': datetime.datetime(2017, 10, 4, 17, 50, 41)}, {'start': datetime.datetime(2017, 10, 5, 7, 49, 2), 'end': datetime.datetime(2017, 10, 5, 8, 12, 55)}, {'start': datetime.datetime(2017, 10, 5, 8, 26, 21), 'end': datetime.datetime(2017, 10, 5, 8, 29, 45)}, {'start': datetime.datetime(2017, 10, 5, 8, 33, 27), 'end': datetime.datetime(2017, 10, 5, 8, 38, 31)}, {'start': datetime.datetime(2017, 10, 5, 16, 35, 35), 'end': datetime.datetime(2017, 10, 5, 16, 51, 52)}, {'start': datetime.datetime(2017, 10, 5, 17, 53, 31), 'end': datetime.datetime(2017, 10, 5, 18, 16, 50)}, {'start': datetime.datetime(2017, 10, 6, 8, 17, 17), 'end': datetime.datetime(2017, 10, 6, 8, 38, 1)}, {'start': datetime.datetime(2017, 10, 6, 11, 39, 40), 'end': datetime.datetime(2017, 10, 6, 11, 50, 38)}, {'start': datetime.datetime(2017, 10, 6, 12, 59, 54), 'end': datetime.datetime(2017, 10, 6, 13, 13, 14)}, {'start': datetime.datetime(2017, 10, 6, 13, 43, 5), 'end': datetime.datetime(2017, 10, 6, 14, 14, 56)}, {'start': datetime.datetime(2017, 10, 6, 14, 28, 15), 'end': datetime.datetime(2017, 10, 6, 15, 9, 26)}, {'start': datetime.datetime(2017, 10, 6, 15, 50, 10), 'end': datetime.datetime(2017, 10, 6, 16, 12, 34)}, {'start': datetime.datetime(2017, 10, 6, 16, 32, 16), 'end': datetime.datetime(2017, 10, 6, 16, 39, 31)}, {'start': datetime.datetime(2017, 10, 6, 16, 44, 8), 'end': datetime.datetime(2017, 10, 6, 16, 48, 39)}, {'start': datetime.datetime(2017, 10, 6, 16, 53, 43), 'end': datetime.datetime(2017, 10, 6, 17, 9, 3)}, {'start': datetime.datetime(2017, 10, 7, 11, 38, 55), 'end': datetime.datetime(2017, 10, 7, 11, 53, 6)}, {'start': datetime.datetime(2017, 10, 7, 14, 3, 36), 'end': datetime.datetime(2017, 10, 7, 14, 7, 5)}, {'start': datetime.datetime(2017, 10, 7, 14, 20, 3), 'end': datetime.datetime(2017, 10, 7, 14, 27, 36)}, {'start': datetime.datetime(2017, 10, 7, 14, 30, 50), 'end': datetime.datetime(2017, 10, 7, 14, 44, 51)}, {'start': datetime.datetime(2017, 10, 8, 0, 28, 26), 'end': datetime.datetime(2017, 10, 8, 0, 30, 48)}, {'start': datetime.datetime(2017, 10, 8, 11, 16, 21), 'end': datetime.datetime(2017, 10, 8, 11, 33, 24)}, {'start': datetime.datetime(2017, 10, 8, 12, 37, 3), 'end': datetime.datetime(2017, 10, 8, 13, 1, 29)}, {'start': datetime.datetime(2017, 10, 8, 13, 30, 37), 'end': datetime.datetime(2017, 10, 8, 13, 57, 53)}, {'start': datetime.datetime(2017, 10, 8, 14, 16, 40), 'end': datetime.datetime(2017, 10, 8, 15, 7, 19)}, {'start': datetime.datetime(2017, 10, 8, 15, 23, 50), 'end': datetime.datetime(2017, 10, 8, 15, 50, 1)}, {'start': datetime.datetime(2017, 10, 8, 15, 54, 12), 'end': datetime.datetime(2017, 10, 8, 16, 17, 42)}, {'start': datetime.datetime(2017, 10, 8, 16, 28, 52), 'end': datetime.datetime(2017, 10, 8, 16, 35, 18)}, {'start': datetime.datetime(2017, 10, 8, 23, 8, 14), 'end': datetime.datetime(2017, 10, 8, 23, 33, 41)}, {'start': datetime.datetime(2017, 10, 8, 23, 34, 49), 'end': datetime.datetime(2017, 10, 8, 23, 45, 11)}, {'start': datetime.datetime(2017, 10, 8, 23, 46, 47), 'end': datetime.datetime(2017, 10, 9, 0, 10, 57)}, {'start': datetim

e.datetime(2017, 10, 9, 0, 12, 58), 'end': datetime.datetime(2017, 10, 9, 0, 36, 40)},
{'start': datetime.datetime(2017, 10, 9, 0, 37, 2), 'end': datetime.datetime(2017, 10,
9, 0, 53, 33)}, {'start': datetime.datetime(2017, 10, 9, 1, 23, 29), 'end': datetime.dat
etime(2017, 10, 9, 1, 48, 13)}, {'start': datetime.datetime(2017, 10, 9, 1, 49, 25), 'en
d': datetime.datetime(2017, 10, 9, 2, 13, 35)}, {'start': datetime.datetime(2017, 10, 9,
2, 14, 11), 'end': datetime.datetime(2017, 10, 9, 2, 29, 40)}, {'start': datetime.dateti
me(2017, 10, 9, 13, 4, 32), 'end': datetime.datetime(2017, 10, 9, 13, 13, 25)}, {'star
t': datetime.datetime(2017, 10, 9, 14, 30, 10), 'end': datetime.datetime(2017, 10, 9, 1
4, 38, 55)}, {'start': datetime.datetime(2017, 10, 9, 15, 6, 47), 'end': datetime.dateti
me(2017, 10, 9, 15, 11, 30)}, {'start': datetime.datetime(2017, 10, 9, 16, 43, 25), 'en
d': datetime.datetime(2017, 10, 9, 16, 45, 38)}, {'start': datetime.datetime(2017, 10, 1
0, 15, 32, 58), 'end': datetime.datetime(2017, 10, 10, 15, 51, 24)}, {'start': datetime.
datetime(2017, 10, 10, 16, 47, 55), 'end': datetime.datetime(2017, 10, 10, 17, 3, 47)},
{'start': datetime.datetime(2017, 10, 10, 17, 51, 5), 'end': datetime.datetime(2017, 10,
10, 18, 0, 18)}, {'start': datetime.datetime(2017, 10, 10, 18, 8, 12), 'end': datetime.d
atetime(2017, 10, 10, 18, 19, 11)}, {'start': datetime.datetime(2017, 10, 10, 19, 9, 3
5), 'end': datetime.datetime(2017, 10, 10, 19, 14, 32)}, {'start': datetime.datetime(201
7, 10, 10, 19, 17, 11), 'end': datetime.datetime(2017, 10, 10, 19, 23, 8)}, {'start': da
tetime.datetime(2017, 10, 10, 19, 28, 11), 'end': datetime.datetime(2017, 10, 10, 19, 4
4, 40)}, {'start': datetime.datetime(2017, 10, 10, 19, 55, 35), 'end': datetime.datetime
(2017, 10, 10, 20, 11, 54)}, {'start': datetime.datetime(2017, 10, 10, 22, 20, 43), 'en
d': datetime.datetime(2017, 10, 10, 22, 33, 23)}, {'start': datetime.datetime(2017, 10,
11, 4, 40, 52), 'end': datetime.datetime(2017, 10, 11, 4, 59, 22)}, {'start': datetime.d
atetime(2017, 10, 11, 6, 28, 58), 'end': datetime.datetime(2017, 10, 11, 6, 40, 13)},
{'start': datetime.datetime(2017, 10, 11, 16, 41, 7), 'end': datetime.datetime(2017, 10,
11, 17, 1, 14)}, {'start': datetime.datetime(2017, 10, 12, 8, 8, 30), 'end': datetime.da
tetime(2017, 10, 12, 8, 35, 3)}, {'start': datetime.datetime(2017, 10, 12, 8, 47, 2), 'e
nd': datetime.datetime(2017, 10, 12, 8, 59, 50)}, {'start': datetime.datetime(2017, 10,
12, 13, 13, 39), 'end': datetime.datetime(2017, 10, 12, 13, 37, 45)}, {'start': datetim
e.datetime(2017, 10, 12, 13, 40, 12), 'end': datetime.datetime(2017, 10, 12, 13, 48, 1
7)}, {'start': datetime.datetime(2017, 10, 12, 13, 49, 56), 'end': datetime.datetime(201
7, 10, 12, 13, 53, 16)}, {'start': datetime.datetime(2017, 10, 12, 14, 33, 18), 'end': d
atetime.datetime(2017, 10, 12, 14, 39, 57)}, {'start': datetime.datetime(2017, 10, 13, 1
5, 55, 39), 'end': datetime.datetime(2017, 10, 13, 15, 59, 41)}, {'start': datetime.date
time(2017, 10, 17, 17, 58, 48), 'end': datetime.datetime(2017, 10, 17, 18, 1, 38)}, {'st
art': datetime.datetime(2017, 10, 19, 20, 21, 45), 'end': datetime.datetime(2017, 10, 1
9, 20, 29, 15)}, {'start': datetime.datetime(2017, 10, 19, 21, 11, 39), 'end': datetime.
datetime(2017, 10, 19, 21, 29, 37)}, {'start': datetime.datetime(2017, 10, 19, 21, 30,
1), 'end': datetime.datetime(2017, 10, 19, 21, 47, 23)}, {'start': datetime.datetime(201
7, 10, 19, 21, 47, 34), 'end': datetime.datetime(2017, 10, 19, 21, 57, 7)}, {'start': da
tetime.datetime(2017, 10, 19, 21, 57, 24), 'end': datetime.datetime(2017, 10, 19, 22, 9,
52)}, {'start': datetime.datetime(2017, 10, 21, 12, 24, 9), 'end': datetime.datetime(201
7, 10, 21, 12, 36, 24)}, {'start': datetime.datetime(2017, 10, 21, 12, 36, 37), 'end': d
atetime.datetime(2017, 10, 21, 12, 42, 13)}, {'start': datetime.datetime(2017, 10, 21, 1
3, 47, 43), 'end': datetime.datetime(2017, 10, 22, 11, 9, 36)}, {'start': datetime.datet
ime(2017, 10, 22, 13, 28, 53), 'end': datetime.datetime(2017, 10, 22, 13, 31, 44)}, {'st
art': datetime.datetime(2017, 10, 22, 13, 47, 5), 'end': datetime.datetime(2017, 10, 22,
13, 56, 33)}, {'start': datetime.datetime(2017, 10, 22, 14, 26, 41), 'end': datetime.dat
etime(2017, 10, 22, 14, 32, 39)}, {'start': datetime.datetime(2017, 10, 22, 14, 54, 41),
'end': datetime.datetime(2017, 10, 22, 15, 9, 58)}, {'start': datetime.datetime(2017, 1
0, 22, 16, 40, 29), 'end': datetime.datetime(2017, 10, 22, 16, 51, 40)}, {'start': datet
ime.datetime(2017, 10, 22, 17, 58, 46), 'end': datetime.datetime(2017, 10, 22, 18, 28, 3
7)}, {'start': datetime.datetime(2017, 10, 22, 18, 45, 16), 'end': datetime.datetime(201
7, 10, 22, 18, 50, 34)}, {'start': datetime.datetime(2017, 10, 22, 18, 56, 22), 'end': d
atetime.datetime(2017, 10, 22, 19, 11, 10)}, {'start': datetime.datetime(2017, 10, 23, 1
0, 14, 8), 'end': datetime.datetime(2017, 10, 23, 10, 35, 32)}, {'start': datetime.datet
ime(2017, 10, 23, 11, 29, 36), 'end': datetime.datetime(2017, 10, 23, 14, 38, 34)}, {'st
art': datetime.datetime(2017, 10, 23, 15, 4, 52), 'end': datetime.datetime(2017, 10, 23,
15, 32, 58)}, {'start': datetime.datetime(2017, 10, 23, 15, 33, 48), 'end': datetime.dat
etime(2017, 10, 23, 17, 6, 47)}, {'start': datetime.datetime(2017, 10, 23, 17, 13, 16),

'end': datetime.datetime(2017, 10, 23, 19, 31, 26)}, {'start': datetime.datetime(2017, 10, 23, 19, 55, 3), 'end': datetime.datetime(2017, 10, 23, 20, 25, 53)}, {'start': datetime.datetime(2017, 10, 23, 21, 47, 54), 'end': datetime.datetime(2017, 10, 23, 22, 18, 4)}, {'start': datetime.datetime(2017, 10, 23, 22, 34, 12), 'end': datetime.datetime(2017, 10, 23, 22, 48, 42)}, {'start': datetime.datetime(2017, 10, 24, 6, 55, 1), 'end': datetime.datetime(2017, 10, 24, 7, 2, 17)}, {'start': datetime.datetime(2017, 10, 24, 14, 56, 7), 'end': datetime.datetime(2017, 10, 24, 15, 3, 16)}, {'start': datetime.datetime(2017, 10, 24, 15, 51, 36), 'end': datetime.datetime(2017, 10, 24, 15, 59, 50)}, {'start': datetime.datetime(2017, 10, 24, 16, 31, 10), 'end': datetime.datetime(2017, 10, 24, 16, 55, 9)}, {'start': datetime.datetime(2017, 10, 28, 14, 26, 14), 'end': datetime.datetime(2017, 10, 28, 14, 32, 34)}, {'start': datetime.datetime(2017, 11, 1, 9, 41, 54), 'end': datetime.datetime(2017, 11, 1, 9, 52, 23)}, {'start': datetime.datetime(2017, 11, 1, 20, 16, 11), 'end': datetime.datetime(2017, 11, 1, 20, 32, 13)}, {'start': datetime.datetime(2017, 11, 2, 19, 44, 29), 'end': datetime.datetime(2017, 11, 2, 19, 50, 56)}, {'start': datetime.datetime(2017, 11, 2, 20, 14, 37), 'end': datetime.datetime(2017, 11, 2, 20, 30, 29)}, {'start': datetime.datetime(2017, 11, 2, 21, 35, 47), 'end': datetime.datetime(2017, 11, 2, 21, 38, 57)}, {'start': datetime.datetime(2017, 11, 3, 9, 59, 27), 'end': datetime.datetime(2017, 11, 3, 10, 11, 46)}, {'start': datetime.datetime(2017, 11, 3, 10, 13, 22), 'end': datetime.datetime(2017, 11, 3, 10, 32, 2)}, {'start': datetime.datetime(2017, 11, 3, 10, 44, 25), 'end': datetime.datetime(2017, 11, 3, 10, 50, 34)}, {'start': datetime.datetime(2017, 11, 3, 16, 6, 43), 'end': datetime.datetime(2017, 11, 3, 16, 44, 38)}, {'start': datetime.datetime(2017, 11, 3, 16, 45, 54), 'end': datetime.datetime(2017, 11, 3, 17, 0, 27)}, {'start': datetime.datetime(2017, 11, 3, 17, 7, 15), 'end': datetime.datetime(2017, 11, 3, 17, 35, 5)}, {'start': datetime.datetime(2017, 11, 3, 17, 36, 5), 'end': datetime.datetime(2017, 11, 3, 17, 46, 48)}, {'start': datetime.datetime(2017, 11, 3, 17, 50, 31), 'end': datetime.datetime(2017, 11, 3, 18, 0, 3)}, {'start': datetime.datetime(2017, 11, 3, 19, 22, 56), 'end': datetime.datetime(2017, 11, 3, 19, 45, 51)}, {'start': datetime.datetime(2017, 11, 4, 13, 14, 10), 'end': datetime.datetime(2017, 11, 4, 13, 26, 15)}, {'start': datetime.datetime(2017, 11, 4, 14, 18, 37), 'end': datetime.datetime(2017, 11, 4, 14, 30, 5)}, {'start': datetime.datetime(2017, 11, 4, 14, 45, 59), 'end': datetime.datetime(2017, 11, 4, 15, 3, 20)}, {'start': datetime.datetime(2017, 11, 4, 15, 16, 3), 'end': datetime.datetime(2017, 11, 4, 15, 44, 30)}, {'start': datetime.datetime(2017, 11, 4, 16, 37, 46), 'end': datetime.datetime(2017, 11, 4, 16, 58, 22)}, {'start': datetime.datetime(2017, 11, 4, 17, 13, 19), 'end': datetime.datetime(2017, 11, 4, 17, 34, 50)}, {'start': datetime.datetime(2017, 11, 4, 18, 10, 34), 'end': datetime.datetime(2017, 11, 4, 18, 58, 44)}, {'start': datetime.datetime(2017, 11, 5, 1, 56, 50), 'end': datetime.datetime(2017, 11, 5, 1, 1, 4)}, {'start': datetime.datetime(2017, 11, 5, 8, 33, 33), 'end': datetime.datetime(2017, 11, 5, 8, 53, 46)}, {'start': datetime.datetime(2017, 11, 5, 8, 58, 8), 'end': datetime.datetime(2017, 11, 5, 9, 3, 39)}, {'start': datetime.datetime(2017, 11, 5, 11, 5, 8), 'end': datetime.datetime(2017, 11, 5, 11, 30, 5)}, {'start': datetime.datetime(2017, 11, 6, 8, 50, 18), 'end': datetime.datetime(2017, 11, 6, 8, 59, 5)}, {'start': datetime.datetime(2017, 11, 6, 9, 4, 3), 'end': datetime.datetime(2017, 11, 6, 9, 13, 47)}, {'start': datetime.datetime(2017, 11, 6, 16, 19, 36), 'end': datetime.datetime(2017, 11, 6, 17, 2, 55)}, {'start': datetime.datetime(2017, 11, 6, 17, 21, 27), 'end': datetime.datetime(2017, 11, 6, 17, 34, 6)}, {'start': datetime.datetime(2017, 11, 6, 17, 36, 1), 'end': datetime.datetime(2017, 11, 6, 17, 57, 32)}, {'start': datetime.datetime(2017, 11, 6, 17, 59, 52), 'end': datetime.datetime(2017, 11, 6, 18, 15, 8)}, {'start': datetime.datetime(2017, 11, 6, 18, 18, 36), 'end': datetime.datetime(2017, 11, 6, 18, 21, 17)}, {'start': datetime.datetime(2017, 11, 6, 19, 24, 31), 'end': datetime.datetime(2017, 11, 6, 19, 37, 57)}, {'start': datetime.datetime(2017, 11, 6, 19, 49, 16), 'end': datetime.datetime(2017, 11, 6, 20, 3, 14)}, {'start': datetime.datetime(2017, 11, 7, 7, 50, 48), 'end': datetime.datetime(2017, 11, 7, 8, 1, 32)}, {'start': datetime.datetime(2017, 11, 8, 13, 11, 51), 'end': datetime.datetime(2017, 11, 8, 13, 18, 5)}, {'start': datetime.datetime(2017, 11, 8, 21, 34, 47), 'end': datetime.datetime(2017, 11, 8, 21, 46, 5)}, {'start': datetime.datetime(2017, 11, 8, 22, 2, 30), 'end': datetime.datetime(2017, 11, 8, 22, 4, 47)}, {'start': datetime.datetime(2017, 11, 9, 7, 1, 11), 'end': datetime.datetime(2017, 11, 9, 7, 12, 10)}, {'start': datetime.datetime(2017, 11, 9, 8, 2, 2), 'end': datetime.datetime(2017, 11, 9, 8, 8, 28)}, {'start': datetime.datetime(2017, 11, 9, 8, 19, 59), 'end': datetime.datetime(2017, 11, 9, 8, 32, 24)}, {'start': datetime.datetime(2017, 11, 9, 8, 41, 31), 'end': datetime.dateti

me(2017, 11, 9, 8, 48, 59)}, {'start': datetime.datetime(2017, 11, 9, 9, 0, 6), 'end': d
atetime.datetime(2017, 11, 9, 9, 9, 24)}, {'start': datetime.datetime(2017, 11, 9, 9, 9,
37), 'end': datetime.datetime(2017, 11, 9, 9, 24, 25)}, {'start': datetime.datetime(201
7, 11, 9, 13, 14, 37), 'end': datetime.datetime(2017, 11, 9, 13, 25, 39)}, {'start': dat
etime.datetime(2017, 11, 9, 15, 20, 7), 'end': datetime.datetime(2017, 11, 9, 15, 31, 1
0)}, {'start': datetime.datetime(2017, 11, 9, 18, 47, 8), 'end': datetime.datetime(2017,
11, 9, 18, 53, 10)}, {'start': datetime.datetime(2017, 11, 9, 23, 35, 2), 'end': datetim
e.datetime(2017, 11, 9, 23, 43, 35)}, {'start': datetime.datetime(2017, 11, 10, 7, 51, 3
3), 'end': datetime.datetime(2017, 11, 10, 8, 2, 28)}, {'start': datetime.datetime(2017,
11, 10, 8, 38, 28), 'end': datetime.datetime(2017, 11, 10, 8, 42, 9)}, {'start': datetim
e.datetime(2017, 11, 11, 18, 5, 25), 'end': datetime.datetime(2017, 11, 11, 18, 13, 1
4)}, {'start': datetime.datetime(2017, 11, 11, 19, 39, 12), 'end': datetime.datetime(201
7, 11, 11, 19, 46, 22)}, {'start': datetime.datetime(2017, 11, 11, 21, 13, 19), 'end': d
atetime.datetime(2017, 11, 11, 21, 16, 31)}, {'start': datetime.datetime(2017, 11, 12,
9, 46, 19), 'end': datetime.datetime(2017, 11, 12, 9, 51, 43)}, {'start': datetime.datet
ime(2017, 11, 13, 13, 33, 42), 'end': datetime.datetime(2017, 11, 13, 13, 54, 15)}, {'st
art': datetime.datetime(2017, 11, 14, 8, 40, 29), 'end': datetime.datetime(2017, 11, 14,
8, 55, 52)}, {'start': datetime.datetime(2017, 11, 15, 6, 14, 5), 'end': datetime.dateti
me(2017, 11, 15, 6, 30, 6)}, {'start': datetime.datetime(2017, 11, 15, 8, 14, 59), 'en
d': datetime.datetime(2017, 11, 15, 8, 23, 44)}, {'start': datetime.datetime(2017, 11, 1
5, 10, 16, 44), 'end': datetime.datetime(2017, 11, 15, 10, 33, 41)}, {'start': datetime.
datetime(2017, 11, 15, 10, 33, 58), 'end': datetime.datetime(2017, 11, 15, 10, 54, 14)},
{'start': datetime.datetime(2017, 11, 15, 11, 2, 15), 'end': datetime.datetime(2017, 11,
15, 11, 14, 42)}, {'start': datetime.datetime(2017, 11, 16, 9, 27, 41), 'end': datetime.
datetime(2017, 11, 16, 9, 38, 49)}, {'start': datetime.datetime(2017, 11, 16, 9, 57, 4
1), 'end': datetime.datetime(2017, 11, 16, 10, 18)}, {'start': datetime.datetime(2017, 1
1, 16, 17, 25, 5), 'end': datetime.datetime(2017, 11, 16, 17, 44, 47)}, {'start': dateti
me.datetime(2017, 11, 17, 13, 45, 54), 'end': datetime.datetime(2017, 11, 17, 16, 36, 5
6)}, {'start': datetime.datetime(2017, 11, 17, 19, 12, 49), 'end': datetime.datetime(201
7, 11, 17, 19, 31, 15)}, {'start': datetime.datetime(2017, 11, 18, 10, 49, 6), 'end': da
tetime.datetime(2017, 11, 18, 10, 55, 45)}, {'start': datetime.datetime(2017, 11, 18, 1
1, 32, 12), 'end': datetime.datetime(2017, 11, 18, 11, 44, 16)}, {'start': datetime.date
time(2017, 11, 18, 18, 9, 1), 'end': datetime.datetime(2017, 11, 18, 18, 14, 31)}, {'sta
rt': datetime.datetime(2017, 11, 18, 18, 53, 10), 'end': datetime.datetime(2017, 11, 18,
19, 1, 29)}, {'start': datetime.datetime(2017, 11, 19, 14, 15, 41), 'end': datetime.date
time(2017, 11, 19, 14, 31, 49)}, {'start': datetime.datetime(2017, 11, 20, 21, 19, 19),
'end': datetime.datetime(2017, 11, 20, 21, 41, 9)}, {'start': datetime.datetime(2017, 1
1, 20, 22, 39, 48), 'end': datetime.datetime(2017, 11, 20, 23, 23, 37)}, {'start': datet
ime.datetime(2017, 11, 21, 17, 44, 25), 'end': datetime.datetime(2017, 11, 21, 17, 51, 3
2)}, {'start': datetime.datetime(2017, 11, 21, 18, 20, 52), 'end': datetime.datetime(201
7, 11, 21, 18, 34, 51)}, {'start': datetime.datetime(2017, 11, 21, 18, 47, 32), 'end': d
atetime.datetime(2017, 11, 21, 18, 51, 50)}, {'start': datetime.datetime(2017, 11, 21, 1
9, 7, 57), 'end': datetime.datetime(2017, 11, 21, 19, 14, 33)}, {'start': datetime.datet
ime(2017, 11, 21, 20, 4, 56), 'end': datetime.datetime(2017, 11, 21, 20, 8, 54)}, {'star
t': datetime.datetime(2017, 11, 21, 21, 55, 47), 'end': datetime.datetime(2017, 11, 21,
22, 8, 12)}, {'start': datetime.datetime(2017, 11, 23, 23, 47, 43), 'end': datetime.date
time(2017, 11, 23, 23, 57, 56)}, {'start': datetime.datetime(2017, 11, 24, 6, 41, 25),
'end': datetime.datetime(2017, 11, 24, 6, 53, 15)}, {'start': datetime.datetime(2017, 1
1, 24, 6, 58, 56), 'end': datetime.datetime(2017, 11, 24, 7, 33, 24)}, {'start': datetim
e.datetime(2017, 11, 26, 12, 25, 49), 'end': datetime.datetime(2017, 11, 26, 12, 41, 3
6)}, {'start': datetime.datetime(2017, 11, 27, 5, 29, 4), 'end': datetime.datetime(2017,
11, 27, 5, 54, 13)}, {'start': datetime.datetime(2017, 11, 27, 6, 6, 47), 'end': datetim
e.datetime(2017, 11, 27, 6, 11, 1)}, {'start': datetime.datetime(2017, 11, 27, 6, 45, 1
4), 'end': datetime.datetime(2017, 11, 27, 6, 55, 39)}, {'start': datetime.datetime(201
7, 11, 27, 9, 39, 44), 'end': datetime.datetime(2017, 11, 27, 9, 47, 43)}, {'start': dat
etime.datetime(2017, 11, 27, 11, 9, 18), 'end': datetime.datetime(2017, 11, 27, 11, 20,
46)}, {'start': datetime.datetime(2017, 11, 27, 11, 31, 46), 'end': datetime.datetime(20
17, 11, 27, 11, 35, 44)}, {'start': datetime.datetime(2017, 11, 27, 12, 7, 14), 'end': d
atetime.datetime(2017, 11, 27, 12, 12, 36)}, {'start': datetime.datetime(2017, 11, 27, 1
2, 21, 40), 'end': datetime.datetime(2017, 11, 27, 12, 26, 44)}, {'start': datetime.date

time(2017, 11, 27, 17, 26, 31), 'end': datetime.datetime(2017, 11, 27, 17, 36, 7)}, {'st
art': datetime.datetime(2017, 11, 27, 18, 11, 49), 'end': datetime.datetime(2017, 11, 2
7, 18, 29, 4)}, {'start': datetime.datetime(2017, 11, 27, 19, 36, 16), 'end': datetime.d
atetime(2017, 11, 27, 19, 47, 17)}, {'start': datetime.datetime(2017, 11, 27, 20, 12, 5
7), 'end': datetime.datetime(2017, 11, 27, 20, 17, 33)}, {'start': datetime.datetime(201
7, 11, 28, 8, 18, 6), 'end': datetime.datetime(2017, 11, 28, 8, 41, 53)}, {'start': date
time.datetime(2017, 11, 28, 19, 17, 23), 'end': datetime.datetime(2017, 11, 28, 19, 34,
1)}, {'start': datetime.datetime(2017, 11, 28, 19, 34, 15), 'end': datetime.datetime(201
7, 11, 28, 19, 46, 24)}, {'start': datetime.datetime(2017, 11, 28, 21, 27, 29), 'end': d
atetime.datetime(2017, 11, 28, 21, 39, 32)}, {'start': datetime.datetime(2017, 11, 29,
7, 47, 38), 'end': datetime.datetime(2017, 11, 29, 7, 51, 18)}, {'start': datetime.datet
ime(2017, 11, 29, 9, 50, 12), 'end': datetime.datetime(2017, 11, 29, 9, 53, 44)}, {'star
t': datetime.datetime(2017, 11, 29, 17, 3, 42), 'end': datetime.datetime(2017, 11, 29, 1
7, 16, 21)}, {'start': datetime.datetime(2017, 11, 29, 18, 19, 15), 'end': datetime.date
time(2017, 11, 29, 18, 23, 43)}, {'start': datetime.datetime(2017, 12, 1, 17, 3, 58), 'e
nd': datetime.datetime(2017, 12, 1, 17, 10, 12)}, {'start': datetime.datetime(2017, 12,
2, 7, 55, 56), 'end': datetime.datetime(2017, 12, 2, 8, 1, 1)}, {'start': datetime.datet
ime(2017, 12, 2, 9, 16, 14), 'end': datetime.datetime(2017, 12, 2, 9, 21, 18)}, {'star
t': datetime.datetime(2017, 12, 2, 19, 48, 29), 'end': datetime.datetime(2017, 12, 2, 1
9, 53, 18)}, {'start': datetime.datetime(2017, 12, 3, 14, 36, 29), 'end': datetime.datet
ime(2017, 12, 3, 15, 20, 9)}, {'start': datetime.datetime(2017, 12, 3, 16, 4, 2), 'end':
datetime.datetime(2017, 12, 3, 16, 25, 30)}, {'start': datetime.datetime(2017, 12, 3, 1
6, 40, 26), 'end': datetime.datetime(2017, 12, 3, 16, 43, 58)}, {'start': datetime.datet
ime(2017, 12, 3, 17, 20, 17), 'end': datetime.datetime(2017, 12, 3, 18, 4, 33)}, {'star
t': datetime.datetime(2017, 12, 4, 8, 34, 24), 'end': datetime.datetime(2017, 12, 4, 8,
51)}, {'start': datetime.datetime(2017, 12, 4, 17, 49, 26), 'end': datetime.datetime(201
7, 12, 4, 17, 53, 57)}, {'start': datetime.datetime(2017, 12, 4, 18, 38, 52), 'end': dat
etime.datetime(2017, 12, 4, 18, 50, 33)}, {'start': datetime.datetime(2017, 12, 4, 21, 3
9, 20), 'end': datetime.datetime(2017, 12, 4, 21, 46, 58)}, {'start': datetime.datetime
(2017, 12, 4, 21, 54, 21), 'end': datetime.datetime(2017, 12, 4, 21, 56, 17)}, {'start':
datetime.datetime(2017, 12, 5, 8, 50, 50), 'end': datetime.datetime(2017, 12, 5, 8, 52,
54)}, {'start': datetime.datetime(2017, 12, 6, 8, 19, 38), 'end': datetime.datetime(201
7, 12, 6, 8, 24, 14)}, {'start': datetime.datetime(2017, 12, 6, 18, 19, 19), 'end': date
time.datetime(2017, 12, 6, 18, 28, 11)}, {'start': datetime.datetime(2017, 12, 6, 18, 2
8, 55), 'end': datetime.datetime(2017, 12, 6, 18, 33, 12)}, {'start': datetime.datetime
(2017, 12, 6, 20, 3, 29), 'end': datetime.datetime(2017, 12, 6, 20, 21, 38)}, {'start':
datetime.datetime(2017, 12, 6, 20, 36, 42), 'end': datetime.datetime(2017, 12, 6, 20, 3
9, 57)}, {'start': datetime.datetime(2017, 12, 7, 5, 54, 51), 'end': datetime.datetime(2
017, 12, 7, 6, 1, 15)}, {'start': datetime.datetime(2017, 12, 8, 16, 47, 18), 'end': dat
etime.datetime(2017, 12, 8, 16, 55, 49)}, {'start': datetime.datetime(2017, 12, 8, 19, 1
5, 2), 'end': datetime.datetime(2017, 12, 8, 19, 29, 12)}, {'start': datetime.datetime(2
017, 12, 9, 22, 39, 37), 'end': datetime.datetime(2017, 12, 9, 22, 47, 19)}, {'start': d
atetime.datetime(2017, 12, 9, 23, 0, 10), 'end': datetime.datetime(2017, 12, 9, 23, 5, 3
2)}, {'start': datetime.datetime(2017, 12, 10, 0, 39, 24), 'end': datetime.datetime(201
7, 12, 10, 0, 56, 2)}, {'start': datetime.datetime(2017, 12, 10, 1, 2, 42), 'end': datet
ime.datetime(2017, 12, 10, 1, 8, 9)}, {'start': datetime.datetime(2017, 12, 10, 1, 8, 5
7), 'end': datetime.datetime(2017, 12, 10, 1, 11, 30)}, {'start': datetime.datetime(201
7, 12, 10, 13, 49, 9), 'end': datetime.datetime(2017, 12, 10, 13, 51, 41)}, {'start': da
tetime.datetime(2017, 12, 10, 15, 14, 29), 'end': datetime.datetime(2017, 12, 10, 15, 1
8, 19)}, {'start': datetime.datetime(2017, 12, 10, 15, 31, 7), 'end': datetime.datetime
(2017, 12, 10, 15, 36, 28)}, {'start': datetime.datetime(2017, 12, 10, 16, 20, 6), 'en
d': datetime.datetime(2017, 12, 10, 16, 30, 31)}, {'start': datetime.datetime(2017, 12,
10, 17, 7, 54), 'end': datetime.datetime(2017, 12, 10, 17, 14, 25)}, {'start': datetime.
datetime(2017, 12, 10, 17, 23, 47), 'end': datetime.datetime(2017, 12, 10, 17, 45, 25)},
{'start': datetime.datetime(2017, 12, 11, 6, 17, 6), 'end': datetime.datetime(2017, 12,
11, 6, 34, 4)}, {'start': datetime.datetime(2017, 12, 11, 9, 8, 41), 'end': datetime.dat
etime(2017, 12, 11, 9, 12, 21)}, {'start': datetime.datetime(2017, 12, 11, 9, 15, 41),
'end': datetime.datetime(2017, 12, 11, 9, 20, 18)}, {'start': datetime.datetime(2017, 1
2, 12, 8, 55, 53), 'end': datetime.datetime(2017, 12, 12, 8, 59, 34)}, {'start': datetim
e.datetime(2017, 12, 13, 17, 14, 56), 'end': datetime.datetime(2017, 12, 13, 17, 18, 3

2)}, {'start': datetime.datetime(2017, 12, 13, 18, 52, 16), 'end': datetime.datetime(201
7, 12, 13, 19, 0, 45)}, {'start': datetime.datetime(2017, 12, 14, 9, 1, 10), 'end': date
time.datetime(2017, 12, 14, 9, 11, 6)}, {'start': datetime.datetime(2017, 12, 14, 9, 12,
59), 'end': datetime.datetime(2017, 12, 14, 9, 19, 6)}, {'start': datetime.datetime(201
7, 12, 14, 11, 54, 33), 'end': datetime.datetime(2017, 12, 14, 12, 2)}, {'start': dateti
me.datetime(2017, 12, 14, 14, 40, 23), 'end': datetime.datetime(2017, 12, 14, 14, 44, 4
0)}, {'start': datetime.datetime(2017, 12, 14, 15, 8, 55), 'end': datetime.datetime(201
7, 12, 14, 15, 26, 24)}, {'start': datetime.datetime(2017, 12, 14, 17, 46, 17), 'end': d
atetime.datetime(2017, 12, 14, 18, 9, 4)}, {'start': datetime.datetime(2017, 12, 15, 9,
8, 12), 'end': datetime.datetime(2017, 12, 15, 9, 23, 45)}, {'start': datetime.datetime
(2017, 12, 16, 9, 33, 46), 'end': datetime.datetime(2017, 12, 16, 9, 36, 17)}, {'start':
datetime.datetime(2017, 12, 16, 11, 2, 31), 'end': datetime.datetime(2017, 12, 16, 11,
5, 4)}, {'start': datetime.datetime(2017, 12, 17, 10, 9, 47), 'end': datetime.datetime(2
017, 12, 17, 10, 32, 3)}, {'start': datetime.datetime(2017, 12, 18, 8, 2, 36), 'end': da
tetime.datetime(2017, 12, 18, 8, 7, 34)}, {'start': datetime.datetime(2017, 12, 18, 16,
3), 'end': datetime.datetime(2017, 12, 18, 16, 9, 20)}, {'start': datetime.datetime(201
7, 12, 18, 16, 30, 7), 'end': datetime.datetime(2017, 12, 18, 16, 53, 12)}, {'start': da
tetime.datetime(2017, 12, 18, 19, 18, 23), 'end': datetime.datetime(2017, 12, 18, 19, 2
2, 8)}, {'start': datetime.datetime(2017, 12, 18, 20, 14, 46), 'end': datetime.datetime
(2017, 12, 18, 20, 17, 47)}, {'start': datetime.datetime(2017, 12, 19, 19, 14, 8), 'en
d': datetime.datetime(2017, 12, 19, 19, 23, 49)}, {'start': datetime.datetime(2017, 12,
19, 19, 39, 36), 'end': datetime.datetime(2017, 12, 19, 19, 43, 46)}, {'start': datetim
e.datetime(2017, 12, 20, 8, 5, 14), 'end': datetime.datetime(2017, 12, 20, 8, 10, 46)},
{'start': datetime.datetime(2017, 12, 20, 8, 15, 45), 'end': datetime.datetime(2017, 12,
20, 8, 29, 50)}, {'start': datetime.datetime(2017, 12, 20, 8, 33, 32), 'end': datetime.d
atetime(2017, 12, 20, 8, 38, 9)}, {'start': datetime.datetime(2017, 12, 20, 13, 43, 36),
'end': datetime.datetime(2017, 12, 20, 13, 54, 39)}, {'start': datetime.datetime(2017, 1
2, 20, 18, 57, 53), 'end': datetime.datetime(2017, 12, 20, 19, 6, 54)}, {'start': dateti
me.datetime(2017, 12, 21, 7, 21, 11), 'end': datetime.datetime(2017, 12, 21, 7, 32, 3)},
{'start': datetime.datetime(2017, 12, 21, 8, 1, 58), 'end': datetime.datetime(2017, 12,
21, 8, 6, 15)}, {'start': datetime.datetime(2017, 12, 21, 13, 20, 54), 'end': datetime.d
atetime(2017, 12, 21, 13, 33, 49)}, {'start': datetime.datetime(2017, 12, 21, 15, 26,
8), 'end': datetime.datetime(2017, 12, 21, 15, 34, 27)}, {'start': datetime.datetime(201
7, 12, 21, 18, 9, 46), 'end': datetime.datetime(2017, 12, 21, 18, 38, 50)}, {'start': da
tetime.datetime(2017, 12, 22, 16, 14, 21), 'end': datetime.datetime(2017, 12, 22, 16, 2
1, 46)}, {'start': datetime.datetime(2017, 12, 22, 16, 29, 17), 'end': datetime.datetime
(2017, 12, 22, 16, 34, 14)}, {'start': datetime.datetime(2017, 12, 25, 12, 49, 51), 'en
d': datetime.datetime(2017, 12, 25, 13, 18, 27)}, {'start': datetime.datetime(2017, 12,
25, 13, 46, 44), 'end': datetime.datetime(2017, 12, 25, 14, 20, 50)}, {'start': datetim
e.datetime(2017, 12, 26, 10, 40, 16), 'end': datetime.datetime(2017, 12, 26, 10, 53, 4
5)}, {'start': datetime.datetime(2017, 12, 27, 16, 56, 12), 'end': datetime.datetime(201
7, 12, 27, 17, 17, 39)}, {'start': datetime.datetime(2017, 12, 29, 6, 2, 34), 'end': dat
etime.datetime(2017, 12, 29, 6, 12, 30)}, {'start': datetime.datetime(2017, 12, 29, 12,
21, 3), 'end': datetime.datetime(2017, 12, 29, 12, 46, 16)}, {'start': datetime.datetime
(2017, 12, 29, 14, 32, 55), 'end': datetime.datetime(2017, 12, 29, 14, 43, 46)}, {'star
t': datetime.datetime(2017, 12, 29, 15, 8, 26), 'end': datetime.datetime(2017, 12, 29, 1
5, 18, 51)}, {'start': datetime.datetime(2017, 12, 29, 20, 33, 34), 'end': datetime.date
time(2017, 12, 29, 20, 38, 13)}, {'start': datetime.datetime(2017, 12, 30, 13, 51, 3),
'end': datetime.datetime(2017, 12, 30, 13, 54, 33)}, {'start': datetime.datetime(2017, 1
2, 30, 15, 9, 3), 'end': datetime.datetime(2017, 12, 30, 15, 19, 13)}]

Recreating ISO format with strftime()

We used strftime() to create strings from date objects. Now that we know about datetime objects, let's practice doing something similar.

Re-create the .isoformat() method, using .strftime(), and print the first trip start in our data set.

In [30]:
```
# Import datetime
from datetime import datetime

# Pull out the start of the first trip
first_start = onebike_datetimes[0]['start']

# Format to feed to strftime()
fmt = "%Y-%m-%dT%H:%M:%S"

# Print out date with .isoformat(), then with .strftime() to compare
print(first_start.isoformat())
print(datetime.strftime(first_start, fmt))
print(first_start.strftime(fmt))
```

```
2017-10-01T15:23:25
2017-10-01T15:23:25
2017-10-01T15:23:25
```

Unix timestamps

Datetimes are sometimes stored as Unix timestamps: the number of seconds since January 1, 1970. This is especially common with computer infrastructure, like the log files that websites keep when they get visitors.

In [31]:
```
# Import datetime
from datetime import datetime

# Starting timestamps
timestamps = [1514665153, 1514664543]

# Datetime objects
dts = []

# Loop
for ts in timestamps:
    dts.append(datetime.fromtimestamp(ts))

# Print results
print(dts)
```

```
[datetime.datetime(2017, 12, 31, 1, 49, 13), datetime.datetime(2017, 12, 31, 1, 39, 3)]
```

↥ back to top

Working with durations

Much like dates, we can add/subtract/work with intervals in the datetime object as well.

In [32]:
```
start = datetime(2017, 10, 1, 15, 23, 25)
end = datetime(2017, 10, 1, 15, 26, 26)

duration = end - start

# subtract datetimes to create timedelta
print(duration)
print(duration.total_seconds())
```

```
0:03:01
181.0
```

In [33]:
```
# import timedelta
from datetime import timedelta

# create a timedelta
delta1 = timedelta(seconds=1)

print(start,'\t',start+delta1)
```

```
2017-10-01 15:23:25	2017-10-01 15:23:26
```

In [34]:
```
# create a timedelta
delta1 = timedelta(weeks=1, seconds=-100)

print(start,'\t',start+delta1)
```

```
2017-10-01 15:23:25	2017-10-08 15:21:45
```

Turning pairs of datetimes into durations

When working with timestamps, we often want to know how much time has elapsed between events. Thankfully, we can use datetime arithmetic to ask Python to do the heavy lifting for us so we don't need to worry about day, month, or year boundaries. Let's calculate the number of seconds that the bike was out of the dock for each trip..

In [35]:
```
# Initialize a list for all the trip durations
onebike_durations = []

for trip in onebike_datetimes:
    # Create a timedelta object corresponding to the length of the trip
    trip_duration = trip['end'] - trip['start']

    # Get the total elapsed seconds in trip_duration
    trip_length_seconds = trip_duration.total_seconds()

    # Append the results to our list
    onebike_durations.append(trip_length_seconds)

print(onebike_durations)
```

```
[181.0, 7622.0, 343.0, 1278.0, 1277.0, 1366.0, 815.0, 545.0, 491.0, 639.0, 1678.0, 406.
0, 709.0, 514.0, 492.0, 1668.0, 2242.0, 2752.0, 735.0, 330.0, 518.0, 1433.0, 204.0, 304.
0, 977.0, 1399.0, 1244.0, 658.0, 800.0, 1911.0, 2471.0, 1344.0, 435.0, 271.0, 920.0, 85
1.0, 209.0, 453.0, 841.0, 142.0, 1023.0, 1466.0, 1636.0, 3039.0, 1571.0, 1410.0, 386.0,
1527.0, 622.0, 1450.0, 1422.0, 991.0, 1484.0, 1450.0, 929.0, 533.0, 525.0, 283.0, 133.0,
1106.0, 952.0, 553.0, 659.0, 297.0, 357.0, 989.0, 979.0, 760.0, 1110.0, 675.0, 1207.0, 1
593.0, 768.0, 1446.0, 485.0, 200.0, 399.0, 242.0, 170.0, 450.0, 1078.0, 1042.0, 573.0, 7
48.0, 735.0, 336.0, 76913.0, 171.0, 568.0, 358.0, 917.0, 671.0, 1791.0, 318.0, 888.0, 12
84.0, 11338.0, 1686.0, 5579.0, 8290.0, 1850.0, 1810.0, 870.0, 436.0, 429.0, 494.0, 1439.
0, 380.0, 629.0, 962.0, 387.0, 952.0, 190.0, 739.0, 1120.0, 369.0, 2275.0, 873.0, 1670.
0, 643.0, 572.0, 1375.0, 725.0, 688.0, 1041.0, 1707.0, 1236.0, 1291.0, 2890.0, -3346.0,
1213.0, 331.0, 1497.0, 527.0, 584.0, 2599.0, 759.0, 1291.0, 916.0, 161.0, 806.0, 838.0,
644.0, 374.0, 678.0, 137.0, 659.0, 386.0, 745.0, 448.0, 558.0, 888.0, 662.0, 663.0, 362.
```

```
0, 513.0, 655.0, 221.0, 469.0, 430.0, 192.0, 324.0, 1233.0, 923.0, 961.0, 525.0, 1017.0,
1216.0, 747.0, 668.0, 1219.0, 1182.0, 10262.0, 1106.0, 399.0, 724.0, 330.0, 499.0, 968.
0, 1310.0, 2629.0, 427.0, 839.0, 258.0, 396.0, 238.0, 745.0, 613.0, 710.0, 2068.0, 947.
0, 1509.0, 254.0, 625.0, 479.0, 688.0, 238.0, 322.0, 304.0, 576.0, 1035.0, 661.0, 276.0,
1427.0, 998.0, 729.0, 723.0, 220.0, 212.0, 759.0, 268.0, 374.0, 305.0, 304.0, 289.0, 262
0.0, 1288.0, 212.0, 2656.0, 996.0, 271.0, 701.0, 458.0, 116.0, 124.0, 276.0, 532.0, 257.
0, 1089.0, 195.0, 384.0, 511.0, 850.0, 462.0, 322.0, 998.0, 327.0, 153.0, 152.0, 230.0,
321.0, 625.0, 391.0, 1298.0, 1018.0, 220.0, 277.0, 221.0, 216.0, 509.0, 596.0, 367.0, 44
7.0, 257.0, 1049.0, 1367.0, 933.0, 151.0, 153.0, 1336.0, 298.0, 380.0, 1385.0, 225.0, 18
1.0, 581.0, 250.0, 332.0, 845.0, 277.0, 663.0, 541.0, 652.0, 257.0, 775.0, 499.0, 1744.
0, 445.0, 297.0, 1716.0, 2046.0, 809.0, 1287.0, 596.0, 1513.0, 651.0, 625.0, 279.0, 210.
0, 610.0]
```

Average trip time

How long were the trips on average? We can use the built-in Python functions sum() and len() to make this calculation.

Based on your last coding exercise, the data has been loaded as onebike_durations. Each entry is a number of seconds that the bike was out of the dock.

In [36]:
```
# What was the total duration of all trips?
total_elapsed_time = sum(onebike_durations)

# What was the total number of trips?
number_of_trips = len(onebike_durations)

# Divide the total duration by the number of trips
print(total_elapsed_time / number_of_trips)
```

```
1178.9310344827586
```

The long and the short of why time is hard

Out of 291 trips taken, how long was the longest? How short was the shortest? Does anything look fishy?

As before, data has been loaded as onebike_durations

In [37]:
```
# Calculate shortest and longest trips
shortest_trip = min(onebike_durations)
longest_trip = max(onebike_durations)

# Print out the results
print("The shortest trip was {} seconds".format(shortest_trip))
print("The longest trip was {lt} seconds".format(lt=longest_trip))
```

```
The shortest trip was -3346.0 seconds
The longest trip was 76913.0 seconds
```

↥ back to top

Great Job!