

Week 4 – Datapunt 1

Naam: Daan Maat

Studentnr: 1213186

Datum: 24/09/2024

Inhoud:

1. Inleiding
2. Leesbaarheid
3. Comments
4. Consistentie
5. Conclusie + advies
6. Bronnenlijst

1. Inleiding

Wanneer je code schrijft is het slim om vanaf het begin al gestructureerd en zorgvuldig te werk te gaan. Dit houdt in dat je comments gebruikt, structuur aanbrengt aan je code door line breaks en whitespace & bovendien efficiënt werkt met je code. Als je code een puinhoop is dat is je applicatie dat waarschijnlijk ook.

2. Leesbaarheid

Leesbaarheid in je code is enorm belangrijk. Als je code leesbaar is kunnen toekomstige developers door bouwen op de applicatie die jij hebt gemaakt. Verder is het voor jezelf ook makkelijk om updates of patches te maken als je alles in je code kunt terugvinden. Je kunt je code leesbaar maken op meerdere manieren.

- 1 van de voornaamste is ervoor zorgen dat je variabele simpele en begrijpbare namen hebben. Variabele met een logische naam maakt ze makkelijker te begrijpen en welke waarde die zou moeten hebben. Een voorbeeld: Een variabele met de naam leeftijd is veel duidelijker dan bijvoorbeeld alleen jaar. Leeftijd weet je wat de programmeur bedoelt en welke value daar moet worden ingevuld.
- Een andere manier om je code leesbaar te maken is om het overzichtelijk te houden. Behoud een zekere structuur in je code. Gebruik genoeg line breaks & whitespace zodat alles duidelijk gestructureerd is. Alles achter elkaar schrijven op dezelfde line geeft alleen maar verwarring (en in mijn geval ook hoofdpijn).

[wikipedia1]

3. Comments

Een andere manier om je code overzichtelijk te houden is door middel van comments. Comments zorgen ervoor dat mensen die geen code kunnen lezen via comments nog kunnen begrijpen wat er gebeurt. Comments zijn stukken tekst die de applicatie niet ziet/gebruikt. Door comments te gebruiken kan je andere developers laten weten waar en wat je code doet. Comments zijn vooral essentieel als je een complexe code hebt die lastig te begrijpen is. Comments moeten niet te uitgebreid zijn. Leg in of 2 zinnen uit wat je code doet. Gewoon 1 woord als comment is meestal niet genoeg om iemand te laten zien waar je een complex stuk code voor gebruikt en hoe die werkt.

4. Consistentie

Bij consistentie is het belangrijk dat jij en je team dezelfde codestijl gebruiken. Indien dit niet het geval is wordt samenwerking vrijwel onmogelijk. Variabelen zouden dezelfde manier van naamgeving moeten hebben bijvoorbeeld, al jullie variabelen zijn in camelCase of snake_case. Hoofdlettergebruik is ook een essentieel onderdeel van consistentie. Als je een variabele wilt aanroepen of een functie maar vergeet een hoofdletter, dan werkt je code meestal niet omdat die over het algemeen hoofdlettergevoelig zijn. Zelf moet ik meer proberen consequent te zijn in mijn hoofdletter gebruik en persoonlijk programmeer ik meer in camelCase maar dat kan veranderen als ik mijn team niet kan overhalen. Als je bijvoorbeeld 2 variabelen in camelCase hebt en 3 in snake_case wordt het enorm onoverzichtelijk.

[PEP8] [wikipedia2]

5. Conclusie + advies

Leesbaarheid, Comments & Consistentie, dit zijn de 3 steunpilaren voor een goed en snel werkende applicatie. Als je aan t werk gaat zorg ervoor dat je bij de variabelen op hoofdlettergebruik let en ook in dezelfde casing de variabelen benoemd. Gebruik je complexe code of if-else statements, zorg en dan voor dat je deze toelicht door middel van een comment waarin je kort uitlegt wat de code doet en de functie ervan. Daarnaast helpt het als jij en je team allemaal dezelfde regels tijdens het schrijven van code hanteren.

6. Bronnenlijst

Bronnen:

[wikipedia1] https://en.wikipedia.org/wiki/Programming_style

[wikipedia2] https://en.wikipedia.org/wiki/Coding_conventions

[PEP8] <https://peps.python.org/pep-0008/>