

[301] CSV Files

Tyler Caraza-Harter

Learning Objectives Today

CSV format

- purpose
- syntax
- comparison to spreadsheet

Reading CSV files

- without header
- with header
- type casting

Chapter 14 of Sweigart,
to (and including) “Reading Data
from Reader Objects in a for
Loop”

<https://automatetheboringstuff.com/chapter14/>

Today's Outline

Spreadsheets

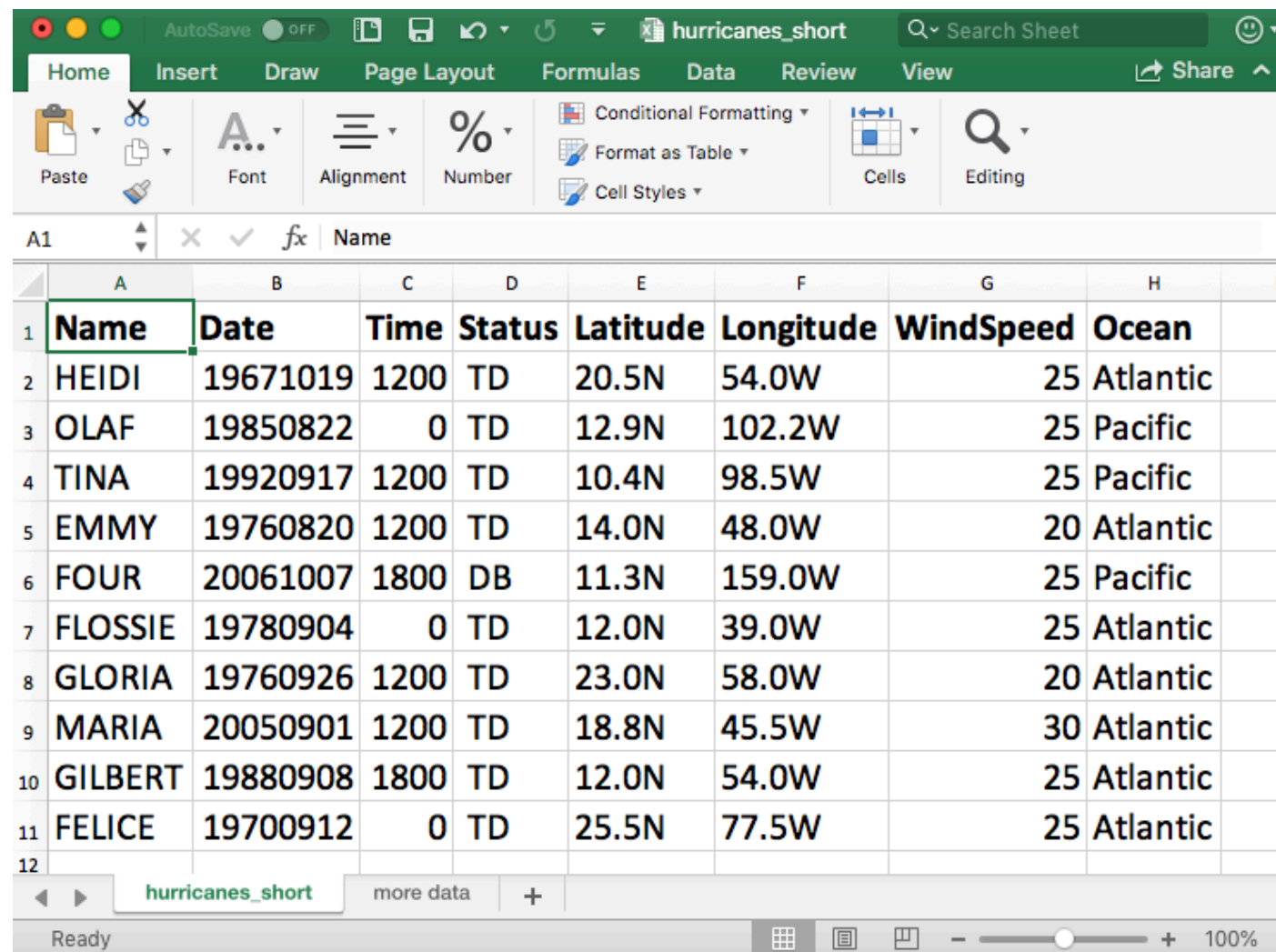
CSVs

Reading a CSV to a list of lists

Coding examples

Spreadsheets (e.g., Excel)

Spreadsheets are tables of cells, organized by rows and columns



The image shows a screenshot of the Microsoft Excel application. The title bar indicates the file name is "hurricanes_short". The ribbon is set to "Home", and the "Cells" group is active. The spreadsheet contains a table with 11 rows of hurricane data. The columns are labeled: Name, Date, Time, Status, Latitude, Longitude, WindSpeed, and Ocean. The data is as follows:

	A	B	C	D	E	F	G	H	I
1	Name	Date	Time	Status	Latitude	Longitude	WindSpeed	Ocean	
2	HEIDI	19671019	1200	TD	20.5N	54.0W	25	Atlantic	
3	OLAF	19850822	0	TD	12.9N	102.2W	25	Pacific	
4	TINA	19920917	1200	TD	10.4N	98.5W	25	Pacific	
5	EMMY	19760820	1200	TD	14.0N	48.0W	20	Atlantic	
6	FOUR	20061007	1800	DB	11.3N	159.0W	25	Pacific	
7	FLOSSIE	19780904	0	TD	12.0N	39.0W	25	Atlantic	
8	GLORIA	19760926	1200	TD	23.0N	58.0W	20	Atlantic	
9	MARIA	20050901	1200	TD	18.8N	45.5W	30	Atlantic	
10	GILBERT	19880908	1800	TD	12.0N	54.0W	25	Atlantic	
11	FELICE	19700912	0	TD	25.5N	77.5W	25	Atlantic	
12									

The status bar at the bottom shows "Ready" and a zoom level of 100%.

Spreadsheets (e.g., Excel)

Spreadsheets often allow different **data types**

hurricanes_short

	A	B	C	D	E	F	G	H	I
1	Name	Date	Time	Status	Latitude	Longitude	WindSpeed	Ocean	
2	HEIDI	19671019	1200	TD	20.5N	54.0W	25	Atlantic	
3	OLAF	19850822	0	TD	12.9N	102.2W	25	Pacific	
4	TINA	19920917	1200	TD	10.4N	98.5W	25	Pacific	
5	EMMY	19760820	1200	TD	14.0N	48.0W	20	Atlantic	
6	FOUR	20061007	1800	DB	11.3N	159.0W	25	Pacific	
7	FLOSSIE	19780904	0	TD	12.0N	39.0W	25	Atlantic	
8	GLORIA	19760926	1200	TD	23.0N	58.0W	20	Atlantic	
9	MARIA	20050901	1200	TD	18.8N	45.5W	30	Atlantic	
10	GILBERT	19880908	1800	TD	12.0N	54.0W	25	Atlantic	
11	FELICE	19700912	0	TD	25.5N	77.5W	25	Atlantic	
12									

hurricanes_short more data +

Ready 100%

text

numbers

Spreadsheets (e.g., Excel)

Spreadsheets often allow different **fonts**

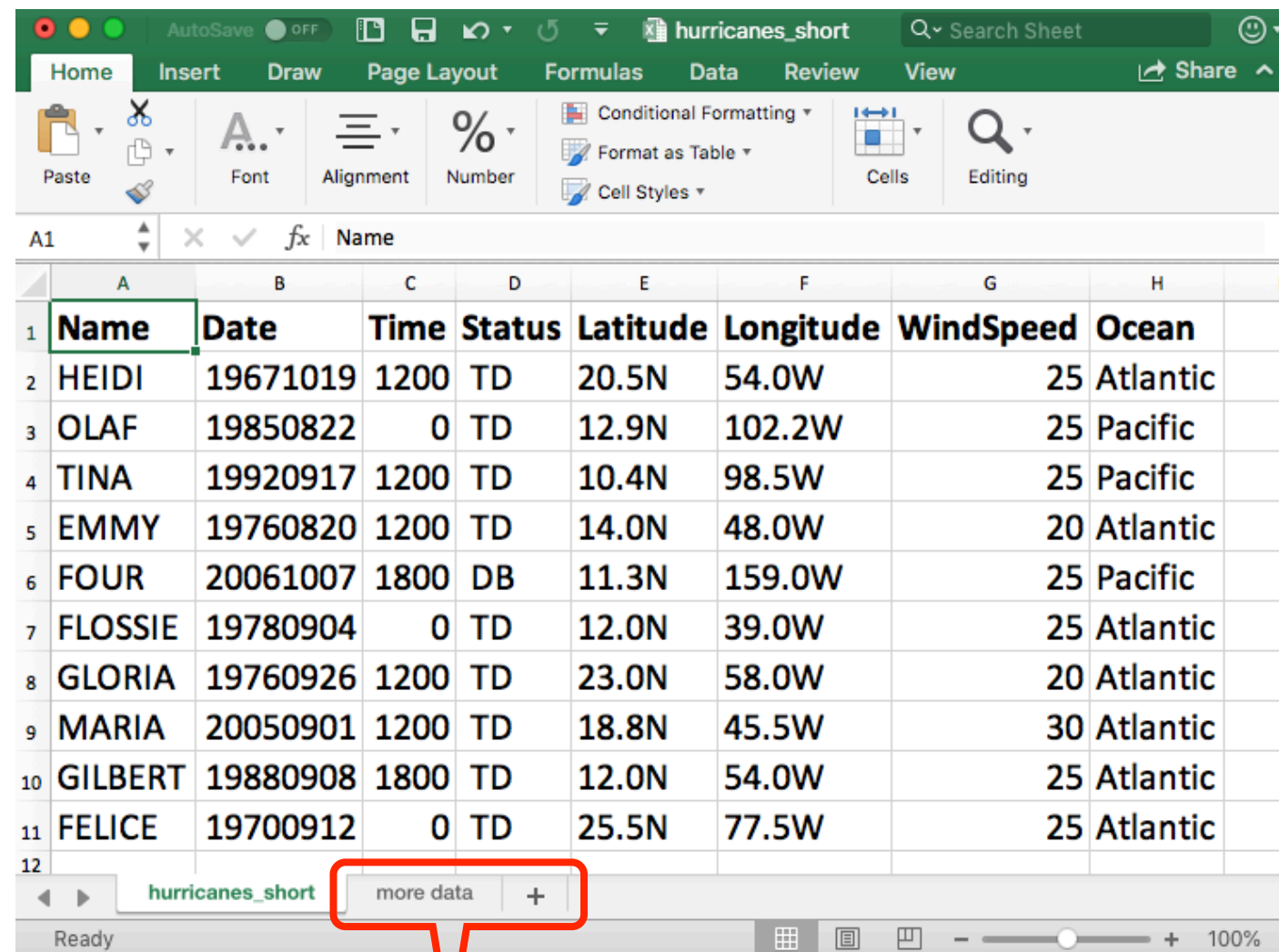
The screenshot shows an Excel spreadsheet with the following data:

Name	Date	Time	Status	Latitude	Longitude	WindSpeed	Ocean
HEIDI	19671019	1200	TD	20.5N	54.0W	25	Atlantic
OLAF	19850822	0	TD	12.9N	102.2W	25	Pacific
TINA	19920917	1200	TD	10.4N	98.5W	25	Pacific
EMMY	19760820	1200	TD	14.0N	48.0W	20	Atlantic
FOUR	20061007	1800	DB	11.3N	159.0W	25	Pacific
FLOSSIE	19780904	0	TD	12.0N	39.0W	25	Atlantic
GLORIA	19760926	1200	TD	23.0N	58.0W	20	Atlantic
MARIA	20050901	1200	TD	18.8N	45.5W	30	Atlantic
GILBERT	19880908	1800	TD	12.0N	54.0W	25	Atlantic
FELICE	19700912	0	TD	25.5N	77.5W	25	Atlantic

Red callouts indicate that the 'Name' header is bolded and the 'EMMY' cell is in regular font.

Spreadsheets (e.g., Excel)

Spreadsheets often support **multiple sheets**



hurricanes_short

	A	B	C	D	E	F	G	H	I
1	Name	Date	Time	Status	Latitude	Longitude	WindSpeed	Ocean	
2	HEIDI	19671019	1200	TD	20.5N	54.0W	25	Atlantic	
3	OLAF	19850822	0	TD	12.9N	102.2W	25	Pacific	
4	TINA	19920917	1200	TD	10.4N	98.5W	25	Pacific	
5	EMMY	19760820	1200	TD	14.0N	48.0W	20	Atlantic	
6	FOUR	20061007	1800	DB	11.3N	159.0W	25	Pacific	
7	FLOSSIE	19780904	0	TD	12.0N	39.0W	25	Atlantic	
8	GLORIA	19760926	1200	TD	23.0N	58.0W	20	Atlantic	
9	MARIA	20050901	1200	TD	18.8N	45.5W	30	Atlantic	
10	GILBERT	19880908	1800	TD	12.0N	54.0W	25	Atlantic	
11	FELICE	19700912	0	TD	25.5N	77.5W	25	Atlantic	
12									

hurricanes_short more data +

more tables of data

Excel Files

Extension: .xlsx

Format: binary

Excel Files

Extension: .xlsx

Format: **binary** just bits in general, not bits representing letters that are easy to read. Need special software...

```
lec-15 — -bash — 67x24
ty-mac:lec-15$ cat hurricanes.xlsx
P!b?h^[Content_Types].xml ?(????N?0E?H?C?-J5??*Q>?ē[c[?i????B?j???
?{2??h?nm????R

????U^/???%??rZY?1__?f??q??R4D?AJ?h>????V?ε

????????NV
?8h?????ji){^??-I?"{?v^?P!XS)bR?r??K?s(??3?`c?0?????????7M4?????ZEk+?|
\|z?(???P??6h_-[?@?!???Pk????2n?}????L??? ??%????????dN"m,?ÄD097*?~??φ
8?0?c|n???E??????B??!$}?????;{???[????2????P!?U0#?L

_rels/.rels ?(???M0?0
??9L?3?sbg_|?l!??USH9i?b?r:"y_dl??D??|-N??R"4?2?G??%??Z?4?"y?? ë??
? ?????P!>???xl/_rels/workbook.xml.rels ?(??RMK?0?T~?I????$?T?G?~??
??<????!??4??;#?w????qu*&r?Fq???v?????GJy(v??*????K??#F??D??W
?=??Z?MY?b???BS?????????ç? ??

????w?v?t/"?UN)?&!

3~??]X?K/o?y???v?5????+??zl?;o??b???G????

?s?>??,?8??(%???"D??4j?0u2j
s??MY?^??S葵 ??? ?)f???C????y?? Iy????!+??E??fMy?k???
??K?5=|?t ??G)?s墙 ?U??tB??)???,???f????????P!u???
```

Excel Files

Extension: .xlsx

Format: **binary** just bits in general, not bits representing letters that are easy to read. Need special software...

```

lec-15 — -bash — 67x24
ty-mac:lec-15$ cat hurricanes.xlsx
P!b?h^[Content_Types].xml ?(????N?0E?H?C?-J5??*Q>?ē[c[?i?i????B?j7??
?{2??h?nm????2R

????U^/???%??rZY?1__?f??q??R4D?AJ?h>????V?ε

????????NV
?8h?????ji){^??-I?"{?v^?P!XS)bR?r??K?s(??`c?0?????????7M4??????ZEk+?|
\|z?(???P??6h_-[?@?!???Pk????2n?}????L??? ??%????????dN"m,?ÅD097*?~???ϕ
8?0?c|n???E??????B??!$}?????;{???[????2????P!U0#?L

    _rels/.rels ?(???M0?0
??9L?3?sbg␣|?l!??USH9i?b?r:"y_dl??D??|-N??R"4?2?G?%??Z?4?"y??  ë??
? ?????P!>???xl/_rels/workbook.xml.rels ?(??RMK?0?T~?I????$?T?G?~??
??<????!??4??;#?w????qu*&r?Fq???v?????GJy(v??*?????K??#F??D??W    ?
?=??Z?MY?b???BS?????????ç? ??

                                ??0w?v?t/"?UN)?&!

3~??]X?K/o?y???v?5????+??zl?;o??b???G????

                                ?s?>??,?8??(%???D??4j?0u2j
s??MY?~???S葵 ??? ?)f???C????y??  Iy!!!!+??E??fMy?k???
??K?5=|?t ??G)?s墙 ?U??tB??)???,???f????????P!u???
```

Writing code to read data from Excel files is tricky, unless you use special modules

Today's Outline

Spreadsheets

CSVs

Reading a CSV to a list of lists

Coding examples

CSVs

CSV is a simple data format that stands for **C**omma-**S**eparated **V**alues

CSVs are like simple spreadsheets

- organize cells of data into rows and columns
- only one sheet per file
- only holds strings
- no way to specify font, borders, cell size, etc

CSV Files

Extension: .csv

Format: **plain text** → just open in any editor (notepad, textedit, PyCharm, etc) and you'll be able to read it

```
lec-15 — -bash — 52x17
ty-mac:lec-15$ cat hurricanes_short.csv
Name,Date,Time,Status,Latitude,Longitude,WindSpeed,0
cean
HEIDI,19671019,1200, TD,20.5N,54.0W,25,Atlantic
OLAF,19850822,0, TD,12.9N,102.2W,25,Pacific
TINA,19920917,1200, TD,10.4N,98.5W,25,Pacific
EMMY,19760820,1200, TD,14.0N,48.0W,20,Atlantic
FOUR,20061007,1800, DB,11.3N,159.0W,25,Pacific
FLOSSIE,19780904,0, TD,12.0N,39.0W,25,Atlantic
GLORIA,19760926,1200, TD,23.0N,58.0W,20,Atlantic
MARIA,20050901,1200, TD,18.8N,45.5W,30,Atlantic
GILBERT,19880908,1800, TD,12.0N,54.0W,25,Atlantic
FELICE,19700912,0, TD,25.5N,77.5W,25,Atlantic
ty-mac:
lec-15$
```

Writing code that understands CSV files is easy

Basic Syntax

Table

Name	Date	Time	Status	Latitude	Longitude	WindSpeed	Ocean
HEIDI	19671019	1200	TD	20.5N	54.0W	25	Atlantic
OLAF	19850822	0	TD	12.9N	102.2W	25	Pacific
TINA	19920917	1200	TD	10.4N	98.5W	25	Pacific
EMMY	19760820	1200	TD	14.0N	48.0W	20	Atlantic

Corresponding CSV

Name,Date,Time,Status,Latitude,Longitude,WindSpeed,Ocean
HEIDI,19671019,1200,TD,20.5N,54.0W,25,Atlantic
OLAF,19850822,0,TD,12.9N,102.2W,25,Pacific
TINA,19920917,1200,TD,10.4N,98.5W,25,Pacific
EMMY,19760820,1200,TD,14.0N,48.0W,20,Atlantic

Basic Syntax

Table

Name	Date	Time	Status	Latitude	Longitude	WindSpeed	Ocean
HEIDI	19671019	1200	TD	20.5N	54.0W	25	Atlantic
OLAF	19850822	0	TD	12.9N	102.2W	25	Pacific
TINA	19920917	1200	TD	10.4N	98.5W	25	Pacific
EMMY	19760820	1200	TD	14.0N	48.0W	20	Atlantic

Corresponding CSV

```
Name,Date,Time,Status,Longitude,Latitude,WindSpeed,Ocean  
HEIDI,19671019,1200,TD,20.5N,54.0W,25,Atlantic  
OLAF,19850822,0,TD,12.9N,102.2W,25,Pacific  
TINA,19920917,1200,TD,10.4N,98.5W,25,Pacific  
EMMY,19760820,1200,TD,14.0N,48.0W,20,Atlantic
```

Each row is a line of the file

Basic Syntax

Table

Name	Date	Time	Status	Latitude	Longitude	WindSpeed	Ocean
HEIDI	19671019	1200	TD	20.5N	54.0W	25	Atlantic
OLAF	19850822	0	TD	12.9N	102.2W	25	Pacific
TINA	19920917	1200	TD	10.4N	98.5W	25	Pacific
EMMY	19760820	1200	TD	14.0N	48.0W	20	Atlantic

Corresponding CSV

Name,Date,Time,Status,Longitude,Latitude,WindSpeed,Ocean

HEIDI,19671019,1200,TD,20.5N,54.0W,25,Atlantic

OLAF,19850822,0,TD,12.9N,102.2W,25,Pacific

TINA,19920917,1200,TD,10.4N,98.5W,25,Pacific

EMMY,19760820,1200,TD,14.0N,48.0W,20,Atlantic

Each row is a line of the file

Basic Syntax

Table

Name	Date	Time	Status	Latitude	Longitude	WindSpeed	Ocean
HEIDI	19671019	1200	TD	20.5N	54.0W	25	Atlantic
OLAF	19850822	0	TD	12.9N	102.2W	25	Pacific
TINA	19920917	1200	TD	10.4N	98.5W	25	Pacific
EMMY	19760820	1200	TD	14.0N	48.0W	20	Atlantic

Corresponding CSV

Name,Date,Time,Status,Longitude,Latitude,WindSpeed,Ocean

HEIDI,19671019,1200,TD,20.5N,54.0W,25,Atlantic

OLAF,19850822,0,TD,12.9N,102.2W,25,Pacific


TINA,19920917,1200,TD,10.4N,98.5W,25,Pacific

EMMY,19760820,1200,TD,14.0N,48.0W,20,Atlantic

Cells...

Basic Syntax

Table



Name	Date	Time	Status	Latitude	Longitude	WindSpeed	Ocean
HEIDI	19671019	1200	TD	20.5N	54.0W	25	Atlantic
OLAF	19850822	0	TD	12.9N	102.2W	25	Pacific
TINA	19920917	1200	TD	10.4N	98.5W	25	Pacific
EMMY	19760820	1200	TD	14.0N	48.0W	20	Atlantic

Corresponding CSV

Name,Date,Time,Status,Longitude,Latitude,WindSpeed,Ocean
HEIDI,19671019,1200,TD,20.5N,54.0W,25,Atlantic
OLAF,19850822,0,TD,12.9N,102.2W,25,Pacific
TINA,19920917,1200,TD,10.4N,98.5W,25,Pacific
EMMY,19760820,1200,TD,14.0N,48.0W,20,Atlantic

... are separated by commas

Basic Syntax

Table

Name	Date	Time	Status	Latitude	Longitude	WindSpeed	Ocean
HEIDI	19671019	1200	TD	20.5N	54.0W	25	Atlantic
OLAF	19850822	0	TD	12.9N	102.2W	25	Pacific
TINA	19920917	1200	TD	10.4N	98.5W	25	Pacific
EMMY	19760820	1200	TD	14.0N	48.0W	20	Atlantic

We call characters that act as separators “**delimiters**”

Name,Date,Time,Status,Latitude,Longitude,WindSpeed,Ocean

HEIDI,19671019,1200,TD,20.5N,54.0W,25,Atlantic

OLAF,19850822,0,TD,12.9N,102.2W,25,Pacific

TINA,19920917,1200,TD,10.4N,98.5W,25,Pacific

EMMY,19760820,1200,TD,14.0N,48.0W,20,Atlantic

... are separated by commas

Advanced Syntax

We won't go into details here, but there are some complexities

Motivation for more complicated syntax

- what if a cell contains a newline?
- what if we want a comma inside a cell?
- what if a cell contains a quote?
- what if we want to use different delimiters between rows/cells?

Today's Outline

Spreadsheets

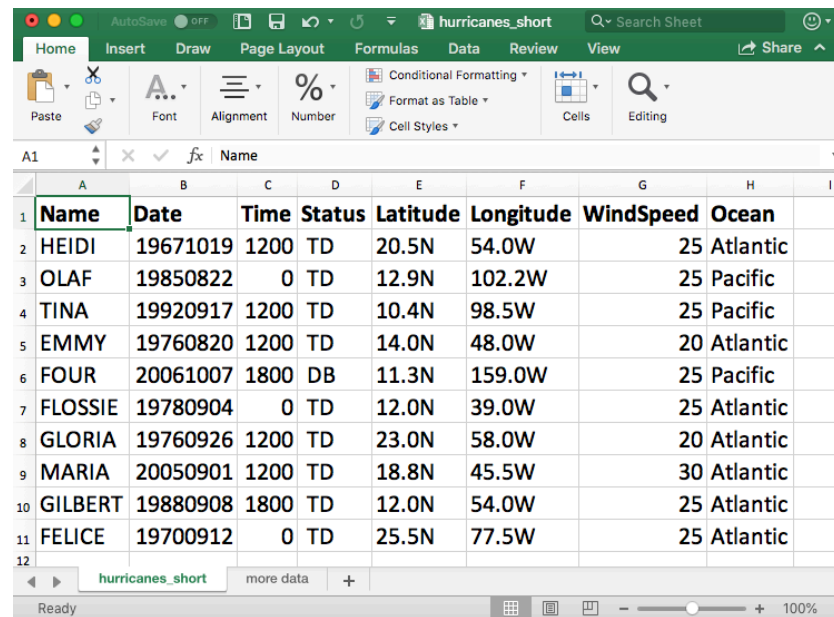
CSVs

Reading a CSV to a list of lists

Coding examples

Data Management

1. spreadsheet in Excel



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I
1	Name	Date	Time	Status	Latitude	Longitude	WindSpeed	Ocean	
2	HEIDI	19671019	1200	TD	20.5N	54.0W	25	Atlantic	
3	OLAF	19850822	0	TD	12.9N	102.2W	25	Pacific	
4	TINA	19920917	1200	TD	10.4N	98.5W	25	Pacific	
5	EMMY	19760820	1200	TD	14.0N	48.0W	20	Atlantic	
6	FOUR	20061007	1800	DB	11.3N	159.0W	25	Pacific	
7	FLOSSIE	19780904	0	TD	12.0N	39.0W	25	Atlantic	
8	GLORIA	19760926	1200	TD	23.0N	58.0W	20	Atlantic	
9	MARIA	20050901	1200	TD	18.8N	45.5W	30	Atlantic	
10	GILBERT	19880908	1800	TD	12.0N	54.0W	25	Atlantic	
11	FELICE	19700912	0	TD	25.5N	77.5W	25	Atlantic	
12									



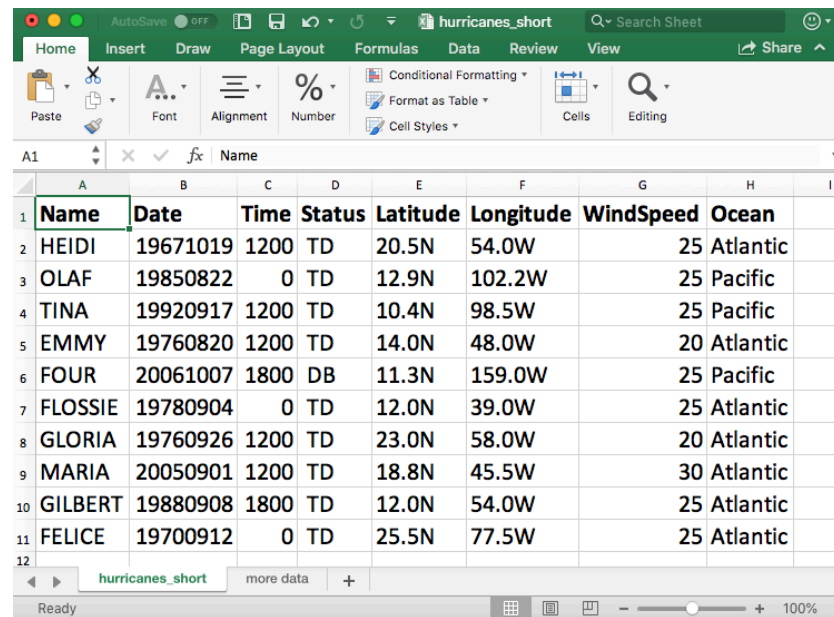
2. CSV file saved somewhere

```
Name,Date,Time,Status,Latitude,Longitude,WindSpeed,Ocean
HEIDI,19671019,1200, TD,20.5N,54.0W,25,Atlantic
OLAF,19850822,0, TD,12.9N,102.2W,25,Pacific
TINA,19920917,1200, TD,10.4N,98.5W,25,Pacific
EMMY,19760820,1200, TD,14.0N,48.0W,20,Atlantic
```

Data Management

3. Python Program

1. spreadsheet in Excel



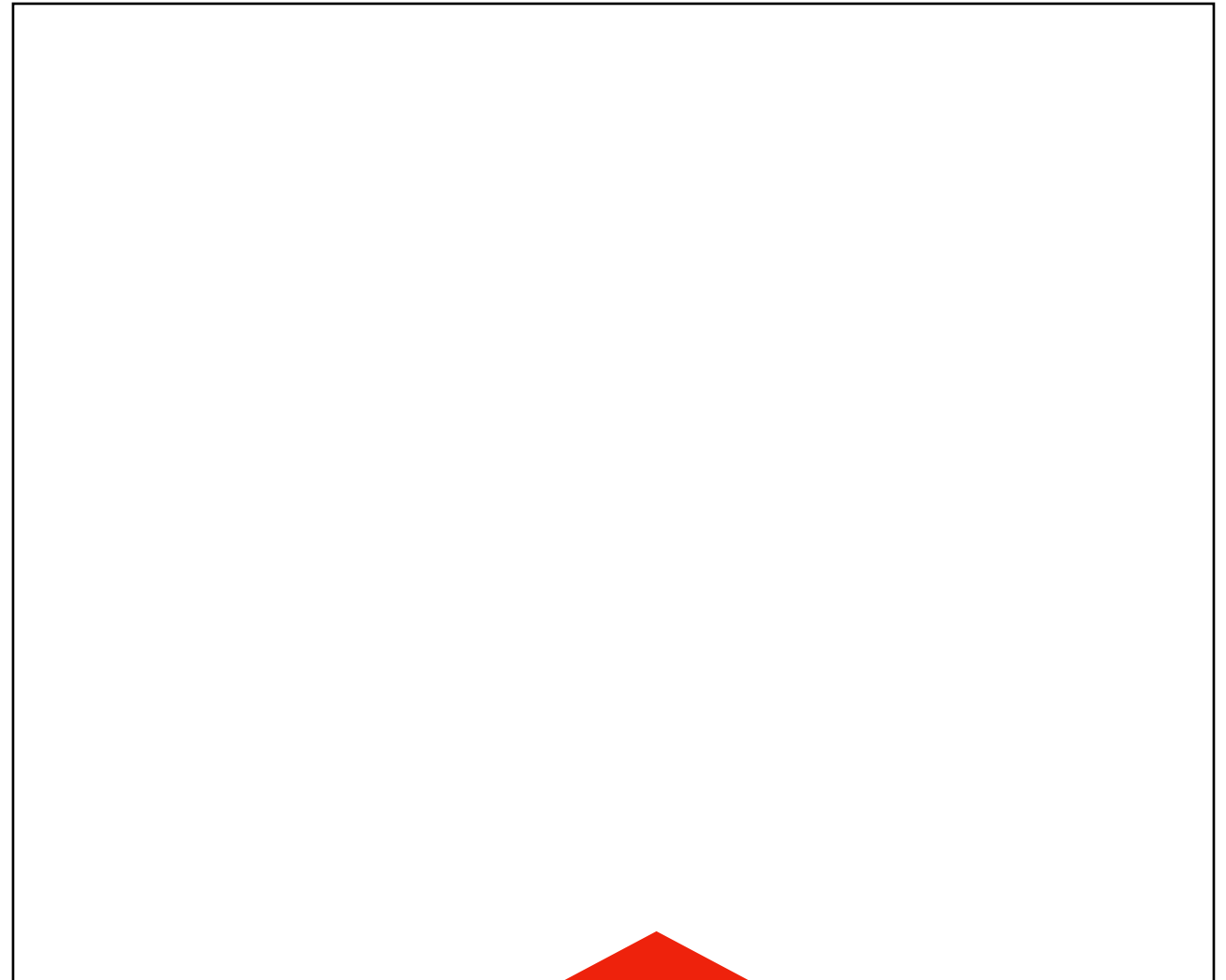
The screenshot shows an Excel spreadsheet with the following data:

Name	Date	Time	Status	Latitude	Longitude	WindSpeed	Ocean
HEIDI	19671019	1200	TD	20.5N	54.0W	25	Atlantic
OLAF	19850822	0	TD	12.9N	102.2W	25	Pacific
TINA	19920917	1200	TD	10.4N	98.5W	25	Pacific
EMMY	19760820	1200	TD	14.0N	48.0W	20	Atlantic
FOUR	20061007	1800	DB	11.3N	159.0W	25	Pacific
FLOSSIE	19780904	0	TD	12.0N	39.0W	25	Atlantic
GLORIA	19760926	1200	TD	23.0N	58.0W	20	Atlantic
MARIA	20050901	1200	TD	18.8N	45.5W	30	Atlantic
GILBERT	19880908	1800	TD	12.0N	54.0W	25	Atlantic
FELICE	19700912	0	TD	25.5N	77.5W	25	Atlantic

**Save As
.CSV**

2. CSV file saved somewhere

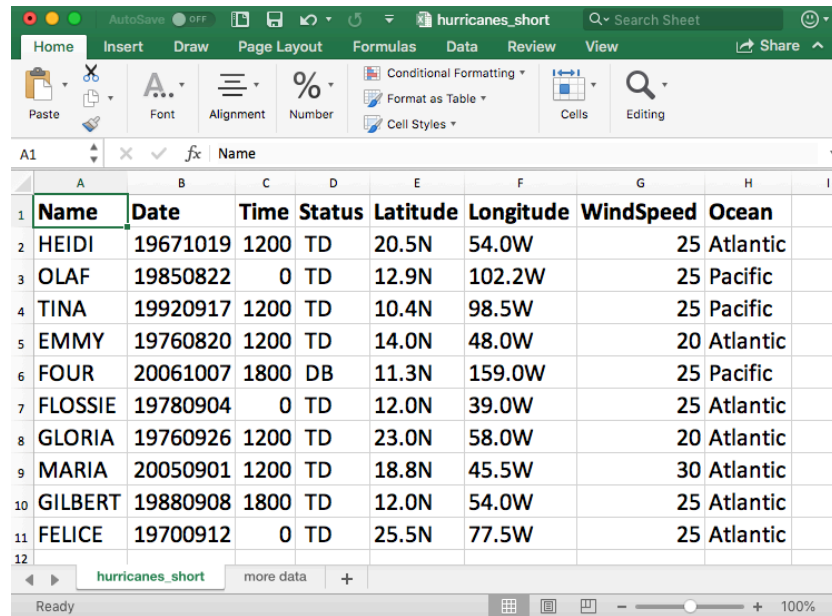
```
Name,Date,Time,Status,Latitude,Longitude,WindSpeed,Ocean  
HEIDI,19671019,1200, TD,20.5N,54.0W,25,Atlantic  
OLAF,19850822,0, TD,12.9N,102.2W,25,Pacific  
TINA,19920917,1200, TD,10.4N,98.5W,25,Pacific  
EMMY,19760820,1200, TD,14.0N,48.0W,20,Atlantic
```



Data Management

3. Python Program

1. spreadsheet in Excel



The screenshot shows an Excel spreadsheet with the following data:

Name	Date	Time	Status	Latitude	Longitude	WindSpeed	Ocean
HEIDI	19671019	1200	TD	20.5N	54.0W	25	Atlantic
OLAF	19850822	0	TD	12.9N	102.2W	25	Pacific
TINA	19920917	1200	TD	10.4N	98.5W	25	Pacific
EMMY	19760820	1200	TD	14.0N	48.0W	20	Atlantic
FOUR	20061007	1800	DB	11.3N	159.0W	25	Pacific
FLOSSIE	19780904	0	TD	12.0N	39.0W	25	Atlantic
GLORIA	19760926	1200	TD	23.0N	58.0W	20	Atlantic
MARIA	20050901	1200	TD	18.8N	45.5W	30	Atlantic
GILBERT	19880908	1800	TD	12.0N	54.0W	25	Atlantic
FELICE	19700912	0	TD	25.5N	77.5W	25	Atlantic

Save As
.CSV

2. CSV file saved somewhere

```
Name,Date,Time,Status,Latitude,Longitude,WindSpeed,Ocean
HEIDI,19671019,1200, TD,20.5N,54.0W,25,Atlantic
OLAF,19850822,0, TD,12.9N,102.2W,25,Pacific
TINA,19920917,1200, TD,10.4N,98.5W,25,Pacific
EMMY,19760820,1200, TD,14.0N,48.0W,20,Atlantic
```

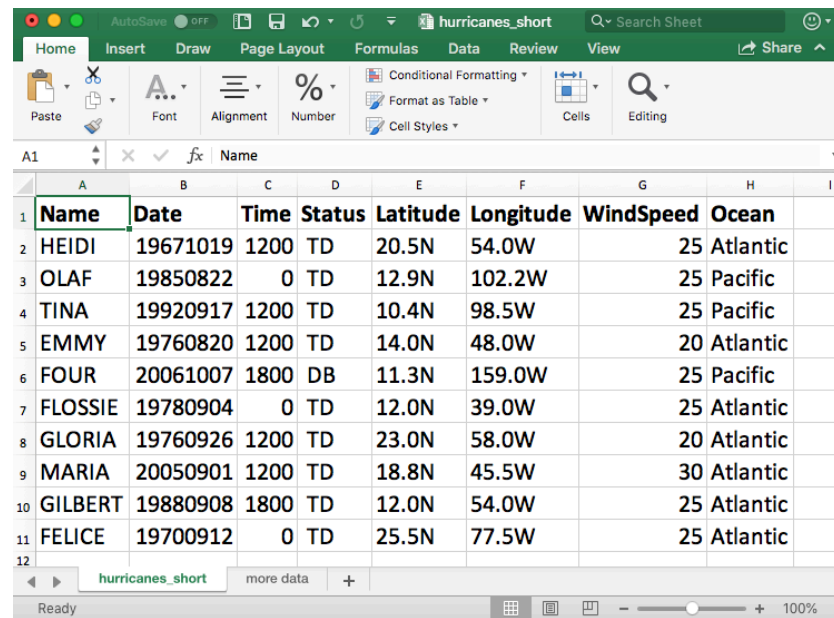
```
[
  ["Name", "Date", ...],
  ["HEIDI", "19671019", ...],
  ...
]
```

Parsing Code

Data Management

3. Python Program

1. spreadsheet in Excel



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I
1	Name	Date	Time	Status	Latitude	Longitude	WindSpeed	Ocean	
2	HEIDI	19671019	1200	TD	20.5N	54.0W	25	Atlantic	
3	OLAF	19850822	0	TD	12.9N	102.2W	25	Pacific	
4	TINA	19920917	1200	TD	10.4N	98.5W	25	Pacific	
5	EMMY	19760820	1200	TD	14.0N	48.0W	20	Atlantic	
6	FOUR	20061007	1800	DB	11.3N	159.0W	25	Pacific	
7	FLOSSIE	19780904	0	TD	12.0N	39.0W	25	Atlantic	
8	GLORIA	19760926	1200	TD	23.0N	58.0W	20	Atlantic	
9	MARIA	20050901	1200	TD	18.8N	45.5W	30	Atlantic	
10	GILBERT	19880908	1800	TD	12.0N	54.0W	25	Atlantic	
11	FELICE	19700912	0	TD	25.5N	77.5W	25	Atlantic	
12									

Save As
.CSV

2. CSV file saved somewhere

```
Name,Date,Time,Status,Latitude,Longitude,WindSpeed,Ocean
HEIDI,19671019,1200, TD,20.5N,54.0W,25,Atlantic
OLAF,19850822,0, TD,12.9N,102.2W,25,Pacific
TINA,19920917,1200, TD,10.4N,98.5W,25,Pacific
EMMY,19760820,1200, TD,14.0N,48.0W,20,Atlantic
```

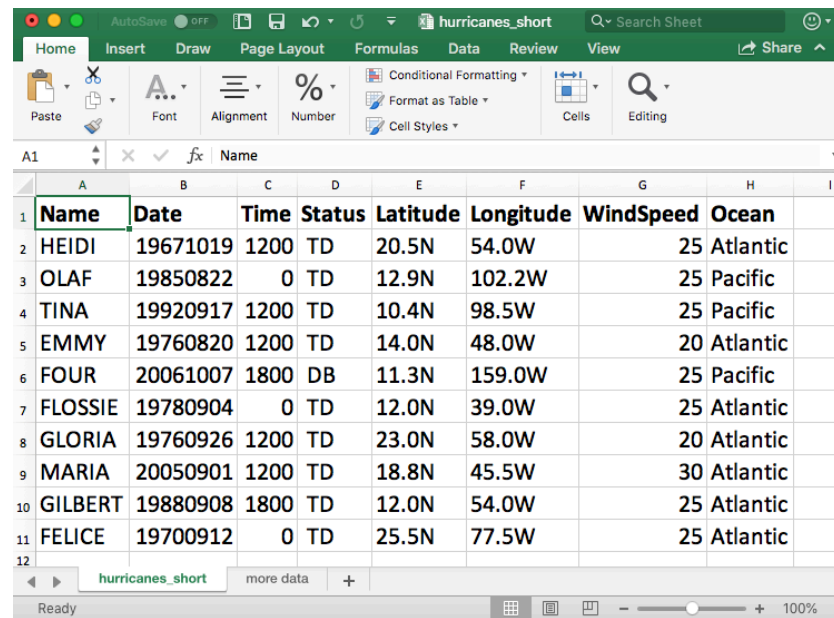
list of lists

```
[
    ["Name", "Date", ...],
    ["HEIDI", "19671019", ...],
    ...
]
```

Parsing Code

Data Management

1. spreadsheet in Excel



The screenshot shows an Excel spreadsheet with the following data:

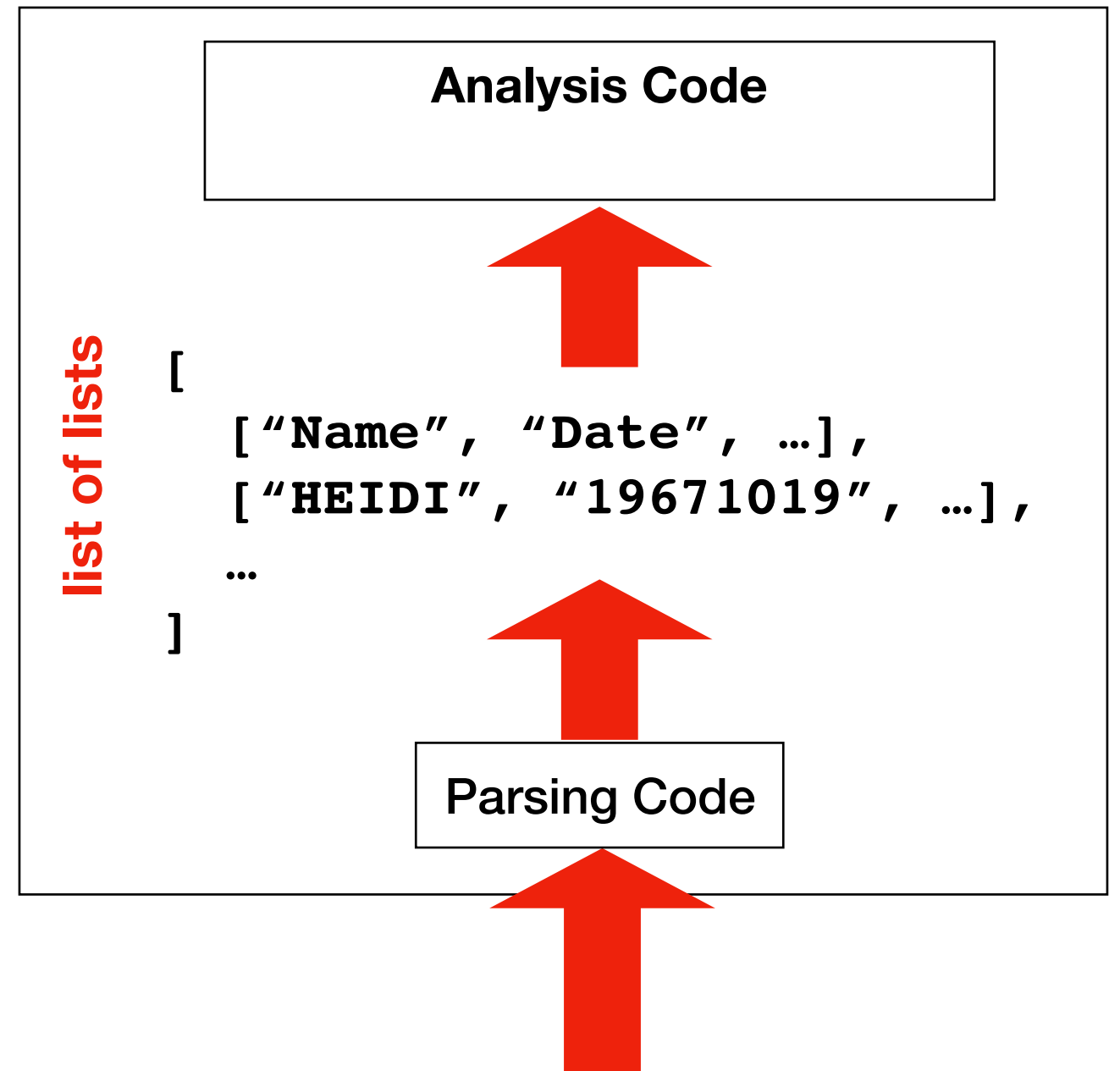
	A	B	C	D	E	F	G	H	I
1	Name	Date	Time	Status	Latitude	Longitude	WindSpeed	Ocean	
2	HEIDI	19671019	1200	TD	20.5N	54.0W	25	Atlantic	
3	OLAF	19850822	0	TD	12.9N	102.2W	25	Pacific	
4	TINA	19920917	1200	TD	10.4N	98.5W	25	Pacific	
5	EMMY	19760820	1200	TD	14.0N	48.0W	20	Atlantic	
6	FOUR	20061007	1800	DB	11.3N	159.0W	25	Pacific	
7	FLOSSIE	19780904	0	TD	12.0N	39.0W	25	Atlantic	
8	GLORIA	19760926	1200	TD	23.0N	58.0W	20	Atlantic	
9	MARIA	20050901	1200	TD	18.8N	45.5W	30	Atlantic	
10	GILBERT	19880908	1800	TD	12.0N	54.0W	25	Atlantic	
11	FELICE	19700912	0	TD	25.5N	77.5W	25	Atlantic	
12									

Save As
.CSV

2. CSV file saved somewhere

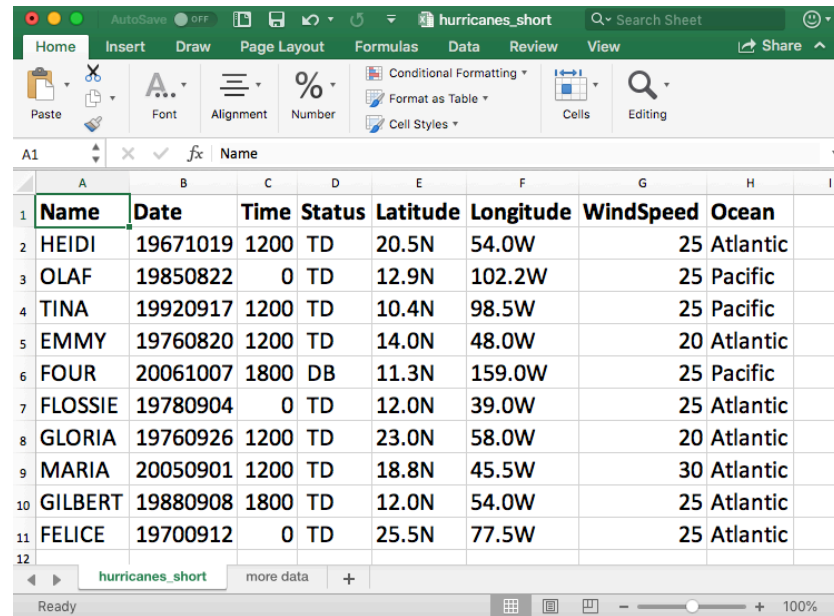
```
Name,Date,Time,Status,Latitude,Longitude,WindSpeed,Ocean
HEIDI,19671019,1200, TD,20.5N,54.0W,25,Atlantic
OLAF,19850822,0, TD,12.9N,102.2W,25,Pacific
TINA,19920917,1200, TD,10.4N,98.5W,25,Pacific
EMMY,19760820,1200, TD,14.0N,48.0W,20,Atlantic
```

3. Python Program



Data Management

1. spreadsheet in Excel



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I
1	Name	Date	Time	Status	Latitude	Longitude	WindSpeed	Ocean	
2	HEIDI	19671019	1200	TD	20.5N	54.0W	25	Atlantic	
3	OLAF	19850822	0	TD	12.9N	102.2W	25	Pacific	
4	TINA	19920917	1200	TD	10.4N	98.5W	25	Pacific	
5	EMMY	19760820	1200	TD	14.0N	48.0W	20	Atlantic	
6	FOUR	20061007	1800	DB	11.3N	159.0W	25	Pacific	
7	FLOSSIE	19780904	0	TD	12.0N	39.0W	25	Atlantic	
8	GLORIA	19760926	1200	TD	23.0N	58.0W	20	Atlantic	
9	MARIA	20050901	1200	TD	18.8N	45.5W	30	Atlantic	
10	GILBERT	19880908	1800	TD	12.0N	54.0W	25	Atlantic	
11	FELICE	19700912	0	TD	25.5N	77.5W	25	Atlantic	
12									

Save As
.CSV

2. CSV file saved somewhere

```
Name,Date,Time,Status,Latitude,Longitude,WindSpeed,Ocean
HEIDI,19671019,1200, TD,20.5N,54.0W,25,Atlantic
OLAF,19850822,0, TD,12.9N,102.2W,25,Pacific
TINA,19920917,1200, TD,10.4N,98.5W,25,Pacific
EMMY,19760820,1200, TD,14.0N,48.0W,20,Atlantic
```

3. Python Program

Analysis Code
`rows[1][0] → "HEIDI"`

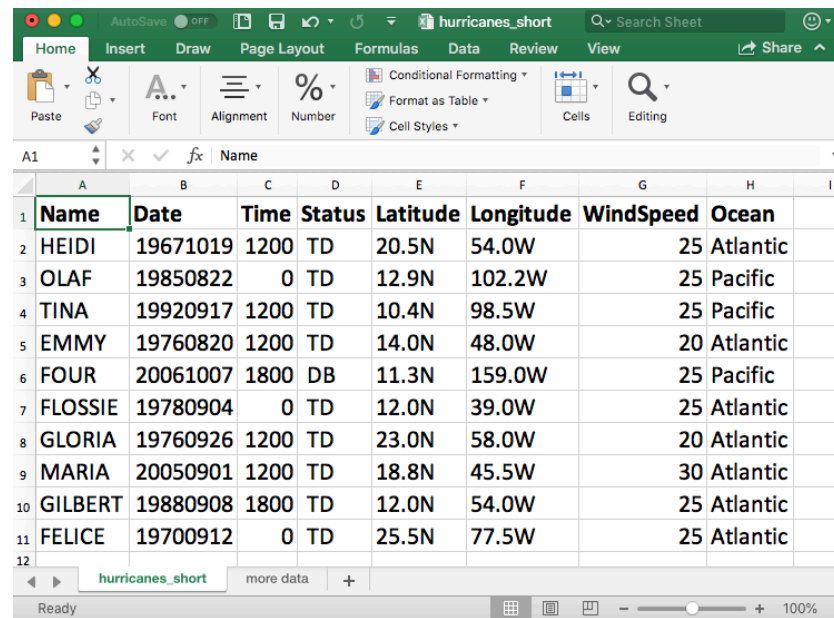
list of lists

```
[  
  ["Name", "Date", ...],  
  ["HEIDI", "19671019", ...],  
  ...  
]
```

Parsing Code

Data Management

1. spreadsheet in Excel



Name	Date	Time	Status	Latitude	Longitude	WindSpeed	Ocean
HEIDI	19671019	1200	TD	20.5N	54.0W	25	Atlantic
OLAF	19850822	0	TD	12.9N	102.2W	25	Pacific
TINA	19920917	1200	TD	10.4N	98.5W	25	Pacific
EMMY	19760820	1200	TD	14.0N	48.0W	20	Atlantic
FOUR	20061007	1800	DB	11.3N	159.0W	25	Pacific
FLOSSIE	19780904	0	TD	12.0N	39.0W	25	Atlantic
GLORIA	19760926	1200	TD	23.0N	58.0W	20	Atlantic
MARIA	20050901	1200	TD	18.8N	45.5W	30	Atlantic
GILBERT	19880908	1800	TD	12.0N	54.0W	25	Atlantic
FELICE	19700912	0	TD	25.5N	77.5W	25	Atlantic

Save As
.CSV

2. CSV file saved somewhere

```
Name,Date,Time,Status,Latitude,Longitude,WindSpeed,Ocean
HEIDI,19671019,1200, TD,20.5N,54.0W,25,Atlantic
OLAF,19850822,0, TD,12.9N,102.2W,25,Pacific
TINA,19920917,1200, TD,10.4N,98.5W,25,Pacific
EMMY,19760820,1200, TD,14.0N,48.0W,20,Atlantic
```

3. Python Program

Analysis Code
`rows[2][-1] → ????`

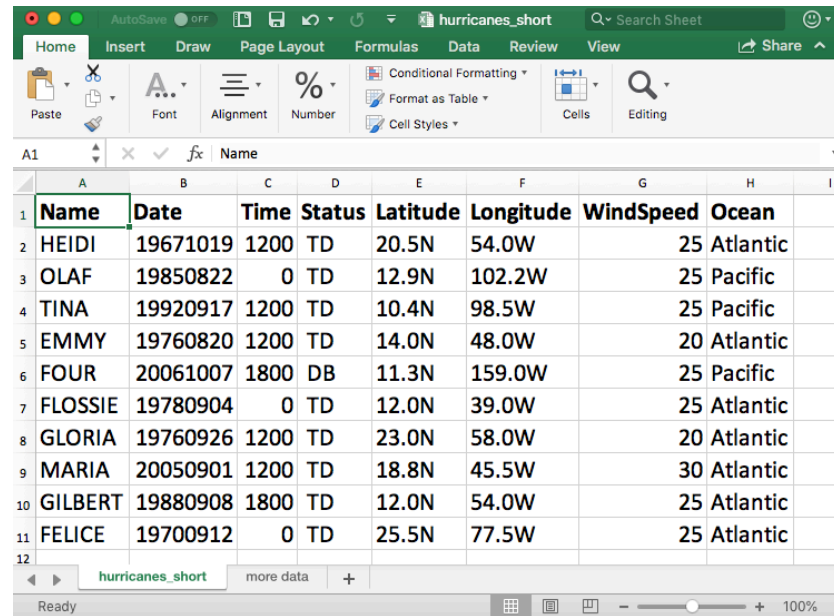
list of lists

```
[
  ["Name", "Date", ...],
  ["HEIDI", "19671019", ...],
  ...
]
```

Parsing Code

Data Management

1. spreadsheet in Excel



The screenshot shows an Excel spreadsheet with the following data:

Name	Date	Time	Status	Latitude	Longitude	WindSpeed	Ocean
HEIDI	19671019	1200	TD	20.5N	54.0W	25	Atlantic
OLAF	19850822	0	TD	12.9N	102.2W	25	Pacific
TINA	19920917	1200	TD	10.4N	98.5W	25	Pacific
EMMY	19760820	1200	TD	14.0N	48.0W	20	Atlantic
FOUR	20061007	1800	DB	11.3N	159.0W	25	Pacific
FLOSSIE	19780904	0	TD	12.0N	39.0W	25	Atlantic
GLORIA	19760926	1200	TD	23.0N	58.0W	20	Atlantic
MARIA	20050901	1200	TD	18.8N	45.5W	30	Atlantic
GILBERT	19880908	1800	TD	12.0N	54.0W	25	Atlantic
FELICE	19700912	0	TD	25.5N	77.5W	25	Atlantic

Save As
.CSV

2. CSV file saved somewhere

```
Name,Date,Time,Status,Latitude,Longitude,WindSpeed,Ocean
HEIDI,19671019,1200, TD,20.5N,54.0W,25,Atlantic
OLAF,19850822,0, TD,12.9N,102.2W,25,Pacific
TINA,19920917,1200, TD,10.4N,98.5W,25,Pacific
EMMY,19760820,1200, TD,14.0N,48.0W,20,Atlantic
```

3. Python Program

Analysis Code
`rows [2] [-1] → "Pacific"`

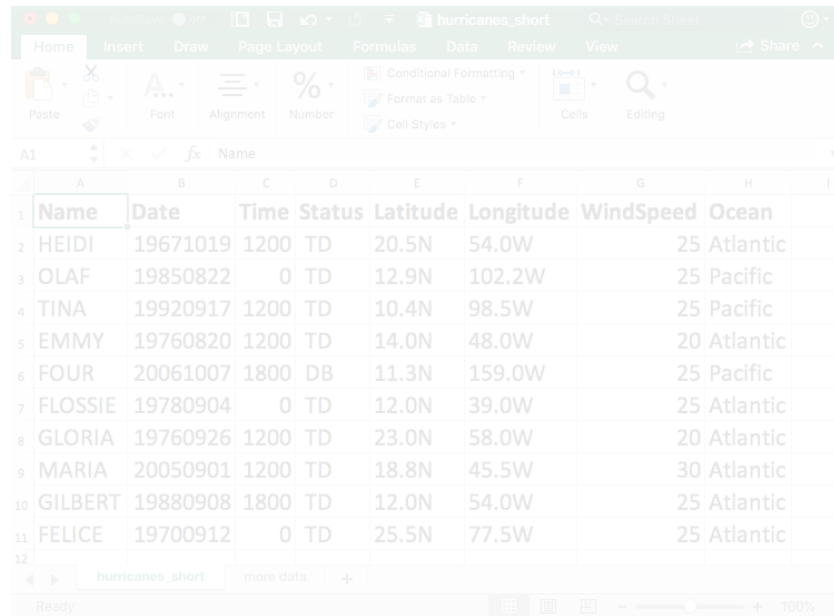
list of lists

```
[  
  [ "Name", "Date", ... ],  
  [ "HEIDI", "19671019", ... ],  
  ...  
]
```

Parsing Code

Data Management

1. spreadsheet in Excel



Name	Date	Time	Status	Latitude	Longitude	WindSpeed	Ocean
HEIDI	19671019	1200	TD	20.5N	54.0W	25	Atlantic
OLAF	19850822	0	TD	12.9N	102.2W	25	Pacific
TINA	19920917	1200	TD	10.4N	98.5W	25	Pacific
EMMY	19760820	1200	TD	14.0N	48.0W	20	Atlantic
FOUR	20061007	1800	DB	11.3N	159.0W	25	Pacific
FLOSSIE	19780904	0	TD	12.0N	39.0W	25	Atlantic
GLORIA	19760926	1200	TD	23.0N	58.0W	20	Atlantic
MARIA	20050901	1200	TD	18.8N	45.5W	30	Atlantic
GILBERT	19880908	1800	TD	12.0N	54.0W	25	Atlantic
FELICE	19700912	0	TD	25.5N	77.5W	25	Atlantic

Save As
.CSV

2. CSV file saved somewhere

```
Name,Date,Time,Status,Latitude,Longitude,WindSpeed,Ocean
HEIDI,19671019,1200,TD,20.5N,54.0W,25,Atlantic
OLAF,19850822,0,TD,12.9N,102.2W,25,Pacific
TINA,19920917,1200,TD,10.4N,98.5W,25,Pacific
EMMY,19760820,1200,TD,14.0N,48.0W,20,Atlantic
```

3. Python Program

Analysis Code
`rows[2][-1] → "Pacific"`

list of lists

```
[
  ["Name", "Date", ...],
  ["HEIDI", "19671019", ...],
  ...
]
```

Parsing Code

What does this look like?

Example From Sweigart Ch 14

Code

```
import csv
exampleFile = open( 'example.csv' )
exampleReader = csv.reader(exampleFile)
exampleData = list(exampleReader)
```

Example From Sweigart Ch 14

Code

```
import csv
exampleFile = open('example.csv')
exampleReader = csv.reader(exampleFile)
exampleData = list(exampleReader)
```

`exampleData`



list of
lists

```
[[['4/5/2015 13:34', 'Apples', '73'], ['4/5/2015 3:41', 'Cherries', '85'],  
  ['4/6/2015 12:46', 'Pears', '14'], ['4/8/2015 8:59', 'Oranges', '52'],  
  ['4/10/2015 2:07', 'Apples', '152'], ['4/10/2015 18:10', 'Bananas', '23'],  
  ['4/10/2015 2:40', 'Strawberries', '98']]]
```

Example From Sweigart Ch 14

```
import csv
exampleFile = open( 'example.csv' )
exampleReader = csv.reader(exampleFile)
exampleData = list(exampleReader)
exampleData
```

**We'll eventually learn more about reading files.
For now, let's copy and paste this to a function
so we don't need to worry about it.**

Example From Sweigart Ch 14

```
def process_csv():  
    import csv  
    exampleFile = open('example.csv')  
    exampleReader = csv.reader(exampleFile)  
    exampleData = list(exampleReader)  
    exampleData
```

**We'll eventually learn more about reading files.
For now, let's copy and paste this to a function
so we don't need to worry about it.**

Example From Sweigart Ch 14

```
import csv

def process_csv():
    import csv
    exampleFile = open('example.csv')
    exampleReader = csv.reader(exampleFile)
    exampleData = list(exampleReader)
    exampleData
```

**We'll eventually learn more about reading files.
For now, let's copy and paste this to a function
so we don't need to worry about it.**

Example From Sweigart Ch 14

```
import csv

def process_csv():
    import csv
    exampleFile = open('example.csv')
    exampleReader = csv.reader(exampleFile)
    exampleData = list(exampleReader)
    return exampleData
```

**We'll eventually learn more about reading files.
For now, let's copy and paste this to a function
so we don't need to worry about it.**

Example From Sweigart Ch 14

```
import csv

def process_csv():
    import csv
    exampleFile = open('example.csv')
    exampleReader = csv.reader(exampleFile)
    exampleData = list(exampleReader)
    return exampleData
```

**We'll eventually learn more about reading files.
For now, let's copy and paste this to a function
so we don't need to worry about it.**

Example From Sweigart Ch 14

```
import csv

def process_csv(filename):
    import csv
    exampleFile = open(filename)
    exampleReader = csv.reader(exampleFile)
    exampleData = list(exampleReader)
    return exampleData
```

**We'll eventually learn more about reading files.
For now, let's copy and paste this to a function
so we don't need to worry about it.**

Example From Sweigart Ch 14

```
import csv
```

```
# copied from https://automatetheboringstuff.com/chapter14/  
def process_csv(filename):  
    import csv  
    exampleFile = open(filename)  
    exampleReader = csv.reader(exampleFile)  
    exampleData = list(exampleReader)  
    return exampleData
```

Reminder!
cite code
copied online

**We'll eventually learn more about reading files.
For now, let's copy and paste this to a function
so we don't need to worry about it.**

Today's Outline

Spreadsheets

CSVs

Reading a CSV to a list of lists

Coding examples

Demo 1: Restaurant Location Lookup

Goal: given a restaurant name, give x,y coordinates for it

Input:

- Restaurant name (and a CSV file)

Output:

- X, Y coordinates

Example:

```
prompt> python rlookup.py subway
```

```
x=1, y=0
```

```
prompt> python rlookup.py mcdonalds
```

```
x=4, y=-3
```

Demo 2: Nearest Restaurant Search

Goal: given a location, find the nearest restaurant

Input:

- X, Y coordinates (and a CSV file)

Output:

- nearest restaurant

Example:

```
prompt> python nearest.py 4,-4
```

```
McDonalds
```

```
prompt> python nearest.py -2,0
```

```
The Sett
```

Demo 3: Hurricane Column Dump

Goal: column name, print that data for all hurricanes

Input:

- column name (and a CSV file)

Output:

- data in given column, associated with name

Example:

```
prompt> python dump.py ocean
```

```
HEIDI: Atlantic
```

```
OLAF: Pacific
```

```
TINA: Pacific
```

```
...
```


Demo 4: Hurricanes per Year

Goal: column name, print that data for all hurricanes

Input:

- none typed (only a CSV file)

Output:

- the number of hurricanes in each year

Example:

```
prompt> python yearly.py ocean
```

```
1967: 23
```

```
1968: 29
```

```
2969: 15
```

```
...
```