Programmierung mit Python

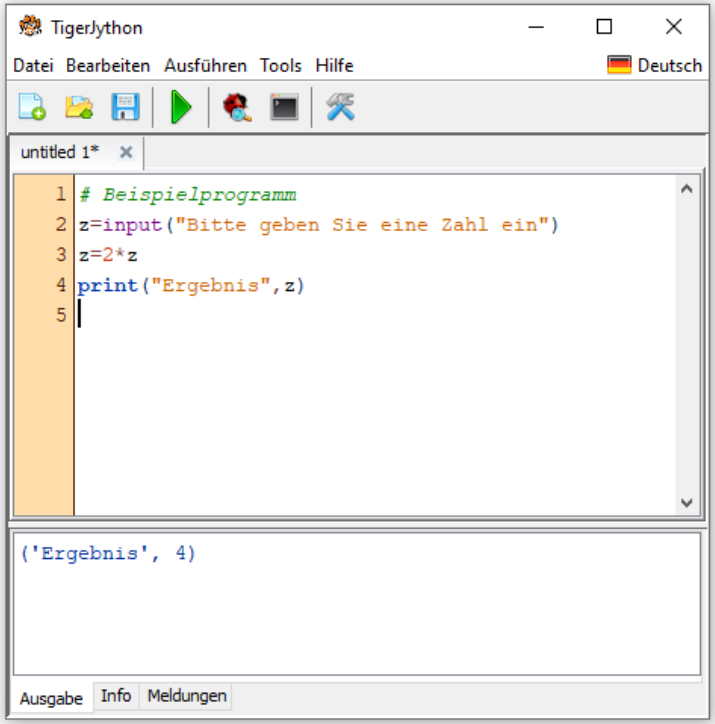
Quellcode

Der Quellcode besteht aus reinen Textzeilen (ASCII-Code) und kann in einem beliebigen Texteditor geschrieben werden.

Mit speziellen Editoren können Fehler schneller erkannt und korrigiert werden.

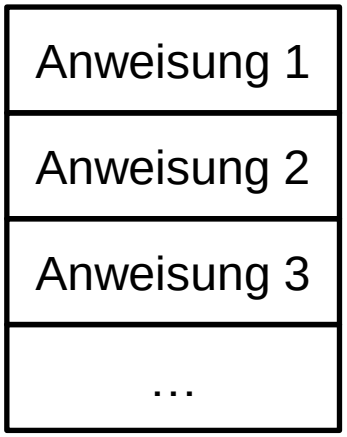
→ *Syntax Highlighting*

- jede Anweisung steht auf einer neuen Textzeile.
- mit # … können (sollten) einzeilige Kommentare eingebunden werden.
- mehrzeilige Kommentare werden mit “““ (dreifachen Anführungszeichen) begonnen und beendet
- die Speicherung des Quellcodes erfolgt mit der Dateierweiterung *.py
- zur Ausführung muss der Quellcode übersetzt werden (▶)



(1) Sequentielle Programme

Sequentielle Programm bilden eine lineare Folge von Anweisungen die genau ein mal abgearbeitet werden.



Programmbeispiel:

Schreiben Sie ein Python-Programm, welches nach der Eingabe von zwei (ganzen) Zahlen (z.B. a und b) die Grundrechenoperationen ausführt.

- beginne mit einer Kommentarzeile (z.B. Name …)
- kommentiere die Eingaben mit einem Text
- lege für jede Berechnung eine Ergebnisvariable fest (z.B. Summe: s)
- kommentiere die Ausgaben der Berechnungen
- speichere den Quellcode im Arbeitsverzeichnis unter **rechnen.py** ab

Ergänze weitere Berechnungen (Potenzieren, Ganzzahldivision)

Formatierungen / komplexe Berechnungen:

Für komplexere Berechnungen und Formatierungen werden spezielle Funktionen genutzt.

Funktionen sind Teilalgorithmen (Unterprogramme) die über einen Funktionsnamen aufgerufen und eingebunden werden können.

int(Wert/Variable)	*wandelt einen Wert/Variable in den Datentyp Ganzzahl (ohne Kommastelle) um*
float(Wert/Variable)	*wandelt einen Wert/Variable in den Datentyp Gleitkommazahl (mit Kommastelle) um*
str(Wert/Variable)	*wandelt einen Wert/Variable in den Datentyp Text um*

Eine Funktion entspricht einer Objektmethode mit name(Parameter).

Weitere Funktionen müssen aus einer Programmbibliothek (Modul) eingebunden werden.

Das Einbinden erfolgt zu Programmbeginn mit:

```
from modulname import *
```

Das Modul math

Übersicht zu ausgewählten Funktionen des Modules math

Der Aufruf erfolgt durch Angabe des Namens und eines Parameters (p) und liefert einen Ergebniswert e zurück.

e = name(p)

Die Konstanten pi und e werden ohne Parameter aufgerufen.

Modul: math	
Funktion	**Beschreibung**
round(z)	rundet die Zahl z ganzzahlig
pow(z, n)	n-te Potenz einer Zahl z (z^n)
sqrt(z)	Quadratwurzel einer Zahl z ($\sqrt[2]{z}$)
exp(x)	Exponentialfunktion (e^x)
log(x, a)	Logarithmus einer Zahl x zur Basis a $log_a(x)$
radians(w)	Umrechnung des Winkels w in Bogenmaß
degrees(b)	Umrechnung des Winkels r in Gradmaß
sin(x)	Sinus eines Winkels x
cos(x)	Cosinus eines Winkels x
tan(x)	Tangens eines Winkels x
asin(x)	Umkehrfunktion zu sin(x)
acos(x)	Umkehrfunktion zu cos(x)
atan(x)	Umkehrfunktion zu tan(x)
floor(z)	nächstkleiner ganzzahliger Wert von z
ceil(z)	nächstgrößerer ganzzahliger Wert von z
pi	Kreiszahl π=3,14...
e	Eulersche Zahl e=2,71…

Das Modul gturtle

Mit dem Modul **gturtle** kann eine Turtle (Schildkröte) auf einer Zeichenfläche gesteuert werden.

Beispiel:

Modul: gturtle	
Funktion	**Beschreibung**
makeTurtle()	Fenster mit Turtle öffnen
hideTurtle()	Turtle unsichtbar machen
showTurtle()	Turtle sichtbar machen
penUp()	Zeichenstift anheben (nicht zeichnen)
PenDown()	Zeichenstift absenken (zeichnen)
setPenColor("f")	Setzt die Zeichenfarbe als (engl.) Namen
setPenWidth(b)	Setzt die Breite b des Stiftes in Pixel
forward(s)	bewegt Turtle s Schritte vorwärts
back(s)	bewegt Turtle s Schritte rückwärts
left(w)	dreht Turtle um Winkel w nach links
right(w)	dreht Turtle um Winkel w nach rechts
delay(t)	wartet Zeit t in ms