

Konstruktor – die Methode `__init__`

Als **Konstruktor** wird die Methode einer Klasse bezeichnet, die automatisch beim Erzeugen (Instanzieren) eines Objekts der Klasse aktiviert wird. In Python ist der Methodenbezeichner dafür vorgegeben: `__init__`. `init` steht dabei für das Wort Initialisierung, da diese Methode bei der Objekterzeugung automatisch aktiviert wird. Wie jede andere Methode kann auch der Konstruktor `__init__` mit formalen Parametern versehen werden und mit Attributen und Attributwerten arbeiten.

Allgemeiner Konstruktor

Zur Wiederholung der Begrifflichkeiten folgt der beispielhafte Aufbau eines Konstruktors in Python:

```
1 def __init__(self, formalerParam1, formalerParam2, formalerParam3):
2     self.attribut1= formalerParam1
3     self.attribut2= formalerParam2
4     self.attribut3= formalerParam3
5     self.attribut4= "Ein_beliebiger_Text"
6     self.attribut5= 361
```

Konstruktor der Klasse Spiel

Im konkreten Beispiel unseres Sportturniers sieht der Konstruktor der Klasse `Spiel` wie folgt aus:

```
1 def __init__(self, start1, ende1, halle1, mannschaft1, mannschaft2):
2     self.beginn= start1
3     self.ende= ende1
4     self.halle= halle1
5     self.mannschaftA= mannschaft1
6     self.mannschaftB= mannschaft2
7     self.punkteA= -1
8     self.punkteB= -1
```

Soll das Objekt `meinSpiel` aus der Klasse `Spiel` erzeugt werden, so muss man nachsehen, welche und wieviele Parameter benötigt werden:

```
! meinSpiel= Spiel("9:30", "10:00", halleAltbau, dieKaengurus, crazyFrogs)
```

Es wird deutlich, dass die Reihenfolge der aktuellen Parameterwerte mit der Reihenfolge der `init`-Methode übereinstimmen muss.

