Inhaltsverzeichnis

**Es wurden keine Einträge für das Inhaltsverzeichnis gefunden.**

Quelle: Einführung in Python 3 von Bernd klein, Hanser-Verlag 2013

# Kommandos und Programme

## Interaktive Shell

Er steht zwischen den Anwender und dem Betriebssystem. Dabei schützt der Shell das Betriebssystem vor dem Anwender und anderer Seite erspart er dem Anwender von dem Basis-Funktionen des Betriebssystems.

Eigenschaft:

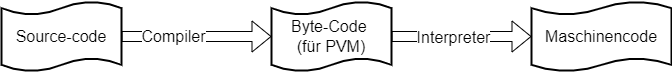
* Ein Befehl wird nach dem Drücken der Eingabetaste direkt ausgeführt.
* Der Unterstrich(\_) wird als Variable zum Aufrufen des vorherigen Ergebnisses verwendet.
* Vorheriger Befehl mit (↑) aufrufbar
* Verlassen des Shells: Ctrl-D oder mit der Funktion exit()
* Variable können deklariert werden und bleiben während der ganzen Session erreichbar.
* Schleife sind auch möglich

# Programme schreiben

* Ein Programm von der Datei hello.py 🡪 in dem Linux oder Windows Shell: python3 hello.py

# Bytecode und Maschinencode

Ablauf zur Ausführung eines Python-Script



PVM: Python-Virtual-Maschine

Fragen:

* Handelt es sich bei Python um eine Programmier- oder eine Skriptsprache?

Programm: Vor Ausführung wird es übersetzt bzw. kompiliert und dann ausgeführt. Kann mehrere Zeilen(tausend, Million) haben.

Skript: wird unmittelbar ausgeführt bzw. interpriert.

Python ist keine Programmiersprache und auch keine Skript Sprache. Aus der oberen Abbildung kann man entnehmen, dass ein Source code zuerst in Byte-Code in einer PVM übersetzt wird und in einer bestimmten Maschinencode übersetzt.

* Wird ein Python-Programm übersetzt oder interpretiert?

Siehe obige Abbildung

* Kann ein Python-code auf verschiedenen Rechnern oder Betriebssystemen laufen?

Ja, sie ist Betriebssystem unabhängig

* Worin liegt der Zusammenhang zwischen Python, Jython oder Java?

Der Ausführungsablauf entspricht dem von Java.

# Datentypen und Variablen

In Python wird der Datentyp einer Variablen automatisch erkannt und kann im Programmablauf geändert werden: Dynamische Typdeklaration.

*Type(i)*

*Isinstance(object,klasse)*

Die Funktion Typ gibt der aktuelle Datentyp von der Variabel i.

Die Funktion isinstance überprüft die Variable/Objekt auf die vorgegebene Klasse/Typ und gibt entweder „True“ oder „False“ zurück. Man kann auch mehrere Datentypen eingeben:

*Isinstance(object,(typ1, typ2))*

Typverletzung: Wenn ungültige Operationen zwischen Variablen verschiedener Datentyp auftreten. ZB. Addieren eines Strings und Integer.

# Sequenzielle Datentypen

Sequenzielle Datentypen sind Variablen, die mehrere Werte haben oder sequenziell angeordnet sind. Man unterscheidet:

* Strings: Anordnung beliebiger Zeichen. Ein Zeichen kann nicht durch Zuweisung geändert werden
* Lists: Anordnung beliebigen Objekten in eckigen Klammern: x = [4, 4.6, “text“]
* Tupel: Anordnung beliebigen Objekten in runden Klammern: x =(4, 4.6, “text“) und kann nicht mehr im Programm verändert werden.

Eigenschaften von sequenziellen Datentypen:

* Indexierung ist gleich.
* Die Slice- oder Ausschneiden-Funktionen sind bei allen gültig.

# Dictionaries

## Definition

Es gibt in Python noch die Kategorie Mapping als Datentyp und diese Kategorie hat einen einzigen implementierten Typ: Dictionary. Dictionary beinhaltet mehrere Objekte, die durch Schlüsselwörter abgebildet sind. Sie sind erweiterbar und veränderbar.

*Empty={} 🡪 Leer Dictionary*

*Empty = {“ham” : ‘yes’ , “spam” : ‘no’}*

*print(Empty) 🡪 {‘ham’ : ‘yes’ , ‘spam’ : ‘no’}*

*print(Empty[“ham”]) 🡪 ‘yes’*

*Empty[“spam”]=”yes”*

*print(Empty) 🡪 {‘ham’ : ‘yes’ , ‘spam’ : ‘yes’}*

* Python kann die Reihenfolge der Dictionary beliebig ausgeben.
* Der Zugriff auf einem wert ist durch das Schlüsselwort.
* Bei Zugriff auf einem nicht existieren Schlüsselwort wird den Error: KeyError erzeugt.

## Fehlerfreie Zugriffe auf Dictionaries