

1. Python Grundlagen

1 PYTHON ALS PROGRAMMIERSPRACHE

- **Interpretierte Sprache:** Zeile für Zeile wird ausgeführt, keine executable wird erzeugt
- **Dynamische Sprache:** Neudefinierung alter Variablennamen ist möglich
- **"type inferred" Sprache:** Datentypen werden implizit angenommen, wenn nichts deklariert wird
- **Einrückungen/Indentation:** Teil der Syntax, teilen Blöcke (zusammengehörige Statements)
- **Case Sensitive:** Groß- und Kleinschreibung muss beachtet werden

2 SYNTAX

- **Comments:** (nur eine Zeile), " oder ''' für Comments über mehrere Zeilen
- **Statements:** werden durch den Newlinecharakter voneinander getrennt (enter)
- **Operatoren** (für Zahlen):
+, -, *, /,
// (gibt den gerundeten Wert zurück), % (Modulo, gibt also nur den Rest wieder),
** (10**2 ist also 10²)
- **Assignment Operator:** "=": Namen links davon wird Wert rechts davon zugewiesen
- **Comparison Operator:** "==": vergleicht den Wert links mit dem Wert rechts (Ergebnis: boolean)
- Implizite Datentypkonvertierung/Erkennung: Python nimmt einen Datentyp an, wenn er nicht explizit angegeben ist und versucht Datenverlust bei Operationen zu vermeiden (bei der Division von zwei Integers wird eine floating point number erzeugt (Python 3))

2.1. Schleifen

- **for [item] in [object]:**, z.B. for counter in range(5):
- **while [condition == True]:** z.B. while i<5:

2.2. Bedingung

- **if** [condition == True]: [do something]
- **elif** [condition2 == True]: [do something else]
- **else:** [do something else]

3 VARIABLEN

3.1. Was sind Variablen?

Namen, die als Platzhalter verwendet werden für eine Speicheradresse. Variablenassignment: einem Speicherplatz wird ein Name gegeben und ein Wert zugewiesen.

3.2. Benennung von Variablen:

Make it meaningful!

- keine Zahlen am Beginn
- keine reservierten Wörter (Keywords oder Built-in-Funktionen)
- kein _ oder __ am Beginn

4 DATENTYPEN

- **numerische Typen:** integer (int), floating point (float), complex (imaginär wird als "j" bezeichnet)
- **strings:** Sequenz von Unicode Characters (also "Buchstaben")
- **Container:** Listen (list), Dictionaries (dict), tuples, sets
- **logischer Datentyp:** Boolean (0 == False, 1 == True)
- **None:** Keyword um Nichtexistenz eines Objectes zu deklarieren

5 LISTEN

Eine Liste ist ein Container mit Ordnung, initialisiert mit "myList = []" bzw. "myList = [1, 2, 3]"

- Inhalt kann verändert werden (z.B. in einer Schleife)
- Variablen können "per Position" aufgerufen werden: Indexierung: myList[2] = 3
- Index beginnt ab 0 zu zählen! (nicht 1)
- Zugriff auf letzten Eintrag: myList[-1]
- Länge der Liste: len(myList) = 3

6 MÖGLICHE FEHLERMELDUNGEN

- **NameError:** Variable wurde benutzt bevor sie deklariert wurde (Achtung vor Tippfehlern!)
- **TypeError:** z.B. Operationen wurde ausgeführt, die nicht zu dem Datentyp passen (Addition von strings und integers)
- **SyntaxError:** Falsch benutztes Keyword, nicht abgeschlossenes Statement (z.B. fehlender Doppelpunkt nach Schleife, ...)
- **ValueError:** Variable kann nicht von string in float konvertiert werden, sie enthält illegale Werte (z.B. tatsächlich Buchstaben)
- **IndexError:** Index ist nicht verfügbar (passiert z.B. wenn die Länge der Liste als Indexrange verwendet wird)