



Einführung - Klassen und Objekte in Python

Einführung

In kleinen Programmen reicht es oft aus, einfache **Funktionen** und **Variablen** zu verwenden. Aber **sobald Programme größer und komplexer** werden, kann es **schwierig** werden, den **Überblick** zu behalten.

Ohne eine klare Struktur könnten Daten und Funktionen verstreut und schwer wiederzufinden sein.

Klassen bieten eine Möglichkeit, Daten (Attribute) und Verhalten (Methoden) zu gruppieren, die logisch zusammengehören. Dadurch wird der Code modular und leicht zu verstehen.

Objekte sind spezifische Instanzen dieser Klassen, **die individuelle Zustände** und Verhalten repräsentieren. Dies erleichtert es, mit mehreren Entitäten gleichzeitig zu arbeiten.



Einführung - Klassen und Objekte in Python

Warum brauchen wir Klassen und Objekte?

- Organisation und Strukturierung des Codes:
Erleichtert das Verständnis und die Pflege des Programms.
- Wiederverwendbarkeit:
Reduziert den Aufwand und erhöht die Effizienz durch das Wiederverwenden von Code.
- Erleichterung der Wartung und Erweiterung:
Macht den Code flexibel und einfacher zu erweitern.
- Modellierung von realen Objekten:
Bietet eine natürliche Möglichkeit, komplexe Systeme darzustellen.
Klassen bieten eine natürliche Möglichkeit, reale Objekte und Konzepte zu modellieren. Sie können z. B. Klassen für Kunden, Produkte, Bestellungen usw. erstellen, die reale Entitäten und deren Verhalten im Programm abbilden.
- Förderung guter Programmierpraktiken:
Unterstützt die Einhaltung bewährter Entwicklungsprinzipien.
Klassen und Objekte fördern die Einhaltung der Prinzipien der Softwareentwicklung, wie DRY (Don't Repeat Yourself), Kapselung und Modularität.



Einführung - Klassen und Objekte in Python

Was sind Klassen und Objekte?

Definition einer Klasse:

Eine Klasse ist ein Bauplan oder eine Vorlage, die beschreibt, wie Objekte einer bestimmten Art strukturiert sind.

Sie legt fest, welche Attribute (Daten) die Objekte haben und welche Methoden (Funktionen) sie ausführen können.

Definition eines Objekts:

Ein Objekt ist eine konkrete Umsetzung einer Klasse. Es wird im Arbeitsspeicher (RAM), genauer gesagt im sogenannten [Heap](#), gespeichert, sobald es erstellt wird. Dort bleibt es so lange bestehen, bis es vom Programm nicht mehr verwendet wird – dann wird es automatisch durch die sogenannte [Garbage Collection](#) gelöscht. Das Objekt speichert bestimmte Werte für die Eigenschaften (Attribute) und kann die in der Klasse definierten Funktionen (Methoden) ausführen.

Real Life Beispiel :

Klasse als **Bauplan**: Stellen Sie sich eine Klasse wie einen **Bauplan für ein Haus** vor. Der Bauplan selbst beschreibt, wie das Haus aussehen soll, wie viele Zimmer es hat, welche Materialien verwendet werden, etc. Er ist jedoch nur eine theoretische Beschreibung und kein tatsächliches Haus.

Objekt als **reales Exemplar**: Ein Objekt ist das **fertige Haus**, das nach diesem Bauplan gebaut wurde. Es ist eine konkrete, greifbare **Umsetzung** des Bauplans. **Während der Bauplan für viele Häuser verwendet werden kann, ist jedes Haus, das nach diesem Plan gebaut wird, ein eigenes Objekt.**



Einführung - Klassen und Objekte in Python

Aufbau einer Klasse

```
class NameDerKlasse:
```

```
    def __init__(self, parameter1, parameter2):
```

```
        self.attribute1 = parameter1
```

```
        self.attribute2 = parameter2
```

class-Schlüsselwort: Leitet die Definition einer Klasse ein.

__init__ Methode: Auch als **Konstruktor** bekannt, wird **automatisch** aufgerufen, wenn ein neues Objekt erstellt wird.

self: Eine Referenz auf das aktuelle Objekt der Klasse. Erlaubt den Zugriff auf die Attribute und Methoden der Klasse.

self. **attribute1** : **parameter1** ist der Wert, der beim **Erstellen des Objekts** übergeben wird, und self. **attribute1** speichert diesen Wert als **Attribut** des Objekts, sodass er später innerhalb der Klasse verwendet werden kann.



Einführung - Klassen und Objekte in Python

Aufbau einer Klasse

```
class Auto:  
    def __init__(self, marke, modell, baujahr):  
        self.marke = marke  
        self.modell = modell  
        self.baujahr = baujahr
```

Hier definiert die Klasse `Auto` drei Attribute: `marke`, `modell`, und `baujahr`.

Erstellung und Verwendung von Objekten

Ein `Objekt` wird durch den Aufruf der Klasse mit den erforderlichen Argumenten erstellt.

```
objekt_name = KlassenName(argument1, argument2)
```

Beispiel:

```
mein_auto = Auto("Toyota", "Corolla", 2020)
```

Das Objekt `mein_auto` ist eine Instanz der Klasse `Auto` mit den Attributwerten „Toyota“, „Corolla“, und „2020“



Einführung - Klassen und Objekte in Python

Erstellung und Verwendung von **Objekten**

Sie können auf die Attribute eines **Objekts** zugreifen, indem Sie die Punktnotation verwenden:

```
print(mein_auto.marke) # Ausgabe: Toyota
```

```
print(mein_auto.baujahr) # Ausgabe: 2020
```



Einführung - Klassen und Objekte in Python

Methoden und Attribute in Klassen

Attribute:

Attribute sind Variablen, die den Zustand eines Objekts beschreiben. Sie werden innerhalb der Klasse definiert und mit dem Schlüsselwort `self` referenziert.

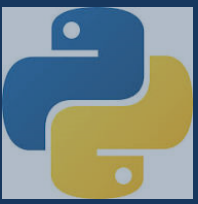
Methoden:

Methoden sind Funktionen, die in einer Klasse definiert sind und das Verhalten der Objekte festlegen.

Beispiel: Eine Methode `bellen` in der Klasse `Hund`, die den Text „bellt“ ausgibt.

```
class Hund:
    def __init__(self, name, alter):
        self.name = name
        self.alter = alter
    def bellen(self):
        print(f"{self.name} bellt!")

mein_hund = Hund("Bello", 3)
mein_hund.bellen() # Ausgabe: Bello bellt!
```



Einführung - Klassen und Objekte in Python

Zusammenfassung

- Klassen sind Baupläne für Objekte, die Attribute und Methoden definieren.
- Objekte sind konkrete Instanzen dieser Klassen mit spezifischen Attributwerten.
- OOP in Python ermöglicht es, Code modular, wiederverwendbar und übersichtlich zu gestalten.
- Die Kenntnis von Klassen und Objekten ist essenziell für das Schreiben von strukturiertem und wartbarem Code in Python.
- Mit Praxisbeispielen kann das Verständnis dieser Konzepte vertieft werden.



Einführung - Klassen und Objekte in Python

Mehr erkunden:

https://www.w3schools.com/python/python_classes.asp

<https://docs.python.org/3/tutorial/classes.html>

<https://www.geeksforgeeks.org/python-classes-and-objects/>