# Reflectie SuperPy

## Stappenplan:

### Research doen:

* Argparse
* CSV lezen / schrijven / aanpassen
* Basics rich
* Basics numpy
* Basics matlib (charts)
* Basiscs panda (database voor chart)

## Plan van aanpak

Allereerst heb ik een overzicht gemaakt op een whiteboard over hoe het programma zou moeten lopen en welke functies erin moeten zitten. Daarna heb ik globaal alles in geprogrammeerd en bestanden aangemaakt om te kijken of alles met elkaar communiceert. Bestanden:

Bought.csv – slaat een historie op van aankopen  
sold.csv – slaat een historie op van verkopen  
pricelist.csv – slaat producten op die gekocht en verkocht kunnen worden.  
timestatus.txt – slaat de datum op

Om te kijken of alles met elkaar werkt heb ik wat simpele teksten geschreven zodat ik via de command line tool kon lezen en schrijven in CSV en txt bestanden.

Ik kwam er al gauw achter dat ik geen positional arguments wou hebben maar alles optional wou doen. Zo ontstaat er geen verwarring voor de user en de developer. En op deze manier kan het ook willekeurig ingevuld worden. Omdat alles optional is maar soms toch required te maken heb ik wat choices toegevoegd.

Na het hele gebeuren te parsen besloot ik ook om de logica eronder te maken. Zodat hier al veel fouten van de user uitgefilterd worden.

De functies had ik eerst los in een bestand geschreven, maar door ze een beetje te sorteren in classes werd het ook vrij leesbaarder. Ik moest een class hebben die bestanden kon lezen, producten kon managen, en tijd kon managen. Daarna heb ik de functies erin gezet. Eerst had Classes “vrij” geprogrammeerd, maar dat betekende ook dat je bij iedere andere functie best veel arguments moet geven, en op een gegeven moment werd het te moeilijk. Ik besloot om voorgedefinieerde functies te maken in de class zodat je die functies kan oproepen.

Toen alles werkte begon ik aan het creëren van de charts.

## Take aways en leermomenten

* Het is overzichtelijker en makkelijker om de code te categoriseren in verschillende bestanden.
* GitHub redt levens
* Tijdens het testen ook controleren op type() en niet alleen uitprinten wat er terug komt.   
  Veel errors komen door het feit dat de type niet klopt.
* Er zijn veel integraties die mooie dingen kunnen doen.

## Minimaal 2 modules gebruiken in code

1. Ik had eerst gekozen voor rich. Gewoon omdat dit de opmaak wat opfrist een zo’n saaie terminal. Rich is makkelijk en doet me wat een HTML denken.
2. Numpy al gebruik ik er slechts een paar lines of code van voor de Charts
3. Pandas ook hier gebruik ik een database functie van voor de Charts
4. Matploblib voor het genereren van de charts en een easter egg.
5. Om tables te maken nog even snel tabulate toegevoegd.