

# LU04b - Konstruktor in Dataclass



In einer Dataclass wird der Konstruktor anhand der Attribute automatisch generiert. Falls Sie eigene Zusicherungen im

## Konstruktor

In einer `@dataclass` definieren wir die Attribute mit ihren Datentypen. Python erzeugt automatisch einen Konstruktor mit allen Parametern und Wertzuweisungen.

## Beispiel

```
@dataclass
class Member:
    givenname: str
    birth_year: int
    membership_fee: float
```

Die oben definierte Klasse erzeugt automatisch diesen Konstruktor:

```
...
def __init__(self, givenname: str, birth_year: int, membership_fee:
float):
    self.givenname = givenname
    self.birth_year = birth_year
    self.membership_fee = membership_fee
```

## Zusicherungen und Programmlogik

Manche Klassen verlangen nach expliziten Zusicherungen oder Logik beim Erzeugen eines Objekts. Anstatt den Konstruktor von Hand zu schreiben, nutzen wir dafür die Methode `post_init()`. Diese Methode, sofern vorhanden, wird am Ende des Konstruktors aufgerufen.

## Beispiel

In unserem Beispiel wollen wir sicherstellen, dass der Mitgliederbeitrag nicht negativ sein kann.

```
...
def __post_init__(self):
    if self.membership_fee < 0:
```

Last update: 2024/08/14 06:52 modul:m320\_2024:learningunits:lu04:dataclass [https://wiki.bzz.ch/modul/m320\\_2024/learningunits/lu04/dataclass?rev=1723611172](https://wiki.bzz.ch/modul/m320_2024/learningunits/lu04/dataclass?rev=1723611172)

```
raise ValueError('membership_fee may not be a negative value')
```

## M320-LU04



René Probst, bearbeitet durch Marcel Suter

From:  
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:  
[https://wiki.bzz.ch/modul/m320\\_2024/learningunits/lu04/dataclass?rev=1723611172](https://wiki.bzz.ch/modul/m320_2024/learningunits/lu04/dataclass?rev=1723611172)

Last update: **2024/08/14 06:52**

