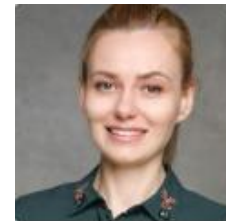


# Anwendungsfall 2: Korrekte Anzahl an Argumenten



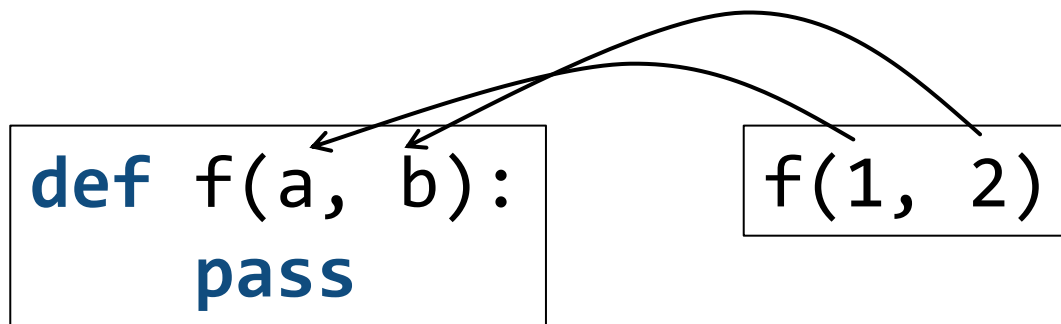
Bernhard Stadler



Klaudia Thellmann

# Die korrekte Anzahl Argumente

- In Python können Argumente bei Funktionsaufrufen durch ihre Position auf Parameter abgebildet werden



- Anzahl Argumente beim Funktionsaufruf identisch zur Anzahl Parameter bei Funktionsdefinition

## Beispiel in scikit-learn

```
from sklearn.base import is_classifier
is_classifier(model)      # OK
is_classifier()           # Error (no args instead of one)
```

## Beispiel in PyTorch

```
from torch.nn import Identity
model = Identity()       # OK
model = Identity(1)     # Error (one arg instead of none)
```

## User Story

*Als Python-Entwickler möchten wir schon während der Programmentwicklung eine Fehlermeldung bekommen, wenn wir bei einem Aufruf einer Bibliotheksfunktion mehr oder weniger Argumente nutzen, als in der Funktionsdefinition vorgesehen.*

*Dadurch können wir Zeit sparen, weil wir das Programm nicht erst ausführen müssen, um diesen Fehler zu entdecken.*

*Hinweis: In künftigen Iterationen werden wir weitere Überprüfungen anfordern. Uns ist wichtig, dass diese Überprüfungen später mit geringem Aufwand in verschiedene IDEs und Code-Editoren eingebaut werden können und überall die gleichen Meldungen liefern.*