

# Arbeitsauftrag zum Einstieg in die Objektorientierte Programmierung (OOP) mit Python



## INPUT



Wenn Du Dich noch nicht mit Objektorientierung beschäftigt hast, sieh Dir zur Einführung das Video von [Programmieren starten](#) an, oder lies den Text unter dem Video durch.

Bearbeite auch den [Arbeitsauftrag auf dem Hessischen Bildungsserver](#).



## Klassen in Python

Wir bleiben bei dem Beispiel mit den Haustieren. Zuerst definieren wir die Klasse HAUSTIER, von der später die Klassen HUND und KATZE erben.

Das Haustier soll die Attribute Name, Art, Farbe, Rasse und Tierarzt haben.

Die Klasse wird wie folgt definiert:

```
#Klasse Haustier
class Haustier:
    #Für die Initialisierung wird die Standardfunktion __init__ verwendet
    #(Beachte: 2 Unterstriche vor und nach init)
    #Der erste Parameter ist immer self für das Objekt selbst,
    #danach folgen die Attribute, die hier mit None vorbelegt sind.
    #Die Vorbelegung mit None bedeutet, dass die Angabe des jeweiligen
    #Attributes bei der Erzeugung einer Instanz optional ist.
    def __init__(self, name = None, art = None, farbe = None, rasse = None, tierarzt = None):
        self.name = name           #die Zuweisung der Parameter zum erzeugten Objekt
        self.art = art
        self.farbe = farbe
        self.rasse = rasse
        self.tierarzt = tierarzt
```

Das Haustier bekommt jetzt die Methode frisst. Dabei soll einfach übergeben werden, ob das Haustier frisst, oder nicht.

```
def frisst(self, wahr = True):
    if wahr:
        print("Ich fresse gerade.")
    else:
        print("Ich fresse gerade nicht.")
```

Nun kann ein Objekt der Klasse Haustier erzeugt werden und die Funktion kann mittels Punktnotation aufgerufen werden. Das Ergebnis wird unterhalb angezeigt:

```
22 #Ein Objekt wird erzeugt
23 miezi = Haustier(name = "Miezi", farbe = "grau", art = "Katze")
24 #Die Art wird ausgegeben
25 print("Ich bin ein(e) " + miezi.art)
26 #Die Funktion frisst wird mittels Punktnotation aufgerufen
27 miezi.frisst()
28
```

```
Ich bin ein(e) Katze
Ich fresse gerade.
```



## ARBEITSAUFTRAG 1

Für die Programmierung kannst Du, sofern Du keinen Python-Interpreter installiert hast, mit [WebTigerJython](#) arbeiten. Du kannst online programmieren und den Code ausführen.

- Übernimm den Code für die Klasse *Haustier* und die Methode *frisst*.
- Ergänze die Klasse um die Methode *schlaeft*.
  - Erzeuge ein Objekt mit dem *Namen Garfield* und der *Art Kater*.
  - Erzeuge ein Objekt mit dem *Namen Waldi*, der *Art Hund* und der *Rasse Dackel*.
  - Gib Namen und Art des Hundes wie folgt aus: „Ich bin ein(e) *Art* , genauer gesagt ein(e) *Rasse* und heiÙe *Name*.“
  - Lege den Hund schlafen.
  - Gib Namen und Art der Katze wie folgt aus: „Ich bin ein(e) *Art* und heiÙe *Name*.“
  - Lass die Katze fressen.
  - Gib den Namen des Hundes wie folgt aus: „Hier ist wieder *Name*“
  - Wecke den Hund wieder auf.



## Kind-Klassen in Python

Nun haben Katzen und Hunde unterschiedliche Fähigkeiten (Methoden) und vielleicht auch Eigenschaften (Attribute).

Daher definieren wir nun die Klassen *Katze* und *Hund* und ergänzen sie um eine jeweils individuelle Fähigkeit (Methode). Die Katze bekommt die Methode *miauen* und der Hund bekommt die Methode *bellen*. Wir können jetzt mit der Erzeugung des Objektes auch die Art festlegen.

Hier das Beispiel für den Hund:

```
#die Klasse Hund ist eine Kindklasse von Haustier
class Hund(Haustier):
    #Hund hat hier dieselben Parameter wie Haustier
    #aber die Art ist hier mit Hund vorbesetzt und wird an die Elternklasse übergeben
    def __init__(self, name = None, art = "Hund", farbe = None, rasse = None, tierarzt = None):
        Haustier.__init__(self, name = name, art = art, farbe = farbe, rasse = rasse, tierarzt = tierarzt)
    #der Hund hat die spezifische Fähigkeit bellen
    def bellt(self):
        print("Wuff-Wuff")
```



## ARBEITSAUFTRAG 2

- Übernimm die Klasse *Hund* in Deine Datei.
- Ergänze entsprechend die Kindklasse *Katze* und erweitere sie um die Methode *miaut*.
- Erzeuge jeweils ein Objekt der Klasse *Hund* und der Klasse *Katze*. Übergib den Namen des Tieres.
- Lasse den Hund bellen und die Katze miauen.



### Arbeitsauftrag 3

- Modelliere nun eine Klasse Fahrzeug.
- Welche Eigenschaften und Fähigkeiten haben verschiedenen Fahrzeuge gemeinsam?
- Modelliere die Kindklassen Auto und Motorrad und Fahrrad mit ihren spezifischen Attributen und Methoden.
- Implementiere die Klassen in Python.
- Erzeuge von jeder Klasse ein Objekt und lasse die Objekte z. B. beschleunigen, bremsen, tanken, Reifendruck prüfen ..... .