CSC 301, Software Installation

Overview

In this class we will use several tools for web development and web scraping, including editors for developing web pages using HTML/CSS/JavaScript, Python for web scraping, and R for web development. Instructions for installing or accessing the necessary tools are provided below.

Text editors for web development

In general, we will use Glitch (https://glitch.com/), an online editor for developing and rendering web pages in the browser. It is recommended that you create a free account on Glitch so that you can save your projects.

In some cases, we will also need to use standalone editors. While many are available, I recommend the two below, primarily for their strong functionality yet ease of use.

- For **Windows** users, I recommend downloading Notepad++ (https://notepad-plus-plus.org/), a user-friendly text editor that has syntax highlighting for many languages. Notepad++ is what we will use in class.

- For **Mac** users, I recommend downloading Brackets (http://brackets.io/) , a text editor designed for web developers.

Installing the Anaconda Individual Distribution

Python (https://www.python.org/) is a general purpose programming language and *Jupyter Notebook* (https://jupyter.org/) is a web-based platform for creating, explaining, and sharing code. We will use both for web scraping and automated web browsing.

Install the Anaconda Distribution for your system by following the directions at the following link: https://www.anaconda.com/products/individual

The Anaconda Distribution comes with both *Python* and *Jupyter Notebook*.
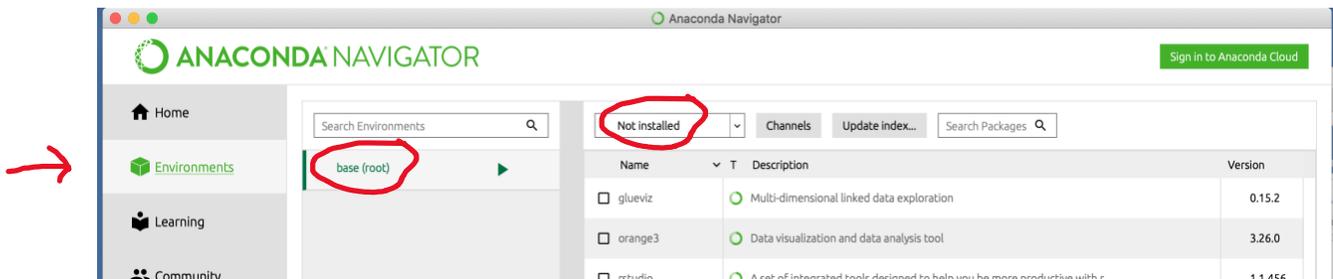
Running Jupyter Notebook

Open Anaconda Navigator, and then click Launch under the Jupyter Notebook app.
On school computers, you can simply double click the Jupyter Notebook icon.

Installing Selenium

Selenium (https://selenium.dev/) is a framework for browser automation (for writing code that opens and uses web browsers). We need to install the *selenium* python module, which can be done from Anaconda Navigator

To select packages for installation, first click on "Environments", and make sure that the "base (root)" environment is selected. Then change the dropdown from "Installed" to "Not installed" or "All".

Install a module by typing the module name in the "Search Packages" box to find the module, then checking the check box next to the module name. You may select multiple modules for installation at the same time. When ready for installation, click "Apply" on the bottom right., and click "Apply" again following the notification of the packages that will be installed. Note that the installation process may take some time. However, if it takes more than 10 minutes, or if you have any other trouble installing packages let me know!

Search for the *selenium* module and install it following the instructions above.

Installing the Firefox webdriver

In addition, you will need to download a *webdriver* which is what allows Python to control a web browser. We will use Firefox in class along with webdriver, called *geckodriver*.

Firefox can be installed from https://www.mozilla.org/en-US/firefox/new/.

You will need to download the appropriate webdriver and also put it in an appropriate location for Selenium to work. Download the appropriate driver from the Assets section at the bottom of the following page, selecting *geckodriver-v0.30.0* for your system, and extract the *geckodriver* file: https://github.com/mozilla/geckodriver/releases/tag/v0.30.0

Move the driver to an acceptable location on your computer's *path* by following "section 2. The PATH Environmental Variable" from the following page: https://www.selenium.dev/documentation/webdriver/getting_started/install_drivers/

If installed correctly, you should be able to execute the following commands in Python without any errors. The code below uses Selenium to open Eastern's homepage.

```
from selenium import webdriver
driver = webdriver.Firefox()
driver.get('https://easternct.edu')
```

Installing R/RStudio

R (https://cran.rstudio.com/) is a programming language for data visualization and analysis, and *R Studio* (https://rstudio.com/) is an integrated developer environment (IDE) for *R*. We will use *R* and its *shiny* package to create interactive, data-driven web pages.

1. Install *R* for your system from https://cran.rstudio.com/
2. Install RStudio by clicking the download button on https://rstudio.com/products/rstudio/download/#download and following the instructions.

R package installation

In class we will be using the packages *dplyr* (for manipulating data), *shiny* (for development of web applications), and *rvest* (for web scraping).

These packages can be installed by running the commands below from within R Studio:

```
install.packages('dplyr')
install.packages('shiny')
install.packages('rvest')
```

Note that during installation you will receive red messages (and possibly warnings) showing the progress of the installation. This is normal. However, if you get an error, then the package was not installed and you should contact me and I will help troubleshoot the issue.

After a package is installed, it can be loaded using the *library* function, as in the code below:

```
library(dplyr)
library(shiny)
library(rvest)
```