CSCI S-33a (Web50) Section 1

Ref: Lectures 1+2 (Git + Python)

Vlad Popil

Jun 25, 2020

Welcome again!

About me:

Vlad Popil
Master's **(**ALM) in Software Engineering
Principal Data Analyst at Capital One

Email: volodymyr.popil@gmail.com

Sections: Wed 8:00-9:30 pm EDT **+** 1st week only on Thu 8:00-9:30 pm
Office Hours: Sat 2:00-3:30pm EDT

Agenda

- Intro/Suggestions
- Sections 10,000 foot overview
- Git / GitHub
- Python
- Anaconda distribution
- Project 0 + submit50
- Grading criteria (not exhaustive)
- IDEs
- Tips
- Q&A

Logistics

TODO This Week

- ❑ Sign up for Ed on the course website
- ❑ Watch Lectures 0, 1, and 2
- ❑ Attend 2 x Sections
- ❑ Fill out the course software form to get ready to submit projects
- ❑ Complete Project 0 by 11:59 pm ET on Sunday, June 28
- ❑ Complete the Project 0 form by 11:59 pm ET on Sunday, June 28

Reminders

- Zoom:
  - Use zoom features like raise hand, chat and other
  - Video presence is STRONGLY encouraged
  - Mute your line when not speaking (enable temporary unmute)
- Projects:
  - Start early (or even better RIGHT AWAY!!!)
  - Post questions on Ed platform
  - Remember: bugs can take time to fix
  - Grade -> 3 × correctness + 2 × design + 1 × style
  - Lateness policy - 0.01 per minute x 60 x 24 x 7 days ~ 100
  - Set a reminder to submit the form for each project
  - Online search, Ed platform, etc.
  - Documentation
  - Project 0 - Due Sunday, Jun 28 at 11:59pm EDT (**3 FULL DAYS LEFT**)

Reminders

- Sections/Office Hours:
  - Sections are recorded, office hours are not
  - Real-time attendance encouraged
  - Video and participation encouraged even more
- Section prep:
  - Watch lecture
  - Review project requirements
- Office hours prep:
  - Write down your questions as you go, TODO, etc.
  - Come with particular questions

10,000 foot overview

- *Section 0 (HTML, CSS)* - Chrome Dev Tools (Inspector), Grading aspects, Overviews
- *Section 1 (Git + Python)* - Python Installation, IDEs, CDT (Network), Project 0
- *Section 2 (Django)* - Markdown, RegEx, IDEs extra, pycodestyle, Debugging, Project 1
- *Section 3 (SQL, Models, Migrations)* - cURL/Postman, Models, DB modeling, Project 2
- *Section 4 (JavaScript)* - AJAX, linting, jshint, CDT Debugging, Project 3
- *Section 5 (User Interfaces)* - Animations, DB modeling, Pagination, Project 4
- *Section 6 (Testing, CI/CD)* - Test Driven Development, DevOps, Final Project
- *Section 7 (Scalability and Security)* - Cryptography, CAs, Attacks, App Deployment

Most sections: material review, logistics, project criteria review, reminders, hints, etc.

Burning Questions?

Please ask questions, or topics to cover today!

Topics:

- *Submit w/out submit50?*
- *Venv or Anaconda?*
- *Handling `self` for class methods?*

Git/GitHub

Git/GitHub

- **Git:** Version control software for tracking changes locally
- **GitHub:** Place to store Git repositories remotely and share them with others
- **Repository:** A directory containing all files/directories associated with a project
- **Branch:** A version on your project where you can work on new features
- **Commit:** Save a snapshot of your repository
- **Push:** Upload your local repository to a remote website
- **Fork:** Create a copy of your another person’s repository that you can edit
- **Merge:** Combine two branches
- **Merge Conflict:** When manual intervention is required to merge two branches
- **Pull Request:** An attempt to merge two branches that must be approved

Repository for Project 0

1. Head to GitHub and create a repository named for the first project
2. If you haven't started the project:
   a. Download and unzip the source code from the website
3. Navigate to the directory the project is in.
4. Run `git init` to initialize a repository
5. Run `git remote add origin REPO_URL` to link local repo to GitHub repo
6. Add and commit your changes regularly as you work on the project:
   a. `git status`
   b. `git add .`
   c. `git commit -m "blah blah changes"`
   d. `git push`

Git

- Track changes of code
- Sync the code between different people or projects
- Revert to old versions of code

Most popular commands:

- clone
- add
- status
- commit
- push
- pull

Git

Let’s create new repo and try the basics!

Git

- .gitignore / .gitkeep
- Branching:
  - branch
  - checkout
  - merge
- Pull Requests
- Demo Forking: https://github.com/octocat/Spoon-Knife

Python

Interesting Concepts

Running Python Programs

- Files should be in the form `file_name.py`
- Run a file by running `python file_name.py` in the terminal.

Sequences

Sequence Type	Ordered	Mutable	Example
String	Yes	No	name = “Bob”
List	Yes	Yes	ls = [1, 3, “hi”, True, [1, 2, 3], None]
Tuple	Yes	No	coordinates = (4, 5)
Set	No	N/A	odds = {1, 3, 5, 7, 9}
Dictionary	No	Yes	person = {“name”: “Bob’s”, “age”: 20}

Modules

- Allow us to import code from other files
- Allow us to import code written by others
- For a file named **name.py**, we can write **import name**

Formatting Strings

- Used to more easily incorporate variables in strings
- Can include function calls and expressions in curly braces
- More formatting options here

```
f“Hello, {name}”
```

```
f“{x} times {y} is {x * y}”
```

Exceptions

- If we expect an exception to be thrown in a certain place, we can handle it without any crashes.

```
x = int(input("Type a number to find it's square root: "))

try:
    print(math.sqrt(x))
except ValueError:
    print(f"Could not take a sqrt of {x}")
```

Optional Arguments

- Arguments can have default values if none are provided
- We can call a function using either the order of arguments or their names.

```
def square_root(x, truncate=False)
    if truncate:
        y = int(math.sqrt(x))
    return math.sqrt(x)

square_root(4)
square_root(4, True)
square_root(truncate=True, x=4)
```

Python

Interactive examples:

- Jupyter Notebook
- Demo...

Anaconda Distribution

- Anaconda - World's Most Popular Python/R Data Science Platform
- Miniconda (lighter version):
  a. Download https://docs.conda.io/en/latest/miniconda.html
  b. Run in terminal in Downloads: `zsh Miniconda3-latest-MacOSX-x86_64.sh`
  c. Run `conda init` ONLY if not prompted during installation
  d. Create new environment: `
     - `conda create -n s33a python=3.7`
  e. See environments:
     - `conda env list`
  f. Deactivate/Activate environment:
     - `conda deactivate`
     - `conda activate s33a`
  g. Install more packages:
     - `conda install django` (preferred)
     - `pip install django` (if conda doesn't find), although
       It is better to `python -m pip install django` (to assure proper pip)

Chrome Developer Tools (Inspect / Network)

In Chrome:

1. Right click
2. Inspect
3. → Demo

Extremely powerful! Let’s try...

Project 0 (Repeat)

- Start early!!!
- Make a checklist of requirement and check all before submission
- Don’t forget to include the .css / .scss file(s)
- Make sure there’s no bugs
- Google Form
- *Next page...*

Project 0

1. Requirement review
2. Chrome developer tools
3. Dual button review
4. Double key:value
5. `submit 50`

Integrated Development Environments (Intro)

- Text Editor or Heavy IDE?
- Options:
  - VS Code
  - PyCharm (Pro)
  - Atom
  - Sublime
  - vim/Emacs
  - And dozens more, including Notepad :)
- My suggestion: VS Code or PyCharm
- Benefits: Debugging, Autocomplete, Navigation, Find Usages, Refactoring, and much more.

Grading criteria generic suggestions (not limited to)

- Correctness:
  - All requirements
- Design (not limited to):
  - Responsive
  - Simplest solution
  - Avoiding repetition (refactoring)
  - Structure (e.g separate files vs inline styling)
- Style (not limited to):
  - File structure
  - Line breaks
  - Spacing
  - Naming
  - Comments

Both Design and Style consider readability but from different perspective.

Resources

- https://github.com/vpopil/e33a-sections-summer-2020

Random Tips

- HTML beautifiers/prettify
- Windows licence
- The Great Suspender
- GitHub Education Pack
- Spotify
- Video Speed Controller

Fruit of the day

<<< If you are watching this recorded >>

Please email the word: **PINEAPPLE**

To: volodymyr.popil@gmail.com

Thank you.

Q&A

Please ask any questions. Ideas:

- Anything discussed today
- Anything from lecture material
- About the project
- Logistics
- *Random*

Resources

- https://github.com/vpopil/e33a-sections-summer-2020

CSCI S-33a (Web50) Section 1

Ref: Lectures 1+2 (Git + Python)

Vlad Popil

Jun 25, 2020