

# Einführung in die Programmierung

Ronja Düffel  
Alina Stürck  
WS2016/17

05. Oktober 2016

# Programmiersprachen

# Programmiersprachen

## Maschinenprogramme

- können direkt vom Computer verstanden und ausgeführt werden.

# Programmiersprachen

## Maschinenprogramme

- können direkt vom Computer verstanden und ausgeführt werden.
- bestehen aus Bit-Folgen

# Programmiersprachen

## Maschinenprogramme

- können direkt vom Computer verstanden und ausgeführt werden.
- bestehen aus Bit-Folgen (0-en und 1-en),

# Programmiersprachen

## Maschinenprogramme

- können direkt vom Computer verstanden und ausgeführt werden.
- bestehen aus Bit-Folgen (0-en und 1-en),
- für Menschen nahezu unverständlich

# Programmiersprachen

## Maschinenprogramme

- können direkt vom Computer verstanden und ausgeführt werden.
- bestehen aus Bit-Folgen (0-en und 1-en),
- für Menschen nahezu unverständlich

## Höhere Programmiersprachen

- für Menschen besser zu lesen und zu verstehen
- *Quelltext* = Programm in höherer Programmiersprachen
- für Computer unverständlich  
⇒ Quelltext muss in Maschinenprogramm übersetzt werden!

# Python





# Python



- Höhere Programmiersprache
- Als Lehrsprache entwickelt
  - Übersichtlichkeit
  - einfache Erlernbarkeit
- benutzt zur Übersetzung einen Interpreter

# IDLE

- Entwicklungsumgebung für Python
- unterstützt bei der Programmierung
- Interaktive Python-Shell mit
  - Syntax-Highlighting
  - Autovervollständigung
  - Anzeige der Befehlssyntax
- Editor zum Programme schreiben

# Datentypen

## Wahrheitswerte Bool

- True    False
- Operatoren: and, or und not

# Datentypen

## Wahrheitswerte Bool

- True    False
- Operatoren: and, or und not

## Zahlen

- Ganzzahlen int (fehlender Dezimalpunkt)

# Datentypen

## Wahrheitswerte Bool

- True    False
- Operatoren: and, or und not

## Zahlen

- Ganzzahlen `int` (fehlender Dezimalpunkt)
- Gleitkommazahlen `float`

# Datentypen

## Wahrheitswerte Bool

- True    False
- Operatoren: and, or und not

## Zahlen

- Ganzzahlen `int` (fehlender Dezimalpunkt)
- Gleitkommazahlen `float` (Dezimalpunkt)

# Datentypen

## Wahrheitswerte Bool

- True    False
- Operatoren: and, or und not

## Zahlen

- Ganzzahlen `int` (fehlender Dezimalpunkt)
- Gleitkommazahlen `float` (Dezimalpunkt)
- Operatoren: +, -, \*, /, //, %, \*\*

# Datentypen

## Wahrheitswerte `Bool`

- `True`    `False`
- Operatoren: `and`, `or` und `not`

## Zahlen

- Ganzzahlen `int` (fehlender Dezimalpunkt)
- Gleitkommazahlen `float` (Dezimalpunkt)
- Operatoren: `+`, `-`, `*`, `/`, `//`, `%`, `**`

## Zeichenketten `string`

- in Anführungszeichen  
    `'Und nun zu etwas ganz anderem...'`



# Datentypen

## Wahrheitswerte Bool

- True    False
- Operatoren: and, or und not

## Zahlen

- Ganzzahlen int (fehlender Dezimalpunkt)
- Gleitkommazahlen float (Dezimalpunkt)
- Operatoren: +, -, \*, /, //, %, \*\*

## Zeichenketten string

- in Anführungszeichen
  - ‘Und nun zu etwas ganz anderem...’
  - ‘‘Setz dich, nimm dir’n Keks...’’

# Datentypen

## Wahrheitswerte Bool

- True    False
- Operatoren: and, or und not

## Zahlen

- Ganzzahlen int (fehlender Dezimalpunkt)
- Gleitkommazahlen float (Dezimalpunkt)
- Operatoren: +, -, \*, /, //, %, \*\*

## Zeichenketten string

- in Anführungszeichen
  - ‘Und nun zu etwas ganz anderem...’
  - ‘‘Setz dich, nimm dir’n Keks...’’
  - ‘‘‘Er sagte:‘‘Geht’s jetzt los?’’ ’’’

# Built-in-Funktionen (int, float)

- `abs(x)`: gibt den Betrag  $|x|$  der Zahl  $x$  zurück
- `max(a,b)`: gibt das Maximum von  $a$  und  $b$  zurück
- `min(a,b)`: gibt das Minimum von  $a$  und  $b$  zurück
- `round(x,n)`: gibt den gerundeten Wert von  $x$  zurück ( $n$ -Nachkommastellen)

# Built-in-Funktionen (string)

- `<string>.isalpha()`: prüft, ob String nur aus Buchstaben besteht
- `<string>.isdigit()`: prüft, ob String nur aus Ziffern besteht
- `ord(c)`: gibt die Nummer des Zeichens `c` zurück
- `chr(<int>)`: gibt das Zeichen zur Ganzzahl `i` zurück
- `+`: fügt zwei Strings zusammen (Operator)

# nützliche Funktionen

- `int(x)`: wandelt `x` in einen ganzzahligen Wert um
- `float(x)`: wandelt `x` in einen Gleitkommawert um
- `string(x)`: wandelt `x` in einen String um
- `input()`: Benutzereingabe lesen
- `print()`: Bildschirmausgabe

# Variablen

- Möglichkeit Informationen zu speichern
- bezeichnet einen bestimmten Speicherbereich
- hat immer:
  - eine Namen
  - einen Typ
  - einen Wert
- in Python durch Zuweisung `<Name> = <Ausdruck>` angelegt

# Variablen

- Möglichkeit Informationen zu speichern
- bezeichnet einen bestimmten Speicherbereich
- hat immer:
  - eine Namen
  - einen Typ
  - einen Wert
- in Python durch Zuweisung `<Name> = <Ausdruck>` angelegt

**! Vergibt man denselben Namen zweimal, so wird der Wert der Variablen überschrieben !**

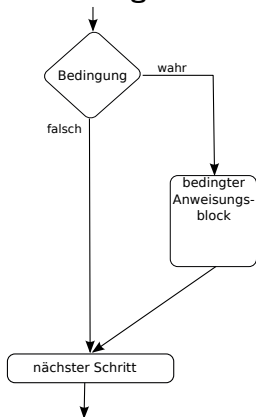
# Kontrollstrukturen

- steuern den Programmablauf



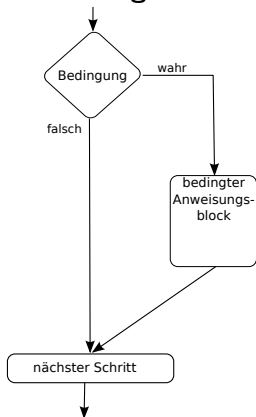
# Verzweigung

## if-Anweisung

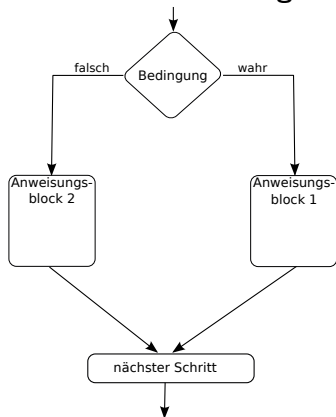


# Verzweigung

## if-Anweisung

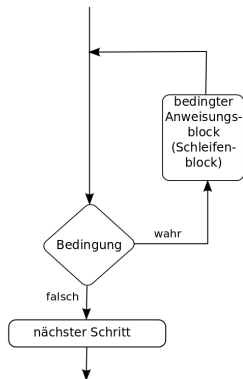


## if...else-Anweisung



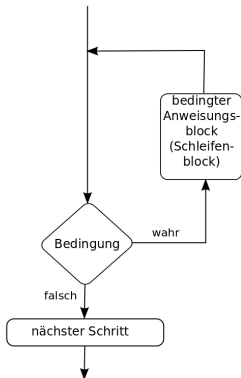
# Schleifen

## while-Schleife

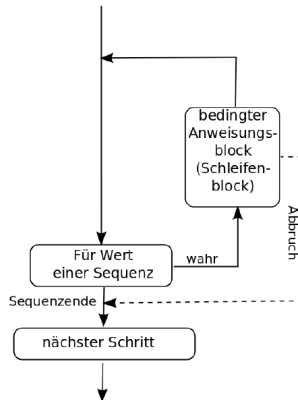


# Schleifen

## while-Schleife



## for-Schleife



# Schleifen-Kontrollanweisungen

- `break`: beendet Schleife sofort; Programmfluss springt zur ersten Anweisung nach dem Schleifenrumpf
- `continue`: bricht aktuellen Schleifendurchlauf ab; Programmfluss springt zum Schleifenkopf
- `pass`: wenn Python eine Anweisung verlangt, aber nichts gemacht werden soll