FS 102
Software Everywhere
Spring 2017

Practical 2
Assigned: Wednesday, February 1, 2017
Due: Wednesday, February 8, 2017 at the start of class
"Checkmark" grade

Introduction

Writers and presenters on the cutting-edge of technology often use a version control system to manage most of the artifacts produced during the phases of drafting and delivering an article or a talk. In this course, we will always use a GitHub repository to host the web site that will feature our writing and presentations. In this practical assignment, you will learn how to create a preliminary version of your mobile-ready web site and take the first step towards adding some of your existing content to that site. After finishing this assignment you should be able to view both a local version of your web site running on your development computer and a publicly available version of the site that is hosted by GitHub. As you are completing this practical assignment, please make sure that you consider the following admonitions about using GitHub to complete a writing assignment.

- **If possible, use the laboratory computers.** If it is absolutely necessary for you to work on a different machine, be sure to regularly transfer your programs to the Alden machines and check their correctness. Please remember that, as stated in the syllabus, students should try to complete assignments using the specialized workstations in the laboratory. If you cannot use a laboratory computer, then, when you are asking questions, please carefully explain the setup of your laptop to a teaching assistant or to the course instructor.
- **Follow each step carefully.** Slowly read each sentence in every assignment sheet, making sure that you precisely follow each instruction. Take notes about each step that you attempt, recording your questions and ideas and the challenges that you faced. If you are stuck, then please tell a teaching assistant or instructor what step you recently completed.
- **Regularly ask and answer questions.** Please log into Slack at the start of a class or practical session and then join the appropriate channel. If you have a question about one of the steps in an assignment, then you can post it to the designated channel. Or, you can ask a student sitting next to you or talk with a teaching assistant or the course instructor.
- **Store your files in Git**. Starting with this laboratory assignment, you will be responsible for storing all of your files in a Git repository. Please verify that you have saved your source code in your Git repository by typing "`git status`" and ensuring everything is up to date.
- **Keep all of your files!** Don't delete your programs, output files, and reports after you hand them in—you will need them again later when you study for the quizzes and examinations and work on the other laboratory, practical, and final project assignments.
- **Back up your files regularly**. Use a flash drive, Google Drive, or your favorite backup method to keep a copy of your files in reserve. In the event of a system failure, you are responsible for ensuring that you have access to a recent backup copy of all your files.

Configuring Git and GitHub

During this practical assignment and subsequent assignments, we will securely communicate use GitHub to host our writing and presentations. If you did not complete these steps in a previous practical assignment, then you must now ensure that you have configured your accounts on the departmental servers and the GitHub service. Throughout this assignment, you should refer to the following web site for more details: `https://guides.github.com/activities/hello-world/`. As you will be required to use Git in the remaining writing, speaking, and practical assignments and during the class sessions, please be sure to keep a record of all of the steps that you complete and the challenges that you face. If you confront and then ultimately resolve a confusing issue, please share your experiences in the `#practicals` channel of Slack team for this course.

1. If you do not already have a GitHub account, then please go to the GitHub web site and create one—make sure that you use your `allegheny.edu` email address so that you can join the GitHub Educational Community as this step becomes necessary. Also, please make sure that you add a description of yourself and an appropriate professional photograph to your GitHub profile. For examples of what a professional GitHub profile might look like, please consider studying `https://github.com/una` and `https://github.com/gkapfham`.

2. If you have never done so before, you must use the `ssh-keygen` program to create secure-shell keys that you can use to support your communication with the GitHub servers. But, to start, this task requires you to type commands in a program that is known as a terminal. To run it, on the left side of your screen, click on the icon that contains the "`>`" symbol. Alternatively, you can type the "Super" key, start typing the word "terminal", and then select that program. Another way to open a terminal involves typing the key combination `<Ctrl>-<Alt>-t`.

3. If you have not done so already, you will now need to run the `ssh-keygen` command in your terminal window. Follow the prompts to create your keys and save them in the default directory (press "Enter" after you are prompted: "`Enter file in which to save the key ... :`", then press "Enter" twice if you do not wish to create a passphrase at this time or type your selected passphrase if you do). What files does `ssh-keygen` produce? Where does this program store these files by default? Do you have questions about this step?

4. Once you have created your ssh keys, you should raise your hand to invite either a teaching assistant or the course instructor to help you with the next steps. First, you must log into GitHub and look in the right corner for an account avatar with a down arrow. Click on this link and then select the "Settings" option. Now, scroll down until you found the "SSH and GPG keys" label on the left, click create a new "SSH key", and then upload your ssh key to GitHub. You can copy your to SSH key to the clipboard by going to the terminal and typing "`cat ~/.ssh/id_rsa.pub`" command and then highlighting this output. When you are completing this step in your terminal window, please make sure that you only highlight the letters and numbers in your key—if you highlight any extra symbols or spaces then this step may not work correctly. Then, paste this into the text field in your web browser.

5. Again, when you are completing these steps, please make sure that you take careful notes about the inputs, outputs, and behavior of each command. If there is something that you do not understand, then please ask the course instructor or the teaching assistant about it.

6. Since this is your first practical assignment and you are still learning how to use the appropriate software, don't become frustrated if you make a mistake. Instead, use your mistakes as an opportunity for learning both about the necessary technology and the background and expertise of the other students in the class, the teaching assistants, and the course instructor. Remember, you can use Slack to talk with the instructor by using "`@gkapfham`" in a channel.

Creating a Mobile-Ready Web Site

As you complete the next part of this practical assignment please make sure that you follow each step carefully. If you make a mistake in one of these steps it may require you to start over and follow all of the steps again. If you are not sure how to do one of the requested actions, then please ask the course instructor or a teaching assistant. To start this phase of the assignment, please visit and read the following web site: `https://github.com/daattali/beautiful-jekyll`.

1. After reading the aforementioned web site, you will know that "Beautiful Jekyll" is a template for creating a mobile ready web site by using the Jekyll static site generator and the Bootstrap responsive layout framework. Before you start the next steps, please watch the movie on this web site to get a sense as to how you are going to download and customize this theme by using GitHub. Next, you should fork the repository by clicking the appropriate button that is in the top right corner of your web browser. What do you think that forking accomplishes?
2. Now that you have forked this repository, you need to go into its settings and rename it to "`<your GitHub user name>.github.io`". Please see the course instructor or a teaching assistant if you do not know your GitHub user name (which should be, if possible, the same as your `allegheny.edu` email user name) or you are not sure how to rename the repository.
3. Using SSH — and not HTTPS! — you should now clone the repository to your laboratory computer. You will do this by clicking the green "Clone" button, copying the address to the clipboard, and then typing "`git clone`" and pasting the address in your terminal window. At this point, you should see the download of the source code for your new web site to your workstation. Ask the instructor for help if you think that this did not work correctly.
4. Next, you should change into the directory for your web site's repository and type the command "`bundle install --path ~/.gem`" and wait for all of the needed packages to install correctly on your computer workstation. Since this command may take a long time to run, you should start it now and then, if necessary, move to the next step in the assignment.
5. Once the packages have installed correctly, you can type "`bundle exec jekyll serve`" to transform the markdown of your web site into HTML and to start a local web server that you can use to preview your site. If this command works, then you can access the web server by typing `http://127.0.0.1:4000` into your web browser. Can you see a web site now?
6. After you have gained access to the local (i.e., "development") version of your web site, it is also a good idea to view the version that is currently hosted by GitHub and thus publicly available. To accomplish this step, you should return to your web browser and go into the settings for your GitHub repository. Now, scroll down until you see the part of the screen with the label "GitHub Pages" and make sure that you are serving content from the master branch of your repository. You should now see a green checkmark and a label that gives the web site that is currently hosting your page. Can you find it? If not, then please ask for help from the instructor. Now, click on this link and then view the public version of your site!

7. If you still have the "`bundle exec jekyll serve`" command running, you can shut it down by typing "`CTRL-c`" in your terminal window. Now, start Atom and load the file called `_config.yml`. In this file, you should change the fields for the URL, title, and description to correspond to values that are most appropriate for your web site. In the section for the links in the navigation bar, you should (at least for now) delete all of them except for the one to "`aboutme`". Finally, go to the bottom of this file and provide information for all of your social media accounts and your contact information. Please be sure that if you have, for instance, a Twitter account that you set the flag for this to `true` in the configuration file.
8. Now, please locate the "About Me" Markdown file that you completed in a previous assignment. Using Atom, copy the contents of this file to the clipboard. Next, locate the file called "`aboutme.md`", delete the content in this file — while taking care not to delete the header information at the top of the file — and paste in the material from your previous assignment. To complete this step, you should also update the header so that it contains appropriate descriptive details for this file. Now, go back to your terminal window and run "`bundle exec jekyll serve`" and reload the locally-served content in your web browser.
9. If you can see changes to the content, then you will have confirmation that your edits are working correctly. Before making this writing publicly available, you should carefully check it to ensure that it is compelling and error-free. After fixing any mistakes that you noticed, please use the "`git commit`" and "`git push`" commands to ensure that your new content is also on the version of your site hosted by GitHub. Can you correctly view your new site?
10. Finally, if you look in the GitHub repository for your new web site, you will notice that it still has the `README.md` file that came with the original version of the "Beautiful Jekyll" template. You should delete the majority of the content in this file and add in some of your own original content. While it is appropriate for you to acknowledge that you created your web site with the help of this template, you should customize the `README.md` with the steps that someone would need to take to download and fully use your new web site. You can use the content in this assignment sheet as a source for the content that you write in this file.
11. After you have a working version of your web site, please paste the reference to the GitHub repository and to the web site into the `#web` channel of our Slack team. Then, please review the web site of at least one other member of the course. Once you find a way to improve your colleague's writing, please raise an issue in the issue tracker of their GitHub repository.

Summary of the Required Deliverables

This practical assignment invites you to complete the following tasks:

1. A cloned version of the "Beautiful Jekyll" template that is customized for your web site.
2. A new "About Me" page that contains the revised version of your previous writing assignment.
3. A customized version of the `README.md` file that is in your web site's repository.
4. An issue that you raised in the GitHub issue tracker for your colleague's web site.

In adherence to the Honor Code, you should complete this practical assignment on an individual basis. While you may have high-level conversations with others, any deliverables that are nearly identical to the work of others will be taken as evidence of violating Allegheny College's Honor Code.