Over the course of the semester as you start to learn how to code, you will start to see the world a little bit differently. That is, you'll start to better understand how coding shapes our lived existence. Maybe you will learn about a digital project in one of your other classes; maybe you learn something from our assigned weekly readings; maybe you hear about something in the news and go, "Hey, I think I know how coding could make that easier"; maybe you follow a coder on TikTok and now know what the heck they're talking about. This is what I call seeing "coding in the wild."

I have set up a discussion board located on our main Brightspace page (NOT your lab page). I want you to post on it five (5) times - no more than one per week (a week is defined as Sunday-Saturday). That means in our 16 total weeks of class, you have a lot of opportunities to complete this assignment. For this reason, I do not accept late posts and you cannot make up posts. If you run into any technical issues when submitting your post on time, email me the post content ASAP to get credit.

Each post will be graded as pass/fail. (Coding in the Wild assignment as a whole is not pass/fail, which means you can get full credit for submitting 5 passed posts, or partial credit for less than 5 passed posts.) To pass (100pts) you must fulfill all the following guidelines, otherwise, you will receive a 0/Fail:

Can only submit one per week (defined as Sunday-Saturday)

Your post does not have to relate to what we're specifically talking about in a given week, it just has to be about coding. (cannot be about technology in general)

It should be 200 words long, minimum

Fully describe your experience - maybe that means describing a conversation; or perhaps you need to link to a meme, video, etc.

Connect this to our course - how has our course helped you "see" this particular thing?

1/27/2024

Joshua Cheuk- Entry #1

While volunteering at a hospital, I had to sort a lot of patient data by looking for specific words in the diagnosis column, the comorbidity column, and the patient history column. It was extremely tedious having to scroll and click through the hundreds of patient's data, manually read through their files, and mark down whether or not the patient had a certain disease in their lifetime on an Excel spreadsheet.

However, this problem reminded me of this video:
https://www.youtube.com/watch?v=hBP-NzOadL0&t=374s

In the video above, the Youtuber makes a bot that reads through Twitter comments and purchases items that were commented on twitter by his fans off Amazon. I found the idea of a program reading information from one site and implementing the information on a different site to be an interesting solution to the monotony I was experiencing working in the hospital. At the time, I wasn't experienced enough to write code and this idea would've likely been shut down by my manager because I was using an untested program/product with patient data, which could make mistakes while extracting the data. However, I still think that creating a program to scape data is a skill I should learn because there are lots of instances where I have to deal with monotonous data acquisition and data cleaning.

2/3/2024
Entry 2

Debugging reminds me of organic chemistry.

In organic chemistry, we have to solve problems with reactants to get a product. When we are solving these problems we need to keep in mind certain concepts and rules. Some of these concepts and rules are the type of atom included in the molecule, the shape of the molecule, the specific properties that pertain to the molecule and how it affects the product, and the type of atom being added to or taken away from the molecule. Similarly, when you're debugging your code you need to keep in mind whether or not you wrote a variable's name correctly, whether or not the thing you wrote down in code is in a dictionary, whether or not you picked the correct symbol, and whether or not you wrote the code in a way that follows the syntax rules of python. If any of these rules and concepts are broken the desired chemical product or print result will not come out. To get the desired result, you have to check if any of the chemical reaction rules and concepts or the coding rules and concepts are correct to get your desired result. I find this similarity fascinating because it really shows that both organic chemistry and coding is a language with its own respective rules and ideas you follow to create or show a desired result.

Entry 3:
2/10/2024

Learning about lists and dictionaries helps me understand how and why games take a long time to develop.

After watching the disappointing release of “Suicide Squad: Kill the Justice League,” it made me wonder what goes into game development and why game developers complain about burnout and being overworked. After watching a video about how to use dictionaries in game development and looking at games that have lots of items, things you can interact with, information on your user interface (like damage points on your screen), voice lines NPC’s and enemies say, skill trees, crafting recipes, mission objectives, and the NPC and enemy’s behaviors that change based on the player’s interactions made me realize that there are lots and lots of dictionaries being written so the player can see, interact, and do things in the game world.

Where I learned how dictionaries are used in games:
https://www.youtube.com/watch?v=vditeSiJdSY

In my experience writing dictionaries and lists, this can be very, very tedious. The fact that dictionaries play such a crucial role in the functioning of a game is astonishing to me, now that I understand how dictionaries and lists can be used in game development. It also helps me understand why game developers are overworked because it takes time to troubleshoot and perfect functions in the game.

Entry 4:

2/16/2024

Logging into the computer helped me see the dictionaries, while loops, and if statements used in this user interface

When I logged into the computer and accidentally typed the wrong password a couple of times, I realized that it was using dictionaries, while loops, and if statements to tell me that I either didn’t use the right password or username. The dictionary contains all the usernames and passwords of students, faculty, employees, and other people who get access to Binghamton University’s computers. If the user’s input in the username matches the username in the dictionary, and if the user’s input in the password matches the password in the dictionary, access to the computer will be granted.