

# Python Data Model

Bas Terwijn

February 9, 2026

## 1 Introductie

Dit document behandelt het Python Data Model.

### 1.1 Doelen

Begrijpen van:

- references
- immutable en mutable data types
- assignment, shallow en deep copy
- name rebinding en mutatie

## 2 Python Data Model

Het [Python Data Model](#) beschrijft hoe Python omgaat met data. Het is belangrijk om dit goed te begrijpen om effectief met Python te kunnen werken. Kijk daarom deze [uitlegvideo over mutability](#), installeer de `memory_graph` package met deze [install-instructie](#) en maak de `band_schedule.py` opdracht die aan het einde wordt besproken.

Lees ook de [Python Data Model uitleg](#) t/m de [Data Model Exercises](#) over deze concepten. De exercises in [Data Model Exercises](#) lijken in eerste instantie misschien lastig, maar bij het consistent toepassen van de uitgelegde concepten kom je tot het juiste antwoord:

1. [immutable](#) vs. [mutable](#) data types
2. [assignment](#) vs. [shallow](#) vs. [deep copy](#)
3. [name rebinding](#) vs. mutatie

Deze concepten zijn belangrijk om bugs te voorkomen, daarom wil je deze altijd in je achterhoofd hebben wanneer je in Python aan het programmeren bent. Op het tentamen komen dan ook zeker dit soort exercises voor, dus bestudeer deze concepten goed en oefen met de exercises!

### 2.1 Opdracht `band_schedule.py`

Zet je uitwerking van de `band_schedule.py` opdracht aan het einde van de [uitlegvideo over mutability](#) in de file `assignments/band_schedule.py`. Gebruik, zoals ook al uitgelegd in deze video, de tests in file `band_schedule_test.py` in deze directory om je uitwerking te testen:

```
$ python band_schedule_test.py
```

## 2.2 Opdracht products.py

In programma `products.py` in de `exercises` directory wordt een lijst van producten aangemaakt en geprint. In de functies `assignment1()` t/m `assignment4()` wordt u gevraagd enkele toevoegingen en aanpassingen te maken aan deze lijst. Gebruik hierbij de `memory_graph` package om een goed begrip te houden van de data zoals weergegeven in figuur 1.

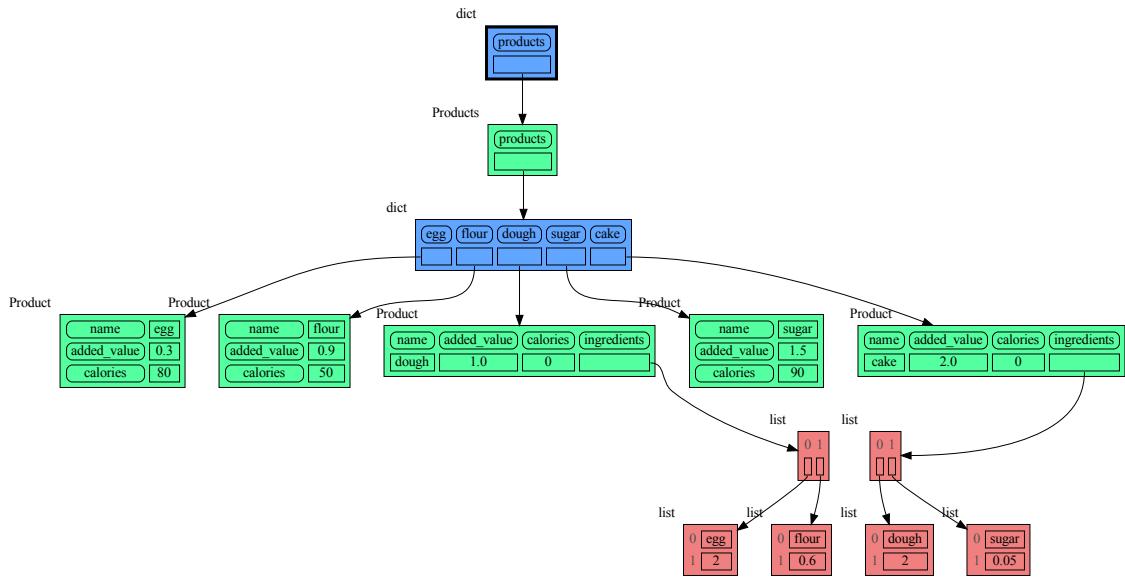


Figure 1: `memory_graph` van `products.py` vooraf aan `assignment1`

Test deze opdracht met:

```
pytest test_products.py
```