

# Intro to Python

By Jessica Garson

# Set up

- To get online make sure you are on the Arlington public library's network
- Slides and other class resources can be found here:  
<https://github.com/JessicaGarson/PythonClass>
- My twitter handle is @jessicagarson – please tweet at me

## A note on this class

- We are going to attempt to create a working blog by the end of today's class
- We are also going to make a text based game
- This is a lot of material and we might not accomplish all of it today and that's okay

# My story



Introductions – Turn to someone you don't know and ask as follows

- Your name
- Favorite movie
- What brings you here today

# Why Python?

It's flexible enough to be used to create websites, do automation tasks, and be used a statistical package. It's a great beginner language.

# About this class

There will be two parts of today's class. Part one will focus on building our fundamentals. In part 2 we will make some internet happen.

# Let's make our own github profile

www.github.com

The screenshot shows a GitHub profile for Jessica Garson. The profile includes a profile picture, name, bio, location, email, and join date. It also displays statistics for followers, starred repositories, and following. The 'Popular repositories' section lists five repositories with their star counts. The 'Repositories contributed to' section shows one repository. The 'Contributions' section features a calendar grid showing activity from October to September.

Search GitHub

Pull requests Issues Gist

Contributions Repositories Public activity Edit profile

**Popular repositories**

| Repository  | Stars |
|---|-------|
| <a href="#">TwitterBot</a><br>Can be followed @jessgarsonbot          | 7 ★   |
| <a href="#">Simple_Pandas</a>   | 3 ★   |
| <a href="#">intro_to_command_line_git</a><br>Tech Lady Hackathon 2015 | 3 ★   |
| <a href="#">MachineLearningFridays</a>                                | 3 ★   |
| <a href="#">MovieSentiment</a><br>Kaggle Competition Code             | 2 ★   |

**Repositories contributed to**

| Repository   | Stars |
|--|-------|
| <a href="#">Burson-Marstel.../burson_tools</a><br>burson_tools | 0 ★   |

**Contributions**

Summary of pull requests, issues opened, and commits. [Learn how we count contributions.](#)

Less More



# Let's create a beginner account on pythonanywhere.com



[Send feedback](#) [Forums](#)

## Plans and pricing

### Beginner: Free!

A **limited account** with one web app at `your-username.pythonanywhere.com`, restricted outbound Internet access from your apps, low CPU/bandwidth, no IPython notebook support.

**It works and it's a great way to get started!**

Create a Beginner account

# Part 1

Python Basics

# For this section start python 2.7 console

## Start a new console:

---

Python: [3.5](#) / [3.4](#) / [3.3](#) / [2.7](#) / [2.6](#) IPython: [3.5](#) / [3.4](#) / [3.3](#) / [2.7](#) / [2.6](#) PyPy: [2.7](#)

Other: [Bash](#) | [MySQL](#)

Custom: [+](#)

# Strings

- Strings are a way to store information
- Strings have quotes around them and there are 2 different options for how to use them

`"This is a string"`

`'This is also a string'`

# Variables

- Variables are containers for information; you can store text, numbers, or any other type of thing!

```
name = "Jessica Garson"
```

```
age = 31
```

# Print

Use the `print` command tells python what it will display

```
print age
```

```
print name
```

Let's make some mistakes

```
print "hello friend"
```

```
Print "yo what's up"
```

```
print hello
```

# String formatting

- You can format your string to make it relevant to the task you have at hand

```
"Hello {0}, you are {1} years old".format(name, age)
```



# Math Operators

| Operator        | Operation                         | Example              | Evaluates to... |
|-----------------|-----------------------------------|----------------------|-----------------|
| <code>**</code> | Exponent                          | <code>2 ** 3</code>  | 8               |
| <code>%</code>  | Modulus/remainder                 | <code>22 % 8</code>  | 6               |
| <code>//</code> | Integer division/floored quotient | <code>22 // 8</code> | 2               |
| <code>/</code>  | Division                          | <code>22 / 8</code>  | 2.75            |
| <code>*</code>  | Multiplication                    | <code>3 * 5</code>   | 15              |
| <code>-</code>  | Subtraction                       | <code>5 - 2</code>   | 3               |
| <code>+</code>  | Addition                          | <code>2 + 2</code>   | 4               |

From Automate the Boring Stuff with Python

# Data types

| Data type              | Examples                               |
|------------------------|--|
| Integers               | -2, -1, 0, 1, 2, 3, 4, 5               |
| Floating-point numbers | -1.25, -1.0, -0.5, 0.0, 0.5, 1.0, 1.25 |
| Strings                | 'a', 'aa', 'aaa', 'Hello!', '11 cats'  |

From Automate the Boring Stuff with Python

You can convert into each of the data types

```
age = 31  
string_age = str(age)  
int_age = int(string_age)  
float_age = float(age)
```

Let's try some math!

$$2 + 3$$

$$3 * 5$$

$$66 / 3$$

# Slicing

- Python starts count at zero and you can use slicing to get chunks
- Let's create a variable called `python` in it that contains the string `python`

`python[0]`

`python[3]`

`python[2:]`

`python[:3]`

`python[2:5]`

| P | Y | T | H | O | N |
|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 |

# Operators

These are ways to compare two things:

**==**     These **two** things are equal

**!=**     **NOT!** equal to

**>**      Greater than

**<**      Less than

**>=**    Greater than or equal to

**<=**    Less than or equal to

Let's test this out together

**4 == 3**

**2 > 1**

# Conditionals

if – is a condition tells the program to do something

elif – allows you to have another condition tells the program to do something else

else – if none of the conditions meet the program will do another thing

```
if class_members == 10:  
    print "Yay"  
elif class_members < class_computers:  
    print "We might need some more computers"  
else:  
    print "I'm really looking forward to this class"
```

# Functions

- Functions allow you to divide your code into reusable chunks

```
def greet(name):  
    return "hello "+name  
  
greet_someone = greet  
print greet_someone("Jess")
```



# Lists

- If you have more than one of item that you want to contain in a variable (ex. student1, student2, student3) it would be annoying to write out variables for each of them.
- Lists are containers that can hold multiple pieces of information. Lists are commonly used to hold strings and numbers.
- We can also slice lists in the same way we did earlier in the class

```
attendees = ['Jess', 'Greg', 'Grace']
```

```
attendees.append('Chelsea')    -adds one to the end
```

```
attendees.extend('Tyson', 'Steve') -adds many to the end
```

# Dictionaries

Dictionaries are another way of storing information in Python. Dictionaries have two components: a key and its corresponding value. This is really great for storing information that has more than one value such as twitter handles and students.

```
student_twitter = { 'jess': 'jessicagarson',  
                    'bob': 'bobsmith', 'lakia': 'lakiajones' }
```

## For Loops

```
days_of_week = [ 'Monday' , 'Tuesday' ,  
                  'Wednesday' ]
```

```
for day in days_of_week:  
    print day
```

For each item in this list:  
 do something with that item

# While Loops

A for loop lets you use each item in a single list one at a time, which is great for performing actions a certain number of times. While loops are the cousins of conditionals.

```
if students >= 2:  
    print "I'm glad I'm not lonely in this  
class"
```

# Libraries

- Libraries are pre-made code that help you accomplish different goals.
- To install libraries in python you would do the following:

**`pip install requests`**

- At the top of your code you would use the following syntax

**`import requests`**

## Prompting People using Raw Input

```
name = raw_input("What's your name? ")  
age = raw_input("How old are you? ")  
color = raw_input("What's your favorite color?  
")
```

# How to run a program

Let's create a program that prints "hello world"

To run this program you can go into your terminal and type the following

```
python helloworld.py
```

# Creating a text based game

- You have 15 minutes to create a text based game based on what we just learned
- If you don't know where to start check out the example in my repository



# Show and Tell

- Let's break into groups and show each other our games

# Part 2

Creating a site with Flask

# What is Flask?

- Flask is a web platform for python . It allows you to quickly create sites with not a lot of set up.

# What about Django?

- Flask is a "microframework" primarily aimed at small applications with simpler requirement while django is a bit better for larger applications.
- Flask allows you to choose a database and django has one out of the box.

## Let's set up our flask app

- Go to the tab where it says web
- Set up a flask app using python 2.7

Edit your app to say something different from the default

```
1
2 # A very simple Flask Hello World app for you to get started with...
3
4 from flask import Flask
5
6 app = Flask(__name__)
7
8 @app.route('/')
9 def hello_world():
10     return 'Hi, I am making internet happen!'
11
12
```

# Let's create an app that looks like this

My scratchpad

This is the first comment.

This is the the second comment. It's no more interesting than the first.

This is the third dummy comment. It's actually quite exciting!

# Conclusion



# Do This at Home

## Shell Environments

- On a mac – there is a Linux environment that comes pre-installed called the terminal
  - To get to the terminal go to finder select utilities and inside the folder you should
    - On a PC – There are many options but I really like babun - <http://babun.github.io/>

## Text Editors

- Sublime Text - <http://www.sublimetext.com/>
- Atom - <https://atom.io/>
- Notepad ++ for PC only - <https://notepad-plus-plus.org/>

# Installing Python on your Computer— Macs

- Python comes bundled with Macs already, so there's no extra setup for now. You'll want to check which version you have, but it's not super important for writing your first scripts.

# Anaconda

- If you are looking to do more math, science and data work you will want to download the anaconda package of python.

<https://www.continuum.io/downloads>

# Installing Python on your Computer – PC and Linux

## Running Python in Windows or Linux

Download and install Python from here: <http://www.python.org/download/releases/2.7.6/>

I prefer to use IDLE (bundled with Python) over the command prompt when possible -- simply put, it's easier to use. If you need to run Python scripts from the command line, the Mac instructions below will work with minimal adjustment.

When you run IDLE, you'll automatically be in the interactive interpreter mode where you can run Python commands one at a time.

To create a new file from IDLE, go to File > New Window. A new, blank screen will open up where you can create your Python projects.

When you have a Python file open in IDLE, you can run it at any time by pressing **F5**.

# Resources for Learning Flask

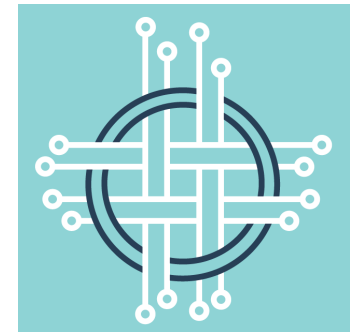
- The official flask tutorial - <http://flask.pocoo.org/docs/0.10/tutorial/>
- Mega flask tutorial - <http://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world>

The coolest thing about DC right now

ICODE  
FOR  
PROGRESS\_



DataKind



HACK  
AND  
TELL

*pyladies*

 Data Community DC

#maptime!



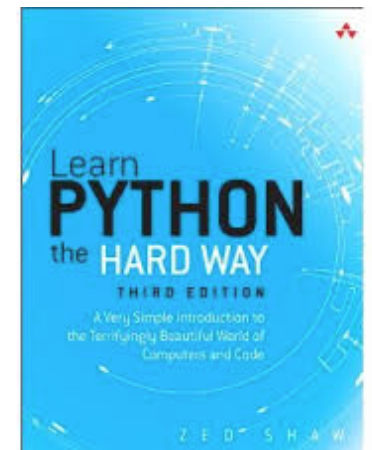
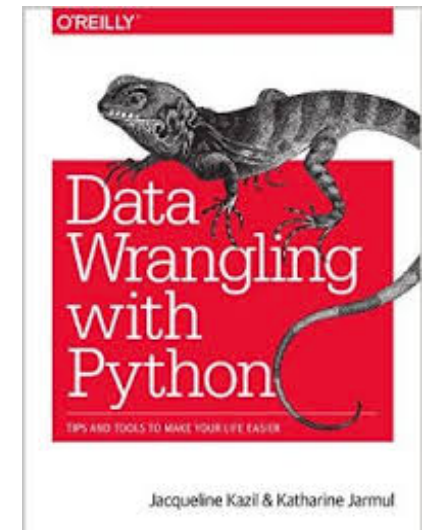
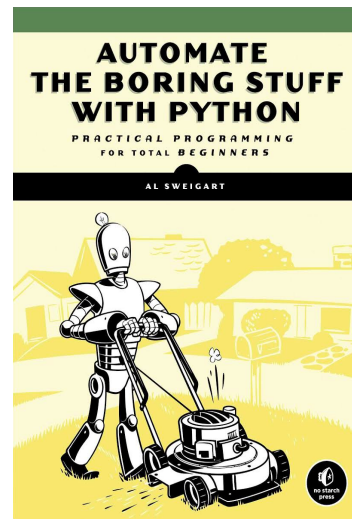
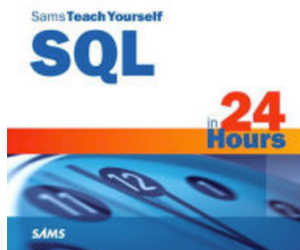
WOMEN WHO  
**CODE**

A few resources I recommend



Ryan Stephens  
Arie O. Jones  
Don Peck

SIXTH EDITION  
Includes Coverage of  
Oracle and  
Microsoft SQL  
Implementations



kaggle™

# See you around

- I'm a co-organizer for hack and tell - <http://www.meetup.com/DC-Hack-and-Tell/>
- I'm planning the Tech Lady Hackathon this year - <http://techladyhackathon.org/>
- I help out with Hear me Code - <http://hearmecode.com/>
- I also have a zine that can be found here – [bit.ly/whatsmyfunctionzine](http://bit.ly/whatsmyfunctionzine)



# Contact

Jessica Garson

[Jessica.Garson@gmail.com](mailto:Jessica.Garson@gmail.com)

[Jessica.Garson@bm.com](mailto:Jessica.Garson@bm.com)

Twitter: @jessicagarson