

PROPÄDEUTIKUM INFORMATIK

SOSE 2019

Martin Mehlhose

ORGANISATORISCHES

ABLAUF PROPÄDEUTIKUM

- ca 14 Wochen Unterricht
- Montag 15:15 - 16:45 Uhr
- Raum Lu4 Zi 17

ABLAUF PROPÄDEUTIKUM

- Propädeutikum kann mit 3 LP bewertet werden
- Programmieraufgabe als Leistungsnachweis
- regelmäßige Teilnahme an den Veranstaltungen vorausgesetzt

ABLAUF PROPÄDEUTIKUM

Folien und Quellcode unter:

- <https://github.com/Francisde/PropaedeutikumSoSe2019>

Bei Fragen jeder Zeit eine E-Mail an:

- E-Mail: fm63byza@studserv.uni-leipzig.de

LITERATUR

- In der Campus-Bibliothek am Augustusplatz
- Jede Menge Tutorials im Web oder auf Youtube
- <https://www.tutorialspoint.com/python/>

STOFFÜBERSICHT

- Einführung in die objektorientierte Programmierung mit Python 3
- Modellierung mit UML
- Maschinenzahlen und Zahlenkonvertierung
- Algorithmen zum arbeiten auf verschiedenen Datenstrukturen

WARUM PYTHON?

- **objektorientiert**, dadurch leicht zu verstehen und zu erweitern
- **robust und sicher**, durch Datenkapselung und aktive Fehlerbehandlung
- **architekturneutral**, undabhängig von Hardware und Betriebssystem
- **leistungsfähig** vor allem durch externe Libraries

EINSTIEG IN PYTHON

- Beschreibung zum installieren von Python 3 folgt demnächst im Git
- Vorerst reicht Online-Interpreter: <https://repl.it/languages/python3>

ERSTES BEISPIEL

```
print("Hello World")
```

- Programm gibt den Text "Hello World" auf dem Bildschirm aus
- print Methode zur Ausgabe auf der Standardausgabe

DATENTYPEN

- im ersten Beispiel nur Ausgabe eines vorgegebenen Textes
- Ziel: beliebige Daten eingeben, verarbeiten und ausgeben

GRUNDELEMENTE PYTHON

GRUNDELEMENTE - IDENTIFIER

Um Variablen, Methoden, Klassen und Objekte ansprechen zu können werden diese mit Namen identifiziert.

- Regeln zum erstellen von Identifiern
 - Namen dürfen Buchstaben, Ziffern, Unterstrich "_" und Dollarzeichen "\$" enthalten
 - Namen dürfen nicht mit Ziffer beginnen
 - Namen dürfen nicht mit python Schlüsselwörtern übereinstimmen
 - Konvention: Variablen beginnen mit einem Kleinbuchstaben

PYTHON 3 SCHLÜSSELWÖRTER

Keywords in Python				
False	class	<u>finally</u>	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	<u>elif</u>	if	or	yield
assert	else	import	pass	
break	except	in	raise	

GRUNDELEMENTE - IDENTIFIER

Konventionen zur Namensvergabe:

- verwendung von "**sprechenden**" Namen
- **Variablen**namen beginnen mit einem Kleinbuchstaben
- Namen von **Klassen** beginnen mit Großbuchstaben
- bei zusammengesetzten Namen beginnt jedes Wort nach dem ersten mit einem Großbuchstaben
- Namen von **Konstanten** bestehen nur aus Großbuchstaben

BEISPIEL IDENTIFIER

```
zahlen_variable = 25  
string_variable = "text"  
kommazahlen_variable = 1.2
```

ELEMENTARE DATENTYPEN

GANZE ZAHLEN

- `int` [-2147483648,2147483647]

ELEMENTARE DATENTYPEN

GLEITKOMMAZAHLEN

- float $[10^{-45}, 10^{38}]$

LOGISCHE WERTE

- boolean Werte: True, False

ELEMENTARE DATENTYPEN

TEXT

Zusammenhängende Zeichen und Sätze werden mit dem Datentyp **String** gespeichert. Der Wert ist dann mit " " oder ' ' anzugeben.

Beispiel: **satz = "Das ist ein Satz"** oder **satz = 'Das ist ein Satz'**

ELEMENTARE DATENTYPEN

KOMMENTARE

Kommentare sind Bereiche im Quellcode die bei der Compoilierung und Programmausführung ignoriert werden. Sie dienen der Dokumentation des Quellcodes und sollen desen Lesbarkeit erhöhen.

ELEMENTARE DATENTYPEN

KOMMENTARE

- Einzeilige Kommentare
 - Kommentarbereich wird durch # markiert

Alles was in der selben Zeile dem # folgt wird ignoriert
- Mehrzeilige Kommentare
 - Kommentarbereich wird durch "" ... "" markiert

Alles was zwischen "" und "" steht wird ignoriert.

BEISPIEL KOMMENTARE

```
int variable1;           # einzeliges Kommentar

#int a=0.5;

"""
Das ist ein
mehrzeiliges Kommentar
"""
```

VARIABLEN IN PYTHON

Variablen sind Platzhalter im Speicher (Ähnlich wie in der Mathematik). Jede Variable hat einen eindeutigen **Namen**, einen **Typ** und einen **Wert**.

- Deklarieren

Zuordnung eines Namen

- Initialisieren

Zuweisung eines Wertes

Beispiel: myInteger=25

BEISPIEL VARIABLEN

```
variable1 = 25      // Deklarieren und Initialisieren  
  
// Deklarieren mehrerer Variablen ist möglich  
a,b,c = 2, 4, 5
```

GRUNDLEGENDE RECHENOPERATIONEN IN PYTHON

- Additions-Operator: +
- Subtraktions-Operator: -
- Divisions-Operator: /
- Multiplikations-Operator: *
- Rest-Operator: %

Klammerung wie beim "normalem" rechnen.

WEITERE MATHE FUNKTIONEN

Klammerung wie beim "normalem" rechnen.

- Wurzelfunktion: **sqrt(x)**;
- Potenzfunktion: **pow(a,b)**; -> repräsentiert a^b .
Achtung! Für Potenzfunktion nicht a^b benutzen.
- diverse trigonometrische und andere Funktion

AUSGABE VON WERTEN

Standardfunktion zur Ausgabe: `print()`

- `print("Ausgabe eines Textes")`
- `print(myVar)`
- Ausgabe mehrerer Strings oder Variablen durch Kommatrennung:
- `print("Ergebnis: ", myVar)`

KONTROLLSTRUKTUREN UND SCHLEIFEN

BOOLESCHE OPERATIONEN

- Negation: **not**
- Logische "und": **and**
- Logische "oder": **or**
- auch hier wieder Klammerung für die Auswertungsreihenfolge beachten

VERGLEICHSOPERATOREN

- gleich: ==
- ungleich: !=
- größergleich: >=
- kleingleich: <=
- strikt größer: >
- strikt kleiner: <

IF ANWEISUNG

- Syntax: **if(logischer Ausdruck): statements**
- Auf eine if Anweisung können mehrere Statements folgen
- Wichtig: Klammerung beachten!

- Syntax: **if(logischer Ausdruck): statements else: statements**

BEISPIEL IDENTIFIER

```
# einfache if Anweisung  
if(a<b):  
    max=b
```

```
# if-else Anweisung  
if(a<b):  
    max=b  
else:  
    max=a
```