

```
1 class Fahrzeug():
2     def __init__(self, geschwindigkeit):
3         self.geschwindigkeit = geschwindigkeit
4     def fahren(self):
5         print("Das Fahrzeug fährt.")
6 class Auto(Fahrzeug):
7     def __init__(self, geschwindigkeit, farbe):
8         super().__init__(geschwindigkeit)
9         self.farbe = farbe
10    def hupen(self):
11        print("Das Auto hupt.")
12 auto1 = Auto(100, "rot")
13 auto1.hupen()    #Ausgabe: Das Auto hupt.
14 auto1.fahren()  #Ausgabe: Das Fahrzeug fährt.
```



Vererbung in Python

Um eine Vererbungsrelation in Python anzugeben, wird der Name der Superklasse bei der Definition der Subklasse in Klammern geschrieben. Im obigen Beispiel erbt also die Klasse Auto von der Klasse Fahrzeug.

Objekte der Subklasse können auf Attribute und Methoden zugreifen, die in der Superklasse definiert wurden. Beispielsweise kann `auto1.fahren()` aufgerufen werden, obwohl die Methode `fahren()` in der Klasse Fahrzeug definiert wurde.



Aufruf von Methoden der Superklasse

Mit `super()` können Methoden der Superklasse aufgerufen werden. Dies ist insbesondere bei den Konstruktoren (`__init__()`-Methoden) hilfreich, um doppelten Code zu sparen.

In Zeile 8 müsste sonst das Attribut `geschwindigkeit` manuell gesetzt werden. Bei vielen Parametern wäre das sehr aufwändig.