

# Dates in Python

WORKING WITH DATES AND TIMES IN PYTHON



**Max Shron**

Data Scientist and Author

# Course overview

- Chapter 1: Dates and Calendars
- Chapter 2: Combining Dates and Times
- Chapter 3: Time zones and Daylight Saving
- Chapter 4: Dates and Times in Pandas

# Dates in Python

*string*  
"Hello"

*array*  

1	2	3	4	5	6
---	---	---	---	---	---

*number*  
42

*date*  


# Why do we need a date class in Python?

```
two_hurricanes = ["10/7/2016", "6/21/2017"]
```

How would you:

- Figure out how many days had elapsed?
- Check that they were in order from earliest to latest?
- Know which day of the week each was?
- Filter out hurricanes which happened between certain dates?

# Creating date objects

```
# Import date
from datetime import date

# Create dates
two_hurricanes_dates = [date(2016, 10, 7), date(2017, 6, 21)]
```

# Attributes of a date

```
# Import date
from datetime import date

# Create dates
two_hurricanes_dates = [date(2016, 10, 7), date(2017, 6, 21)]

print(two_hurricanes_dates[0].year)
print(two_hurricanes_dates[0].month)
print(two_hurricanes_dates[0].day)
```

2016

10

7

# Finding the weekday of a date

```
print(two_hurricanes_dates[0].weekday())
```

4

- Weekdays in Python
  - 0 = Monday
  - 1 = Tuesday
  - 2 = Wednesday
  - ...
  - 6 = Sunday

# Dates in Python

WORKING WITH DATES AND TIMES IN PYTHON



# Math with Dates

WORKING WITH DATES AND TIMES IN PYTHON



**Max Shron**

Data Scientist and Author

# Math with dates

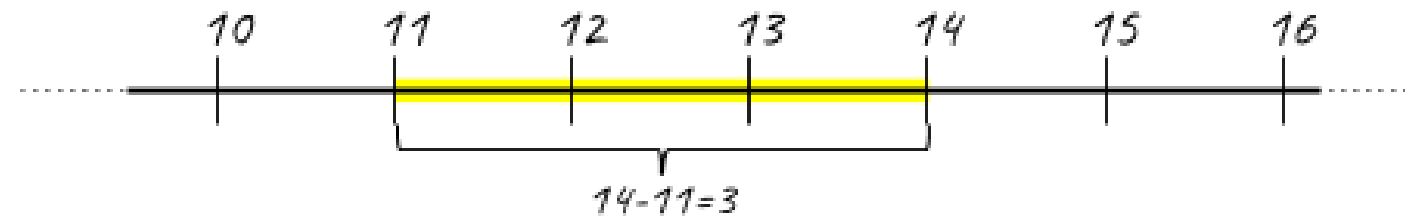


```
# Example numbers
a = 11
b = 14
l = [a, b]

# Find the least least in the list
print(min(l))

11
```

# Math with dates



```
# Subtract two numbers
```

```
print(b - a)
```

```
3
```

```
# Add 3 to a
```

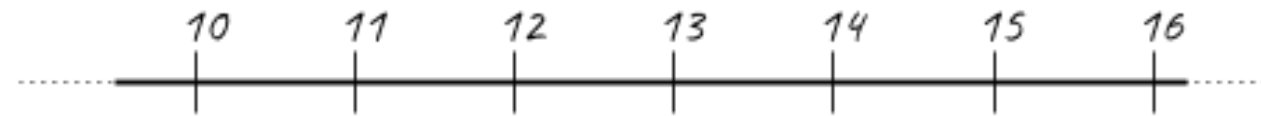
```
print(a + 3)
```

```
14
```

# Math with dates



# Math with dates

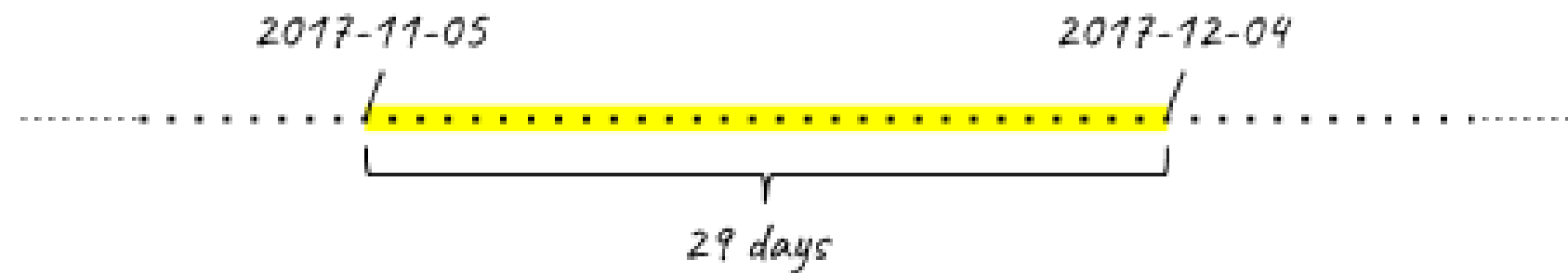


```
# Import date
from datetime import date

# Create our dates
d1 = date(2017, 11, 5)
d2 = date(2017, 12, 4)
l = [d1, d2]

print(min(l))
2017-11-05
```

# Math with dates



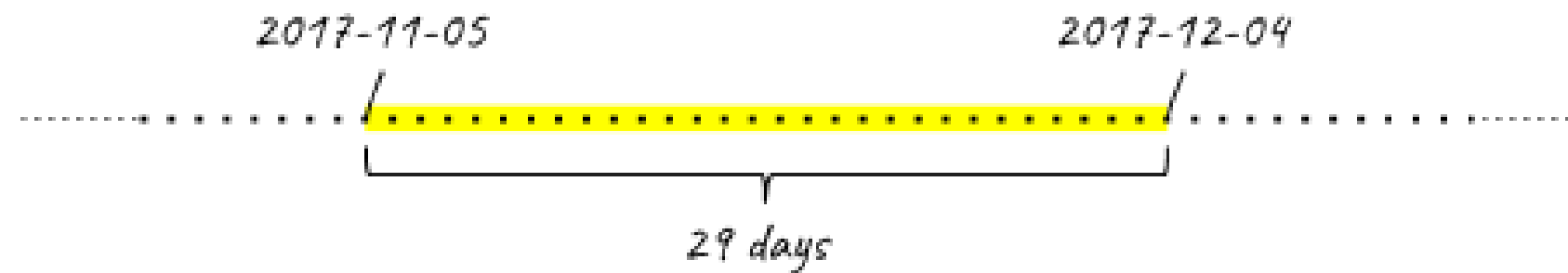
```
# Subtract two dates
```

```
delta = d2 - d1
```

```
print(delta.days)
```

```
29
```

# Math with dates



```
# Import timedelta
from datetime import timedelta

# Create a 29 day timedelta
td = timedelta(days=29)
print(d1 + td)
2017-12-04
```

# Incrementing variables with +=

```
# Initialize x to be zero
```

```
x = 0
```

```
# Increment x
```

```
x = x + 1
```

```
print(x)
```

```
1
```

```
# Initialize x to be zero
```

```
x = 0
```

```
# Increment x
```

```
x += 1
```

```
print(x)
```

```
1
```



# Let's Practice!

WORKING WITH DATES AND TIMES IN PYTHON

# Turning dates into strings

WORKING WITH DATES AND TIMES IN PYTHON



**Max Shron**  
Data Scientist and Author

# ISO 8601 format

```
from datetime import date

# Example date
d = date(2017, 11, 5)

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

```
2017-11-05
```

```
# Express the date in ISO 8601 format and put it in a list
print( [d.isoformat()] )
```

```
[ '2017-11-05' ]
```

# ISO 8601 format

```
# A few dates that computers once had trouble with
some_dates = ['2000-01-01', '1999-12-31']

# Print them in order
print(sorted(some_dates))
```

```
['1999-12-31', '2000-01-01']
```

# Every other format

```
d.strftime()
```

# Every other format: strftime

```
# Example date  
d = date(2017, 1, 5)  
  
print(d.strftime("%Y"))
```

```
2017
```

```
# Format string with more text in it  
print(d.strftime("Year is %Y"))
```

```
Year is 2017
```

# Every other format: strftime

```
# Format: YYYY/MM/DD  
print(d.strftime("%Y/%m/%d"))
```

```
2017/01/05
```

# Turning dates into strings

WORKING WITH DATES AND TIMES IN PYTHON