Python in 1 Hour (Part 1)

David Quigley
University of Colorado Boulder

What you've done so far...

Gotten a Python environment up and running
 - Including the NumPy and SciKitLearn Packages

I recommend the Anaconda distribution:
https://www.anaconda.com/products/individual

Hello World

What Just Happened?

```
print(“hello, world!”)
```

- Python found a set of double quotes and took everything between those double quotes as a *string*.
- Python found a *function*, **print**, and tried to execute that command on anything inside the parentheses.
    - The **print** function can work well with strings!

What is a string?

- A *string* is an ordered collection (a.k.a. string) of *characters*

What is a character?

- A single letter, symbol, etc.
    - In this class, we won't be using them much.

Can we do anything besides put Strings in print functions?

Variables? But not Math?

```
X = “hello 2.0”
print(X)
```

- Now I put my string into a variable, then I asked my print function to use the contents of that variable.

Can computers do math?

Variables? But Math?

```
X = “3*4”
Y = 3*4
print(X)
print(Y)
```

- Here, I’ve made a string variable “3*4”, and a numeric variable that is the result of the multiplication.
    - Python stores the results as a numeric value in the latter case
- Print can work with a numeric value!
    - Where did I learn that * will do multiplication???

Python in 1 Hour (Part 2)

David Quigley

CSCI 2830

Summer 2020

INTERLUDE: The power of Searching

We did multiplication with *.

We can probably do +, -, /, but can we do more?

What about taking the square root? Logarithms? Other mathematical functions?

Searching

How-To Guides
- W3 Schools (originally for Web Programming)

Python Documentation
- docs.python.org

Community Helpdesk
- StackOverflow

Can Computers do better math?

Math Functions

```
import math
print(1+2*3-4/5)
print(math.sqrt(2))
print(math.log(10))
```

- Here, we have two new functions, *sq(uare)r(oo)t* and *log(arithm)*
    - Notice that they come from the math *Library*
- We get new precision with numbers – floating point numbers!
    - Notice that *log* doesn't use base 10, it actually uses base *e* for calculations.

What is a Library?

- A library is a collection of functions, variables, and other information

When Things Go Wrong

When Things Go Wrong

Messing up capitalization?

```
Print(1+2*3-4/5)
```

Messing up parentheses?

```
print(sqrt(2)
```

Forgetting about `import math`?

```
print(math.log(10))
```

Variable name error?

```
abc = 10
print(abd)
```

Python in 1 Hour (Part 3)

David Quigley

CSCI 2830

Summer 2020

Reading a Data File

Reading a Data File

```
from pandas import read_csv
records = read_csv('testScores.csv')
print(records.head())
```

- Opened the file itself
    - It had to be in the same folder (directory), otherwise we would have to indicate location as well!
- printed the "records" variable (or, the beginning of it) that is created

Does that do anything particularly useful?

Pandas DataFrame

A storage space (frame) for information (data)
- The CSV keeps things in rows and columns

What's in Data?

Columns : Features
- Each different facet of information you've taken in your data sampling

Rows : Samples
- Each different item / individual / time from which you have sampled data

Conditional Statements (if-else statements)

Let's write a grading program

- X >= 90 is an A
- 80 <= X < 90 is a B
- 70 <= X < 80 is a C
- 60 <= X < 70 is a D
- Anything else is F

Conditionals

```
if data[1][student_number] > 90:
      print(data[0][student_number] + “ got an A!”)
elif data[1][student_number] > 80:
      print(data[0][student_number] + “ got a B!”)
elif data[1][student_number] > 70:
      print(data[0][student_number] + “ got a C!”)
elif data[1][student_number] > 60:
      print(data[0][student_number] + “ got a D!”)
else:
      print(data[0][student_number] + “ got an F!”)
```

Extracting Data from a DataFrame

- How did “`franklin`” do?

Extracting Data from a DataFrame

- How did “`franklin`” do?
- From the entire DataFrame, find the row with franklin’s information
    - From the row with franklin’s information, extract their score

Getting Franklin's Score

```
franklin = records.loc[records.username == “franklin”]
```

- We've (loc)ated *any and all* rows in the records DataFrame where the `username` column's value is equal to “`franklin`”

What did that get us?

- Still a DataFrame
    - Excluded extraneous data

Getting Franklin's Score

```
franklin = records.loc[records.username == “franklin”]
studentScore = franklin.iat[0,2]
```

- We've extracted the value that is (i)ndexed (at) the indeces row (0), column (2)

What did that get us?

- A scalar value
    - A numeric value (89, in this case)
    - Franklin's score!

Extracting Franklin's Score in One Step

```
studentScore = records.loc[records.username == "franklin"]["score"].values[0]
```

- We've done a series of steps:
    - located *any and all entries* in records where `username == "franklin"`
    - isolated the "score" column from our subframe of records
    - converted the DataFrame into an array of values
    - grabbed the value at index 0
    - assigned that value to studentScore

What did that get us?

- A scalar value
    - A numeric value (89, in this case)
    - Franklin's score!