

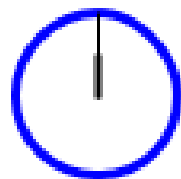
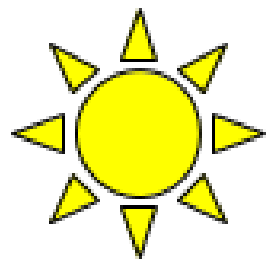
UTC offsets

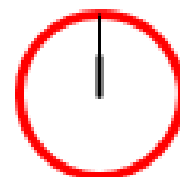
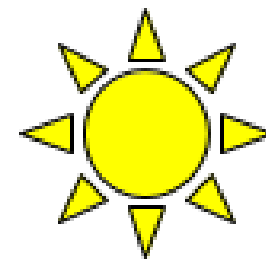
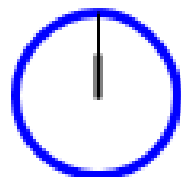
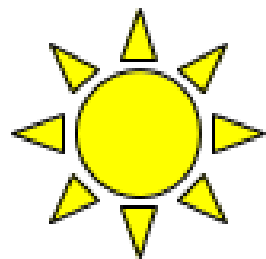
WORKING WITH DATES AND TIMES IN PYTHON

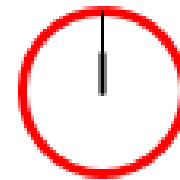
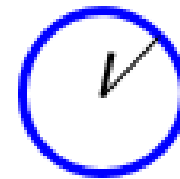
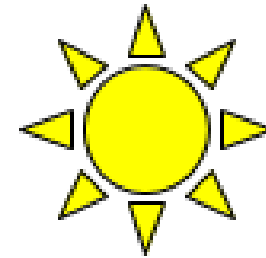
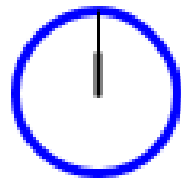
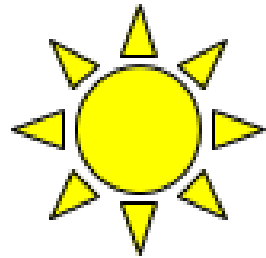


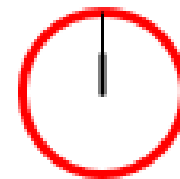
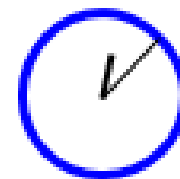
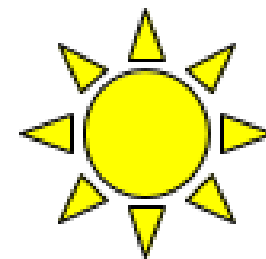
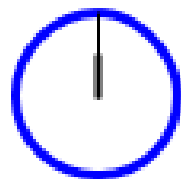
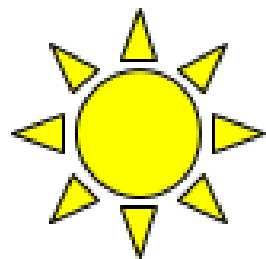
Max Shron

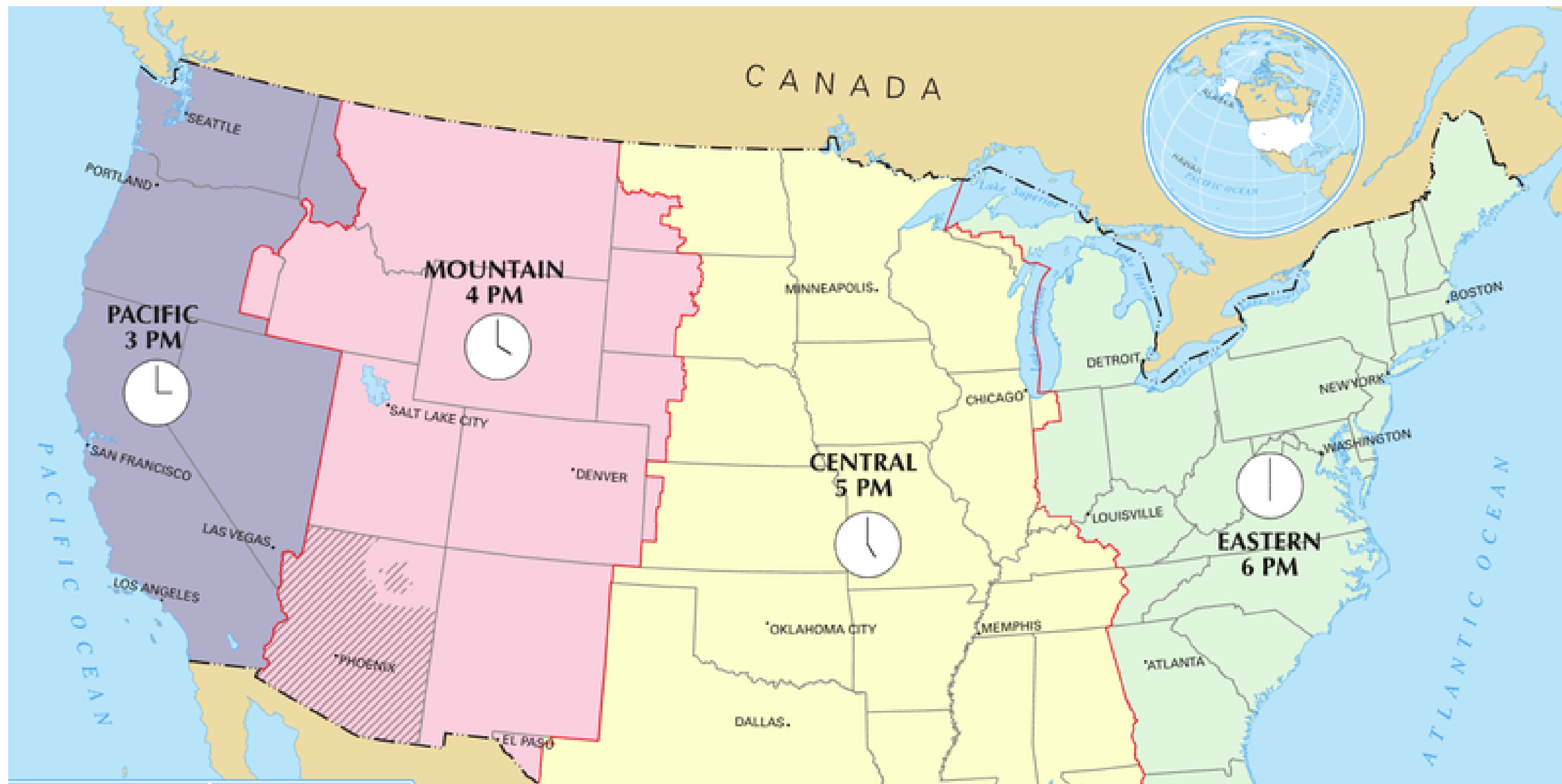
Data Scientist and Author

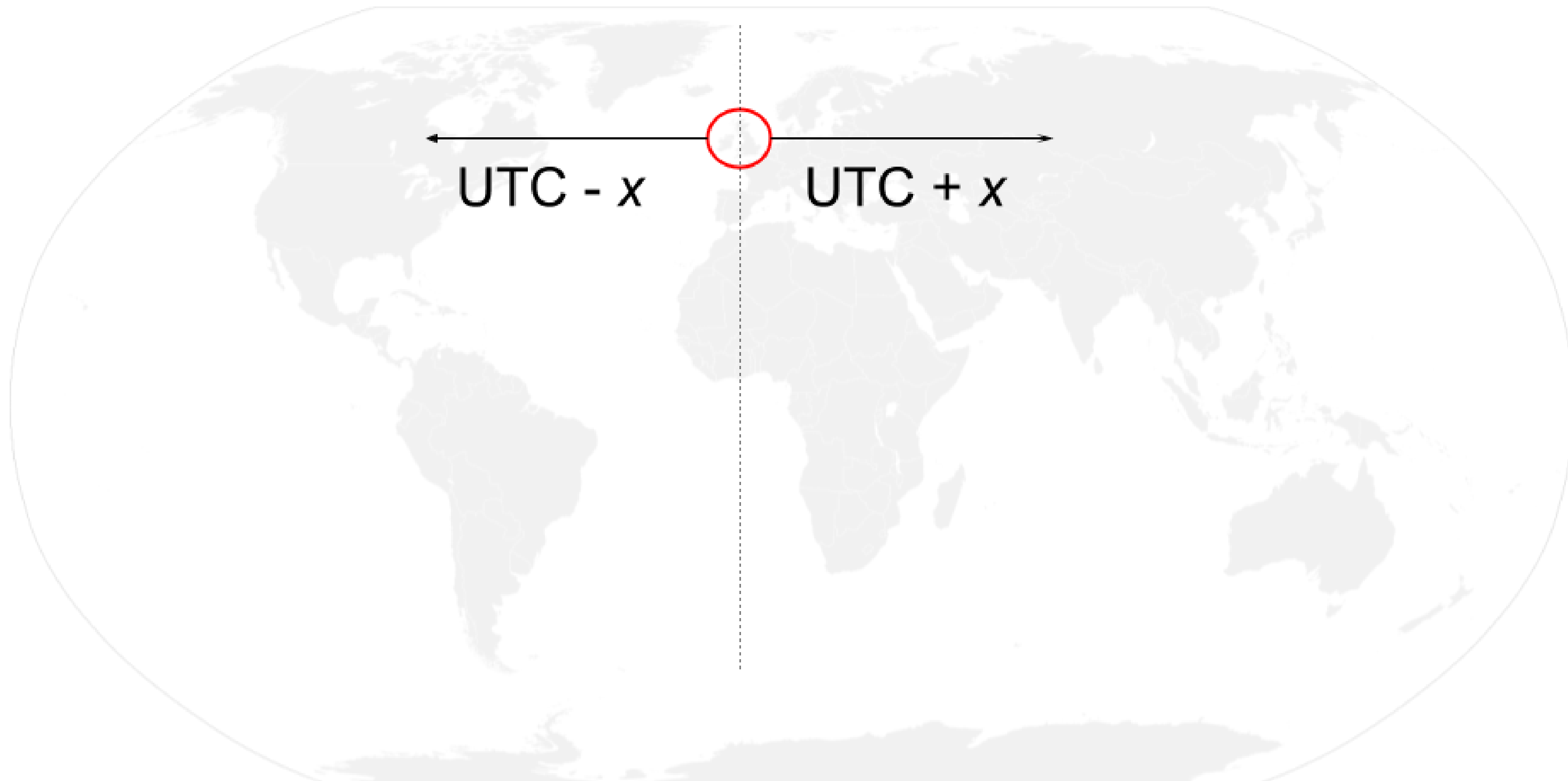












UTC

```
# Import relevant classes  
from datetime import datetime, timedelta, timezone
```


UTC

```
# Import relevant classes
from datetime import datetime, timedelta, timezone

# US Eastern Standard time zone
ET = timezone(timedelta(hours=-5))

# Timezone-aware datetime
dt = datetime(2017, 12, 30, 15, 9, 3, tzinfo = ET)
```

```
print(dt)
```

```
'2017-12-30 15:09:03-05:00'
```

UTC

```
# India Standard time zone
IST = timezone(timedelta(hours=5, minutes=30))

# Convert to IST
print(dt.astimezone(IST))
```

```
'2017-12-31 01:39:03+05:30'
```

Adjusting timezone vs changing tzinfo

```
print(dt)
```

```
'2017-12-30 15:09:03-05:00'
```

```
print(dt.replace(tzinfo=timezone.utc))
```

```
'2017-12-30 15:09:03+00:00'
```

```
# Change original to match UTC  
print(dt.astimezone(timezone.utc))
```

```
'2017-12-30 20:09:03+00:00'
```

UTC Offsets

WORKING WITH DATES AND TIMES IN PYTHON

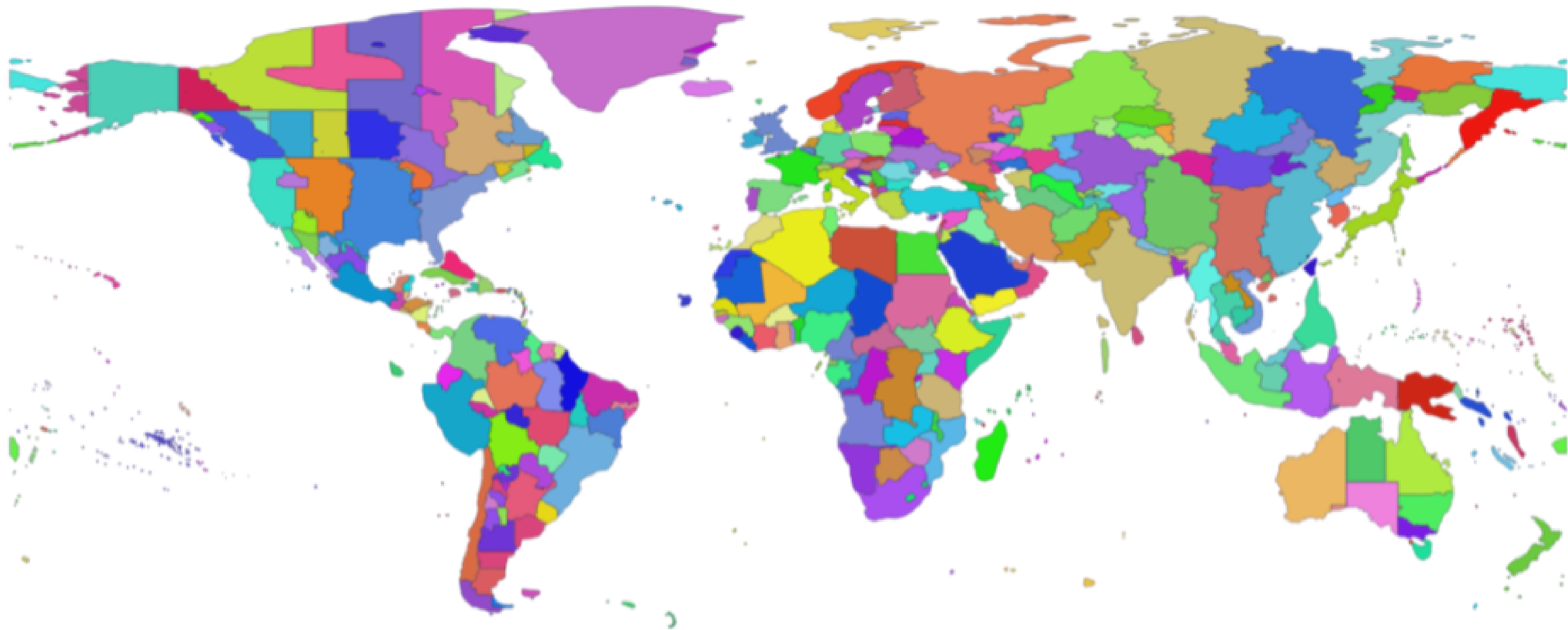
Time zone database

WORKING WITH DATES AND TIMES IN PYTHON



Max Shron

Data Scientist and Author



Time zone database

```
# Imports
from datetime import datetime
from dateutil import tz
```

tz database

Time zone database

```
# Imports
from datetime import datetime
from dateutil import tz

# Eastern time
et = tz.gettz('America/New_York')
```

tz database

- Format: 'Continent/City'

Time zone database

```
# Imports
from datetime import datetime
from dateutil import tz

# Eastern time
et = tz.gettz('America/New_York')
```

tz database

- Format: 'Continent/City'
- Examples:
 - 'America/New_York'
 - 'America/Mexico_City'
 - 'Europe/London'
 - 'Africa/Accra'

Time zone database

```
# Last ride  
last = datetime(2017, 12, 30, 15, 9, 3, tzinfo=et)
```

```
print(last)
```

```
'2017-12-30 15:09:03-05:00'
```

Time zone database

```
# Last ride  
last = datetime(2017, 12, 30, 15, 9, 3, tzinfo=et)  
print(last)
```

```
'2017-12-30 15:09:03-05:00'
```

```
# First ride  
first = datetime(2017, 10, 1, 15, 23, 25, tzinfo=et)  
print(first)
```

```
'2017-10-01 15:23:25-04:00'
```

Time zone database

WORKING WITH DATES AND TIMES IN PYTHON

Starting Daylight Saving Time

WORKING WITH DATES AND TIMES IN PYTHON

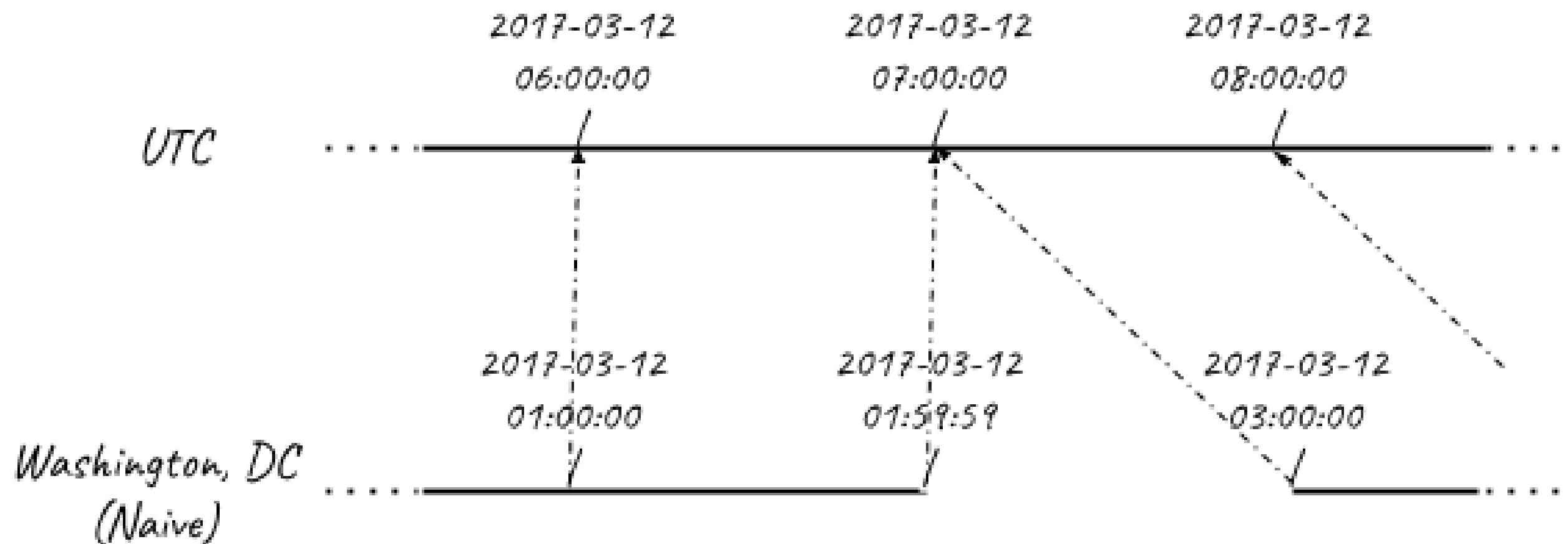


Max Shron
Data Scientist and Author

Washington, DC (Naive)

2017-03-12	2017-03-12	2017-03-12
01:00:00	01:59:59	03:00:00

Timeline diagram showing a jump in time on 2017-03-12. The timeline consists of three horizontal segments. The first segment starts with an ellipsis and ends at a vertical tick mark labeled '01:00:00' above the date '2017-03-12'. The second segment starts at a vertical tick mark labeled '01:59:59' above the date '2017-03-12' and ends at another vertical tick mark. The third segment starts at a vertical tick mark labeled '03:00:00' above the date '2017-03-12' and ends with an ellipsis. There is a significant gap between the end of the second segment and the start of the third segment, representing the jump in time.



Start of Daylight Saving Time

```
spring_ahead_159am = datetime(2017, 3, 12, 1, 59, 59)
spring_ahead_159am.isoformat()
```

```
'2017-03-12T01:59:59'
```

```
spring_ahead_3am = datetime(2017, 3, 12, 3, 0, 0)
spring_ahead_3am.isoformat()
```

```
'2017-03-12T03:00:00'
```

```
(spring_ahead_3am - spring_ahead_159am).total_seconds()
```

```
3601
```


Start of Daylight Saving Time

```
from datetime import timezone, timedelta
```

```
EST = timezone(timedelta(hours=-5))
```

```
EDT = timezone(timedelta(hours=-4))
```

Start of Daylight Saving Time

```
spring_ahead_159am = spring_ahead_159am.replace(tzinfo = EST)
spring_ahead_159am.isoformat()
```

```
'2017-03-12T01:59:59-05:00'
```

```
spring_ahead_3am = spring_ahead_159am.replace(tzinfo = EDT)
spring_ahead_3am.isoformat()
```

```
'2017-03-12T03:00:00-04:00'
```

```
(spring_ahead_3am - spring_ahead_159am).seconds
```

```
1
```

Start of Daylight Saving Time

Using `dateutil`

```
# Import tz
from dateutil import tz

# Create eastern timezone
eastern = tz.gettz('America/New_York')

# 2017-03-12 01:59:59 in Eastern Time (EST)
spring_ahead_159am = datetime(2017, 3, 12, 1, 59, 59,
                               tzinfo = eastern)

# 2017-03-12 03:00:00 in Eastern Time (EDT)
spring_ahead_3am = datetime(2017, 3, 12, 3, 0, 0,
                             tzinfo = eastern)
```

Daylight Saving

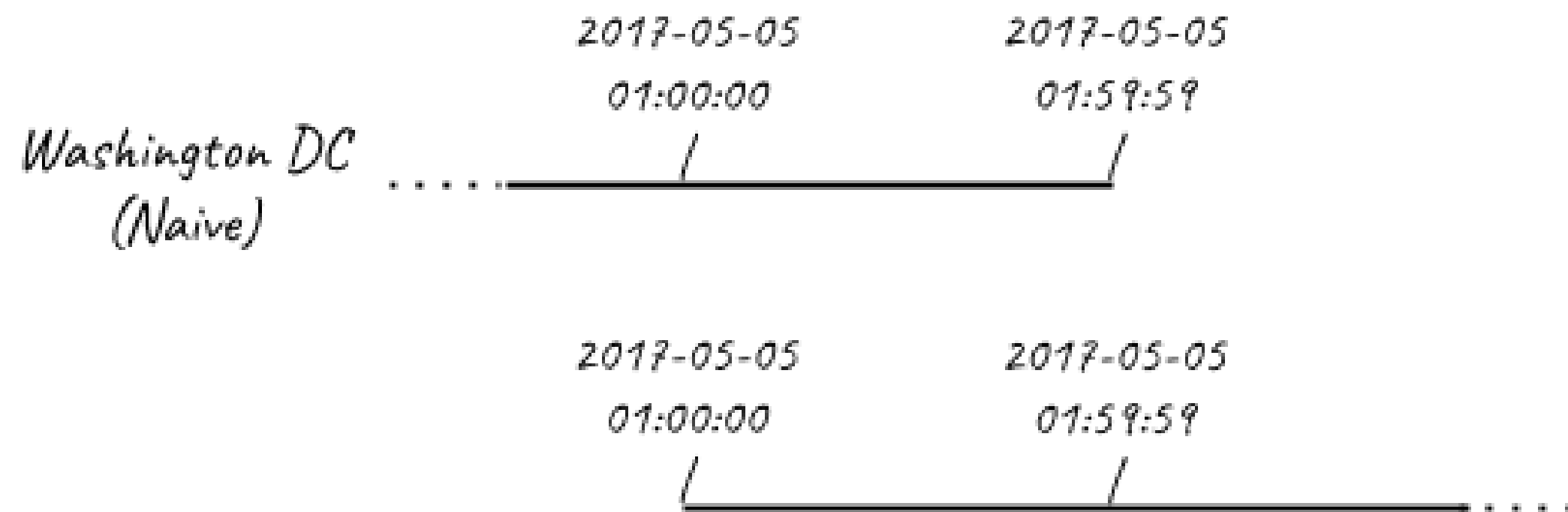
WORKING WITH DATES AND TIMES IN PYTHON

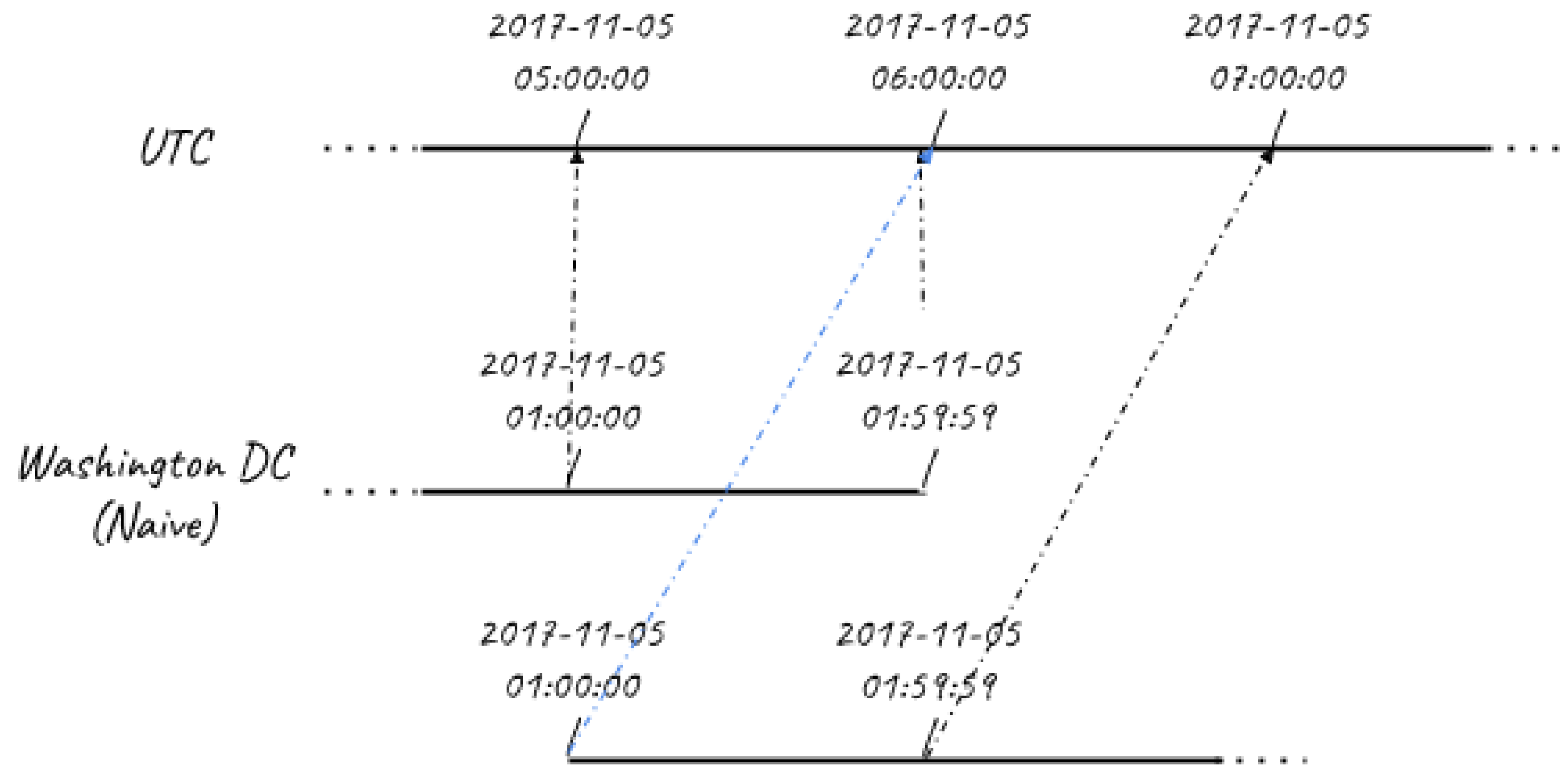
Ending Daylight Saving Time

WORKING WITH DATES AND TIMES IN PYTHON



Max Shron
Data Scientist and Author





Ending Daylight Saving Time

```
eastern = tz.gettz('US/Eastern')
# 2017-11-05 01:00:00
first_1am = datetime(2017, 11, 5, 1, 0, 0,
                     tzinfo = eastern)
tz.datetime_ambiguous(first_1am)
```

True

```
# 2017-11-05 01:00:00 again
second_1am = datetime(2017, 11, 5, 1, 0, 0,
                      tzinfo = eastern)
second_1am = tz.enfold(second_1am)
```


Ending Daylight Saving Time

```
(first_1am - second_1am).total_seconds()
```

```
0.0
```

```
first_1am = first_1am.astimezone(tz.UTC)  
second_1am = second_1am.astimezone(tz.UTC)  
(first_1am - second_1am).total_seconds()
```

```
3600.0
```

Ending Daylight Saving Time

WORKING WITH DATES AND TIMES IN PYTHON