Table of Contents

GIS 321 Principles of Programming for GIScience

Instructor and contact information:

Instructor: Sergio Rey

Office hours: Tuesdays 2-3 or by appointment

Email: srey@asu.edu

Course prerequisites: **CSE 110**

Course description:

Contemporary research in analytical geography has placed an increasing demand on the computational skills of its practitioners. The advances in spatial data analysis and geographical modeling have also largely out-paced the capabilities of standard statistical software. At the same time, the multidisciplinary nature of the spatial sciences often translates into the need to deal with disparate data sources, formats and programming languages. As such, students undertaking research are often confronted with a daunting set of tasks that are seldom covered in an integrated fashion in course work. This course is designed to address this situation.

Course learning outcomes:

- Introduce geography students to basic computational concepts using Python, an object-oriented scripting language, for data processing, analysis and application development in geographic research.
- Familiarize students with the fundamental tools used in collaborative programming and research projects in an open source and cross-platform environment.
- Provide students with skill sets that are in high demand within academic GIScience and commercial GIS development.

Grading policy:

Grading in the course will be based on the following point system:

Component	Points
9 Exercises	360
Exam 1	120
Exam 2	220
Exam 3	300
Total	1000

Exams will be based on the readings, discussion forum posts, and exercises. All exams are cumulative. Late exams will not be accepted.

Exercises will be completed and submitted on Github.

Extra credit will be given for accepted pull requests that provide corrections or enhancements to course materials. 10 points for each accepted pull request.

Required and recommended readings:

Wentworth, P., et al. (2012) *How to Think Like a Computer Scientist: Learning with Python 3. OpenBookProject* How to Think Like a Computer Scientist

Percival, H. (2014) *Test-Driven Development with Python. O'Reilly, Sebastopol.* Test-Driven Development with Python

Chacon, S. and Straub, B. (2014) *Pro Git.* Git: Distributed Even if Your Workflow Isn't

Other readings to be assigned.

Weekly activities All readings are from the How to Think Like a Computer Scientist text unless noted, e.g. with a `Git Ch1` descriptor.

Week	Date	Topics	Readings	Assigned	Due
1	8.22	Course Introduction, Intro to github	`Git Ch`	E0 GitHub	

2	8.29	Software installation, shells, git local	shells	E1 Shells	E0
	9.05	Labor Day Holiday			
3	9.12	git distributed, git Collaboration		E2 git collaboration	E1
4	9.19	Python Introduction, Test Driven Development		E3 Testing	E2
5	9.26	Continuous integration, Operators-operands		Exam 1	E3
6	10.03	Sequences, Dictionaries, Sets		E4	Exam 1
	10.10	Fall Break			
7	10.17	Functions, Modules		E5	E4
8	10.24	Object orientation		E6	E5
9	10.31	Composition		Exam 2	E6
10	11.07	Functional programming		E7	Exam 2
11	11.14	NumPy and SciPy		E8	E7
12	11.21	Visualization		E9	E8
13	11.28	Geospatial Python		Exam 3	E9
	12.05				Exam 3

Academic integrity

The ASU student academic integrity policy lists violations in detail. These violations fall into five broad areas that include but are not limited to: cheating on an academic evaluation or assignment, plagiarizing, academic deceit, such as fabricating data or information, aiding academic integrity policy violations and inappropriately collaborating, or falsifying academic records. For more information about the ASU student academic integrity policy, please use the following web link

http://provost.asu.edu/academicintegrity

Disability accommodations

Qualified students with disabilities who will require disability accommodations in this class are encouraged to make their requests to me at the beginning of the semester either during office hours or by appointment. Note: Prior to receiving disability accommodations, verification of eligibility from the Disability Resource Center (DRC) is required. Disability information is confidential.

Code of Conduct

As course instructor, I am dedicated dedicated to providing a harassment-free learning experience for all students, regardless of gender, sexual orientation, disability, physical appearance, body size, race, religion, or choice of operating system. All course participants are expected to show respect and courtesy to other students throughout the semester. As a learning community we do not tolerate harassment of participants in any form.

All communication should be appropriate for a professional audience including people of many different backgrounds. Sexual language and imagery is not appropriate in this course.

Be kind to others. Do not insult or put down other students. Behave professionally. Remember that harassment and sexist, racist, or exclusionary jokes are not appropriate for GIS321.

Students violating these rules may be asked to leave the course, and their violations will be reported to the ASU administration.

This code of conduct is an adaptation of the SciPy 2016 Code of Conduct.

Introduction to git and GitHub

Since this is our first meeting, the hands-on portion of the session will provide an overview of GitHub. The instructor will be doing the driving and the students can follow along on the lab computers. Next week you will install all the software for the course on your own laptop and the sessions will consist of you doing live coding.

In this first session, we introduce GitHub and cover:

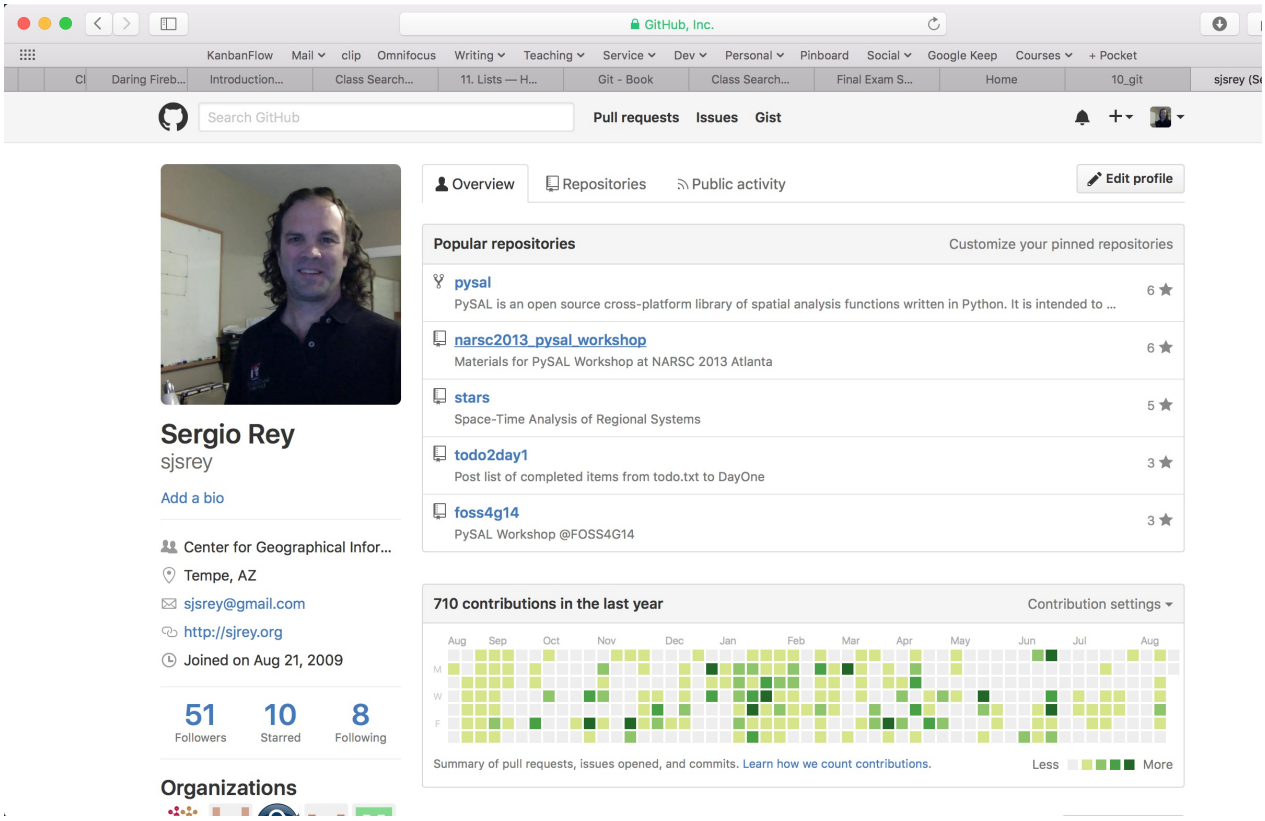
- What is GitHub
- Why use GitHub
- How to get started on your own

What is GitHub

From their website

> ## What is GitHub?
>
> GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.
>
> This tutorial teaches you GitHub essentials like *repositories*, *branches*, *commits*, and *Pull Requests*. You'll create your own Hello World repository and learn GitHub's Pull Request workflow, a popular way to create and review code.

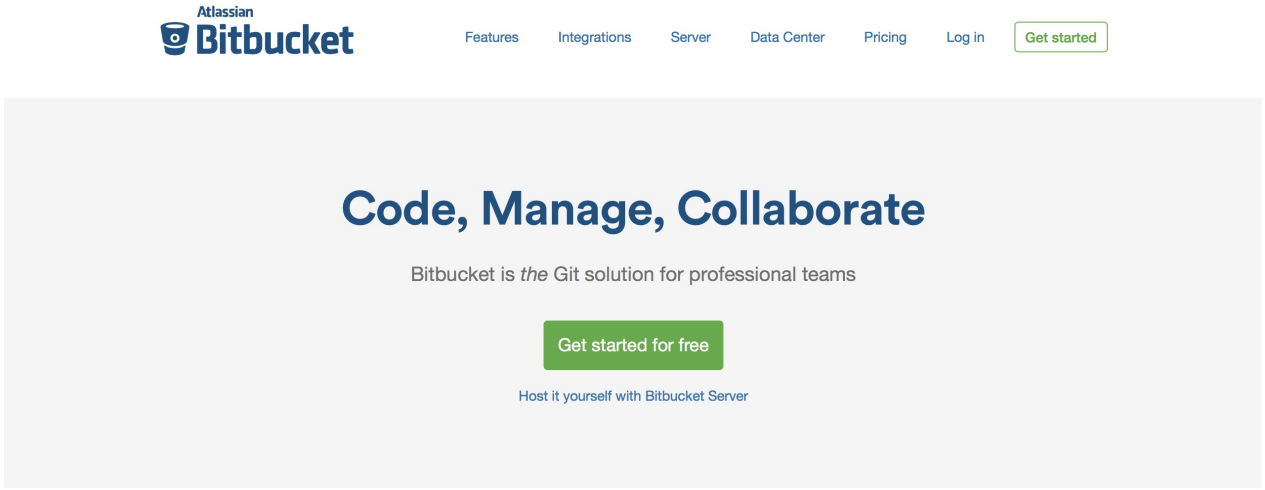
Why use GitHub

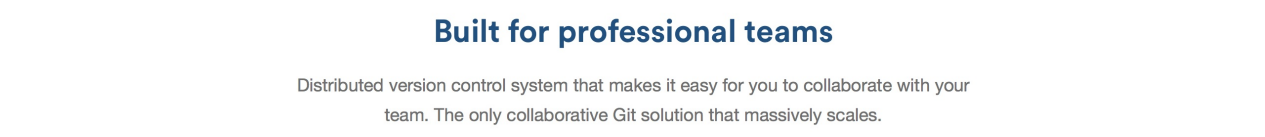


How to get started with GitHub

BitBucket

There is an alternative service called BitBucket that you may be interested in





Setting Up

This chapter guides you through the installation of the software used throughout the course as well as providing a discussion of different programming tools that will be useful in your computational work and research.

```python
```

Shells

Shells can be thought of as a type of interface that sit between you, a person, and the operating system.

Why Shells

The first time you encounter a shell you might be saying to yourself

> The 1970's called and they want their interface back.

Afterall, we currently live in the world of ubiquitous and (relatively) cheap laptop computers that typically come with a nice *graphical user interface*.

```python
```

Python

The language that we use in this course is Python. This chapter gives a high level overview of the language and motivates why we have selected it for this course.

Why Python?

Numpy

This chapter introduces the foundation of the scientific Python stack - *numpy*

```python
```