

# Syllabus: How to Build a TwitterBot

---

## Logistics

---

Instructor: [James Davenport, PhD](#) ([@jradavenport](#) on Twitter)

Email: [jrad@uw.edu](mailto:jrad@uw.edu)

Class Location & Time: TBD

A couple bots I've built: [@Apollo11Landing](#) ([blog post](#)), [@Mock\\_Twain](#) ([Medium post](#))

## Course Goals

---

In this course we will discuss the mechanics of building an automated Twitter account (or TwitterBot) using the Twitter API. Topics will include: a history of Twitter and TwitterBots, scheduling and automation, basic human computer interaction concepts, and the ethics of online bots.

The course is intended for general mid-to-upper division STEM majors who have basic programming ability in Python, R, or a similar language (see Prerequisites below).

By the end of this course you will be able to:

- implement basic automation of Python code using tools like `cron`
- use the Twitter API and other tools to interact with Twitter
- design, implement, and evaluate the impact of a simple TwitterBot
- consider the broader societal impacts of automated social media accounts

The course will be evaluated based on successful completion of homework assignments, participation in team discussions, development, and presentation of a TwitterBot, and a term paper.

## Prerequisites

---

There are no formal course prerequisites, but the course is designed for mid-level University

students who have a working knowledge of programming, nominally in Python. For a rough idea of programming expectations: students should be able to read CSV or text file, make basic [graphs](#), write an output text file, basic [string](#) manipulation, do basic math (e.g. via [NumPy](#)). Advanced knowledge of algorithms, AI/machine learning, etc. are not required. Students with such skills (e.g. CS or iSchool majors) are encouraged to integrate them into their 'Bots, but students from all majors will learn ample creative ways to build TwitterBots.

The course will be taught assuming students have a working knowledge of the Python programming language (Python 3, not 2). However, any language that can interface with the Twitter API is allowed, but the instructor cannot provide detailed support for languages besides Python. Regardless of language used, students are expected to include thorough documentation throughout their code. Just like in your math courses, Show Your Work!

A few technical requirements:

- Have a GitHub account to turn in assignments and host code
- You will need to sign up for a Twitter "[Individual Developer Account](#)". We'll talk about this more in Week 1 of the course
- I prefer to install Python via the [Anaconda distribution](#), which works on nearly all platforms (Mac, PC, linux) and includes most packages needed for data analysis and bot development.
- To interact with the Twitter API in Python, I prefer to use the [Tweepy](#) package, which can be installed via `pip`

## Recommended Reading

---

[Twitterbots](#), by Tony Veale and Mike Cook (2018, MIT Press)

## Code of Conduct

---

While the course will at times be informal and collaborative in nature, normal rules of academic classroom and professional behavior are expected. We seek to build an inclusive environment for learning. Please ask questions and engage with the instructor and your peers. Be respectful of those who are talking by actively listening. Disruptive, rude, harassing, or inappropriate behavior of any kind will not be tolerated, and academic misconduct will be reported to the University.

Please review the [Student Code of Conduct](#)

# Weekly Lectures

---

Class consists of weekly lectures on a variety of topics. Links to relevant materials, notes, or reading will be posted each week. Students are expected to attend all lectures, ask questions, and turn in assignments.

## Assignments

---

*in prep*

1. Find 3 examples of twitter bots. tell me their handle, in general what they do, and why you think they're a bot
2. Form team of 2 ppl, decide on what your bot will try to do. Create a short proposal (detailed instructions to follow)

## Grading

---

Students will be graded on successfully completing ~5 homework assignments, designing and implementing a TwitterBot as a team, and producing a final report (short term paper). We will adopt the [Informatics standard grading scale](#). Here is the Grade breakdown:

- 30% Homework
- 50% Bot Development and team participation
- 20% Final Report

No late work will be accepted by default. However, I fully understand that life happens -- please talk to me if you are having trouble making due dates!