

FS24 CAS PML - Python

6. Python Syntax Regeln

Syntax Regeln: Anweisungen

- ▶ Eine Anweisung endet am Ende einer Zeile ausser wenn:
 - ▶ ein dreifach quotierter string offen ist
 - ▶ ein Klammerpaar nicht geschlossen wurde
 - ▶ die Zeile mit einem backslash endet
- ▶ Mehrere Anweisungen auf einer Zeile werden mit Semikola getrennt

```
s = """multiline  
string"""  
  
a = (1, 2,  
     3, 4)  
  
b = 3 * \  
    2
```

```
a = 2 ; b = 3
```

Syntax Regeln: Namen

▶ Variablenamen:

- ▶ beginnen mit einem Buchstaben oder Unterstrich
- ▶ Enthalten nur Buchstaben, Unterstriche und Ziffern
- ▶ dürfen keinem reserviertem Wort entsprechen:

```
total = 10
Total = 15
_1 = "blabla"
true = False
```

- | | |
|-------------------------------|--------------------------|
| ▶ if, elif, else | ▶ def, class, lambda |
| ▶ True, False, None | ▶ return, yield, pass |
| ▶ and, or, not | ▶ from, import, as, with |
| ▶ for, while, continue, break | ▶ is, in, assert |
| ▶ try, except, finally, raise | ▶ del, global, nonlocal |

Syntax Regeln: Namen Konvention

- ▶ Built-in Funktionen nicht überschreiben
- ▶ Explizite Namen auswählen
- ▶ Namen die mit “_” beginnen werden als “Privat” angesehen
- ▶ Namen die mit “__” beginnen haben spezielle Bedeutungen für den Interpreter
- ▶ Klassennamen beginnen oft mit einem Grossbuchstaben

```
In [1]: print = min
...: print([1,2,3])
...:
Out[1]: 1

In [2]: min.__str__()
Out[2]: '<built-in function min>'

In [3]: str(min)
Out[3]: '<built-in function min>'

In [4]: class MyClass():
...:     pass
...:
```

Syntax Regeln

- ▶ Code-Blocks werden durch Einrückung definiert

```
[In [2]: message="i{0}{1}"
...: for i in range(3):
...:     if i==1:
...:         operator="="
...:     else:
...:         operator="!="
...:     print(message.format(operator,i))
...:
...:
i!=0
i=1
i!=2
```