

private Attribute

Wozu
private Attribute
?

private Attribute

Die Aufgabe zum Ändern der Farbe hat gezeigt:

- Bei einem Zugriff auf Attribute sollte gewährleistet sein, dass der Attributwert und seine Repräsentationen konsistent sind.

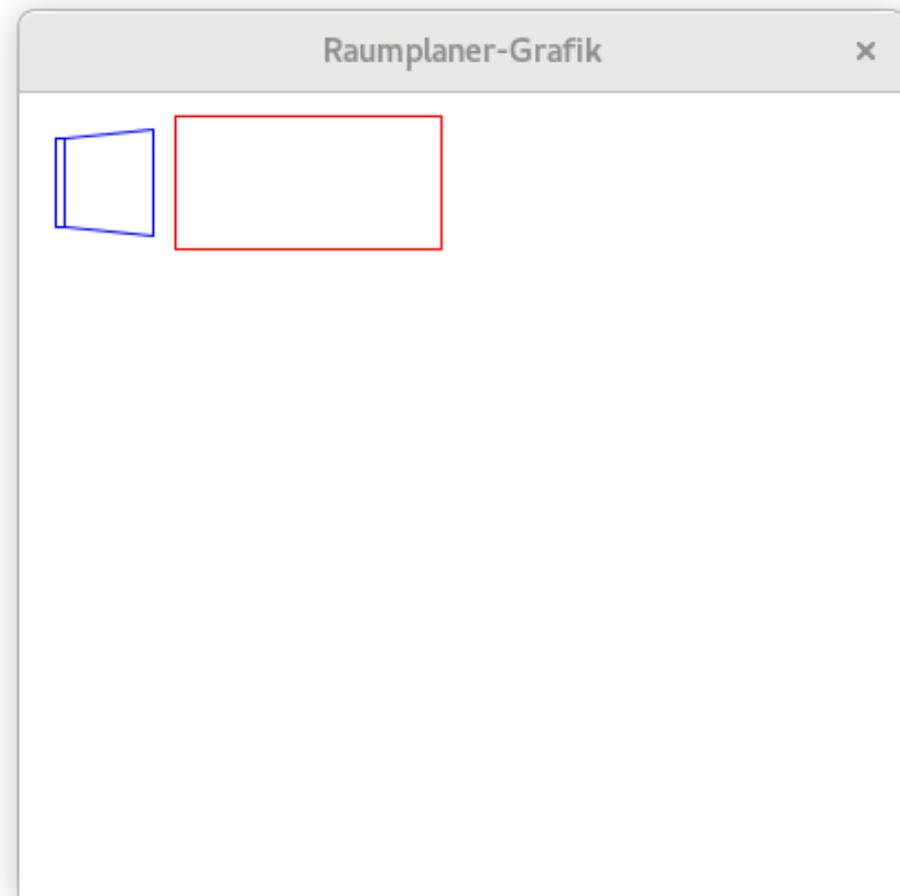
In unserem Projekt heißt das konkret:

- Ändert man Attributwerte direkt, kann es dazu kommen, dass der Zustand des Objekts nicht mit seiner Darstellung übereinstimmt.

private Attribute

```
PyShell
File Edit View Options Help
PyShell 0.9.8 - The Flakiest Python Shell
Python 3.6.10 (default, Jan 16 2020, 09:12:04) [GCC] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> stuhl.f="brown"
>>> stuhl.GibFarbe()
'brown'
>>> stuhl.f
'brown'
>>>
```

PyShell 0.9.8 - The Flakiest Python Shell



private Attribute

- Dass der Attributwert richtig ist und richtig verarbeitet wird, kann man sehen, wenn man das Grafikfenster leicht verschiebt.
- Dann wird das Fenster neu gezeichnet und das Stuhlsymbol ist nun braun.
- Denselben Effekt erzielt man, wenn man die Methode Zeige aufruft:
stuhl.Zeige()

private Attribute

- Um einen direkten Zugriff auf Attributwerte zu verhindern, ändert man ihre Sichtbarkeit auf ***private***.
- Dafür gibt es zwar in der Sprache Java einen Modifizierer, der vor die Typdeklaration gesetzt wird. Das gibt es bei Python jedoch nicht.
- Man kann bei Python aber den Attributnamen so ändern, dass ein direkter Zugriff nicht mehr möglich ist.

private Attribute

- Dazu setzt man vor die Attributnamen nach dem Punkt zwei Unterstreichungsstriche.
- Für das Beispiel `stuhl.x`:

```
def __init__(self, sichtbar=False):  
    """einfacher Konstruktor"""  
    self.__x=20  
    self.y=20  
    self.b=40  
    self.t=40  
    self.w=270  
    self.f="blue"  
    self.s=sichtbar  
    if sichtbar: self.Zeige()
```

private Attribute

- Konsequenterweise sollte man das für alle Attributnamen machen.

```
def __init__(self, sichtbar=False):  
    """einfacher Konstruktor"""  
    self.__x=20  
    self.__y=20  
    self.__b=40  
    self.__t=40  
    self.__w=270  
    self.__f="blue"  
    self.__s=sichtbar  
    if sichtbar: self.Zeige()
```

private Attribute

- Selbstverständlich sollte man die *Suchen und Ersetzen* – Funktion des Editors nutzen um die Änderungen an allen Positionen der Dateien durchzuführen.
- Das bleibt aber dennoch nicht folgenlos, da die Attributwerte **self.__x**, **self.__y**, **self.__w** und **self.__s** nun zwar immer noch veränderbar sind, die für Farbe, Breite und Tiefe aber nicht.

private Attribute

- Das mag bei einem Stuhl bei der Breite und Tiefe möglicherweise noch akzeptabel sein.
- Tischobjekte sollten aber unterschiedliche Breite und Tiefe haben dürfen.
- Unterschiedliche Farben darzustellen wird man bei allen Symbole wollen.

private Attribute

- Konsequenterweise werden für alle Attribute verändernde Methoden bereit gestellt, die gewährleisten, dass Attributwert und Darstellung übereinstimmen.

AendereFarbe(self, neueFarbe)

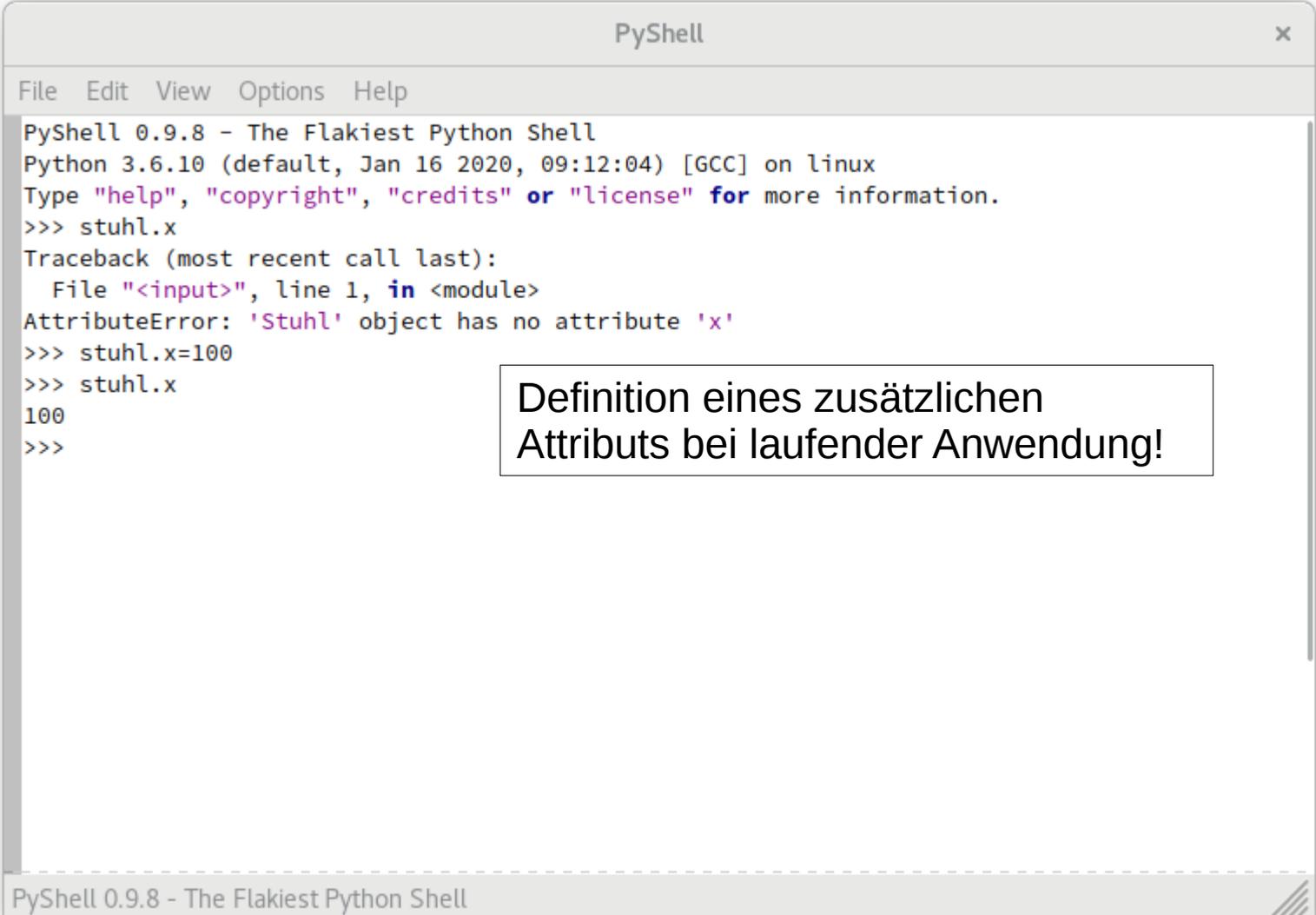
AendereBreite(self, neueBreite)

AendereTiefe(self, neueTiefe)

sorgen dann jeweils zusätzlich zum Ändern des Attributwerts dafür, das Objekt neu darzustellen.

private Attribute

- Vorsicht bei Python!



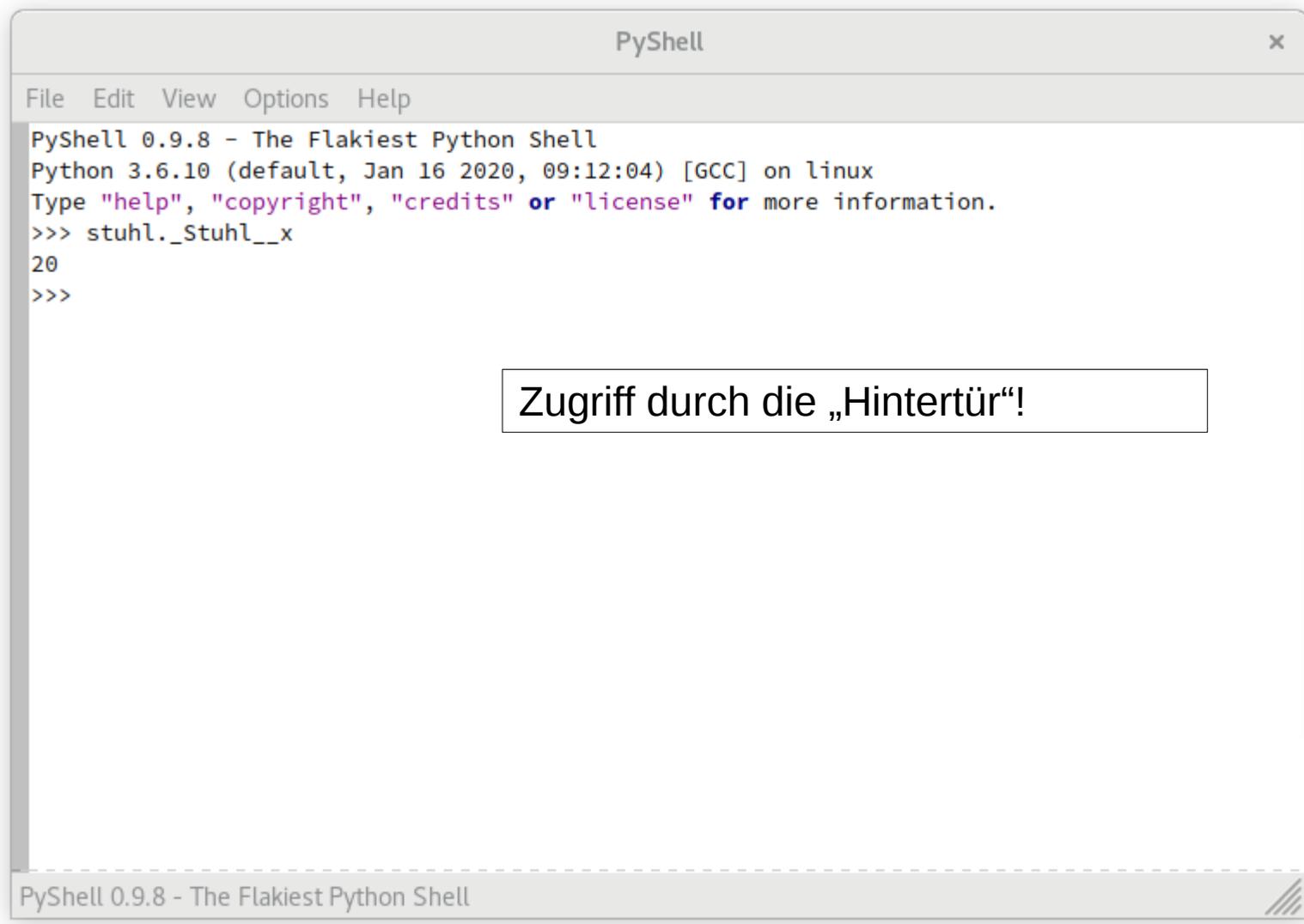
```
PyShell
File Edit View Options Help
PyShell 0.9.8 - The Flakiest Python Shell
Python 3.6.10 (default, Jan 16 2020, 09:12:04) [GCC] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> stuhl.x
Traceback (most recent call last):
  File "<input>", line 1, in <module>
AttributeError: 'Stuhl' object has no attribute 'x'
>>> stuhl.x=100
>>> stuhl.x
100
>>>
```

Definition eines zusätzlichen Attributs bei laufender Anwendung!

PyShell 0.9.8 - The Flakiest Python Shell

private Attribute

- Vorsicht bei Python!



```
PyShell
File Edit View Options Help
PyShell 0.9.8 - The Flakiest Python Shell
Python 3.6.10 (default, Jan 16 2020, 09:12:04) [GCC] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> stuhl._Stuhl__x
20
>>>
```

Zugriff durch die „Hintertür“!

PyShell 0.9.8 - The Flakiest Python Shell