

Reflectie Python Eindopdracht – Fabian Dekker

Inhoud

1. Wat ik heb geleerd

- 1.1 Werken in Python
- 1.2 Git workflow
- 1.3 Werken met Pseudo Code

2. Waar ik tegenaan liep

- 2.1 Scope creep
- 2.2 Time blindness

3. Takeaways

- 3.1 Meer uit Git halen
- 3.2 Timeboxen
- 3.3 Duidelijk leerdoel stellen en daar op focussen
- 3.4 Hoofdzaak van bijzaak scheiden

Wat ik heb geleerd

1.1 Werken in python

Ik kan zeer comfortabel werken in python. Ik heb veel tijd gestoken in uitvogelen hoe ik bepaalde functies op de beste manier kan schrijven en implementeren. Ook daagde ik mezelf uit om eerst zelf tot een bepaald resultaat te komen voordat ik het internet raadpleegde. Dit heeft ertoe geleid dat ik gemakkelijk werk in python en goed weet hoe ik kennis die ik nog niet bezit kan vergaren. (Werken met modules, documentatie lezen, weten welke zoektermen ik gebruik om de juiste informatie te vinden)

1.2 Git workflow

Ik heb geleerd over version control en ik heb goed leren werken met GitHub. Ik ben er ook achter gekomen dat deze werkwijze voor mij persoonlijk heel goed werkt. Ik heb vanuit mezelf de neiging om een project heel sporadisch aan te pakken en alles tegelijk te doen. Dit zorgt er in het begin voor dat ik snel tot resultaten kom, maar zodra het project gecompliceerder wordt werkt dit me juist tegen omdat ik vastloop wanneer ik alle facetten van de opdracht meer tegelijk in mijn hoofd kan houden.

Werken met commits moedigt voor mij een workflow aan waarbij ik de opdracht in stukjes verdeel, en dan steeds een deel afmaak voordat ik verder ga naar het volgende stukje. Deze werkwijze voorkomt dat ik vastloop.

1.3 Werken met Pseudo Code

Aan het begin van de lessenreeks zag ik niet echt het nut van pseudo code. Ik had al snel een plan van hoe ik de eindopdracht wilde aanpakken en begon meteen met coderen. Toen, wanneer we de uitleg over pseudo code kregen, moest ik de code die ik al had geschreven opeens vertalen naar pseudo code wat verwarring was. Echter, tijdens het werken aan de opdracht kwam ik erachter hoe handig het was om eerst een ontwerp maken van een functie waarbij ik me geen zorgen hoeft te maken over de juiste syntax. Voorheen werkte ik heel iteratief, en was hierdoor lang bezig met bugs uit de code halen. Nu met pseudo code hoef ik alleen maar de pseudo code te vertalen naar de juiste syntax. Hierdoor kan ik sneller werken

Waar ik tegenaan liep

2.1 Scope creep

In mijn enthousiasme heb ik bij de opdracht een zeer ambitieus doel gesteld voor mezelf. Ik heb naast de lessen uit jaar 1 verder geen ervaring met programmeren. Veel dingen waren dus nieuw voor me en hebben me een hoop tijd gekost. Daarnaast kreeg ik terwijl ik bezig was met de opdracht steeds meer ideeën die ik heel koppig allemaal bleef implementeren. Uiteindelijk ben ik hierdoor in de knoop geraakt met tijd en heb ik belangrijke delen van mijn ontwerp moeten schrappen.

2.2 Time blindness

Soms heb ik niet goed door hoe lang ik achter elkaar werk. Dit heeft ervoor gezorgd dat ik op momenten veel te lang doorging. Als ik lang achter elkaar werk ga ik, zonder dat ik dit door heb, steeds langzamer werken door vermoeidheid die opbouwt. Ik heb hierdoor onnodig veel tijd in de eindopdracht gestoken en ik heb mezelf behoorlijk overwerkt wat invloed heeft gehad andere vakken die ik volg en mijn gezondheid.

Takeaways

3.1 Meer uit Git halen

Ik ben al voor een groot deel gewend aan de workflow met Git, toch denk ik dat ik meer uit GitHub kan halen. Ik heb namelijk nog niet gewerkt met development branches en het combineren van verschillende branches. Ook heb ik maar een paar keer gepulled vanaf een repository. Ik wil het pullen van de sessions die op de csd2 git staan opnemen in mijn workflow zodat ik good practice voorbeelden meer bij de hand heb zodat ik niet steeds het wiel opnieuw aan het uitvinden ben (ook al leer ik daar wel veel van). Daarnaast heb ik ontdekt dat het voor mij voor overzicht fijn om de Markdown editor Obsidian te gebruiken om het opdrachten.md bestand te bekijken. Kortom, ik kan veel dingen overzichtelijker maken voor mezelf door GitHub beter te gebruiken.

3.2 Timeboxen

Ik denk dat het voor mij heel belangrijk is om goed te kunnen timeboxen. Dit is ook meteen mijn leerdoel voor de komende lessenreeks. Met timeboxen bedoel ik een vaste hoeveelheid tijd toewijzen aan een bepaalde opdracht. Ook wil ik gaan werken met de pomodoro methode en hiermee voorkomen dat ik te lang achter elkaar werk.

3.3 Duidelijk leerdoel stellen en daar op focussen

Dit is voor mij ook een belangrijke takeaway. Ik denk dat ik bij het werken aan de eindopdracht een beetje mijn oorspronkelijke leerdoel uit het oog verloor. Ik bleef hangen in een perfect eindresultaat willen leveren en ik ging soms meerdere keren over al werkende code heen met de gedachte dat ik het nog net iets netter of efficiënter kon maken. Mijn perfectionisme koste veel tijd, en ik had deze tijd denk ik beter kunnen besteden aan het verkrijgen van een nieuwe vaardigheid of aan een ander vak. Daarbij komt ook dat ik niet eens zo veel heb aan het leveren van een perfect eindresultaat omdat ik dan minder heb aan de feedback die ik krijg op mijn werk.

3.4 Hoofdzaak van bijzaak scheiden

Ik weet van mezelf dat ik het soms moeilijk vindt om hoofdzaak van bijzaak te scheiden. Dat heeft waarschijnlijk ook een rol gespeeld in mijn perfectionisme bij de python eindopdracht omdat alles even belangrijk voelde. De hoofdzaak van een opdracht, presentatie of vraag in de les vinden is een belangrijke vaardigheid voor mij om verder te ontwikkelen want veel van de dingen waar ik afgelopen lessenreeks tegenaan ben gelopen staan hiermee in verbinding. Ik wil deze vaardigheid ontwikkelen door voor mezelf een overzicht te maken door het bijvoorbeeld op te schrijven en er bewust bij stil te staan. Als ik er niet zelf uitkom kan ik eventueel een docent of klasgenoot raadplegen voor hulp hierbij.