

# Algorithmen und Programmieren II

## Einführung in Python



SS 2012

Prof. Dr. Margarita Esponda

## Was ist Python?

- eine **Skript-Sprache**
- Anfang der **90er** Jahre entwickelt.
- Erfinder: Guido van Rossum an der **Universität von Amsterdam**
- Unterstützung des **strukturierten Programmierens** aus der **ABC**-Sprache übernommen.

ALGOL → ABC → Python

- **Philosophie:**
  - Einfachheit, Lesbarkeit und Orthogonalität
  - **Schnelle Programmentwicklung** ist **wichtiger** als **schnelle Programme**

## Skriptsprachen vs. konventionelle Sprachen

Durch die rapide Internet-Entwicklung haben Skript-Sprachen eine rasante Entwicklung erlebt.

Perl  
Tcl  
Rexx  
JavaScript  
ActionScript  
Python  
Ruby  
PHP  
...

- Web-Anwendungen
- Benutzerschnittstellen
- Prototyp-Entwicklung

- interpretiert
- dynamische Typ-Kontrolle
- Kopplung-Sprachen/"glue"-Sprachen

C  
C++  
Java  
C#  
...

- kompiliert
- statische Typ-Kontrolle
- Komponentensprachen

## Skriptsprachen vs. konventionelle Sprachen

### Vorteile

- Schnell erlernbar
- schnelles Prototyping
- Kleiner Fehlersuche-Zyklus
- Kleinere Programme und schnellere Entwicklung
- Unterstützen viele moderne Paradigmen
- Breite Verwendbarkeit in der Software-Industrie

### Nachteile

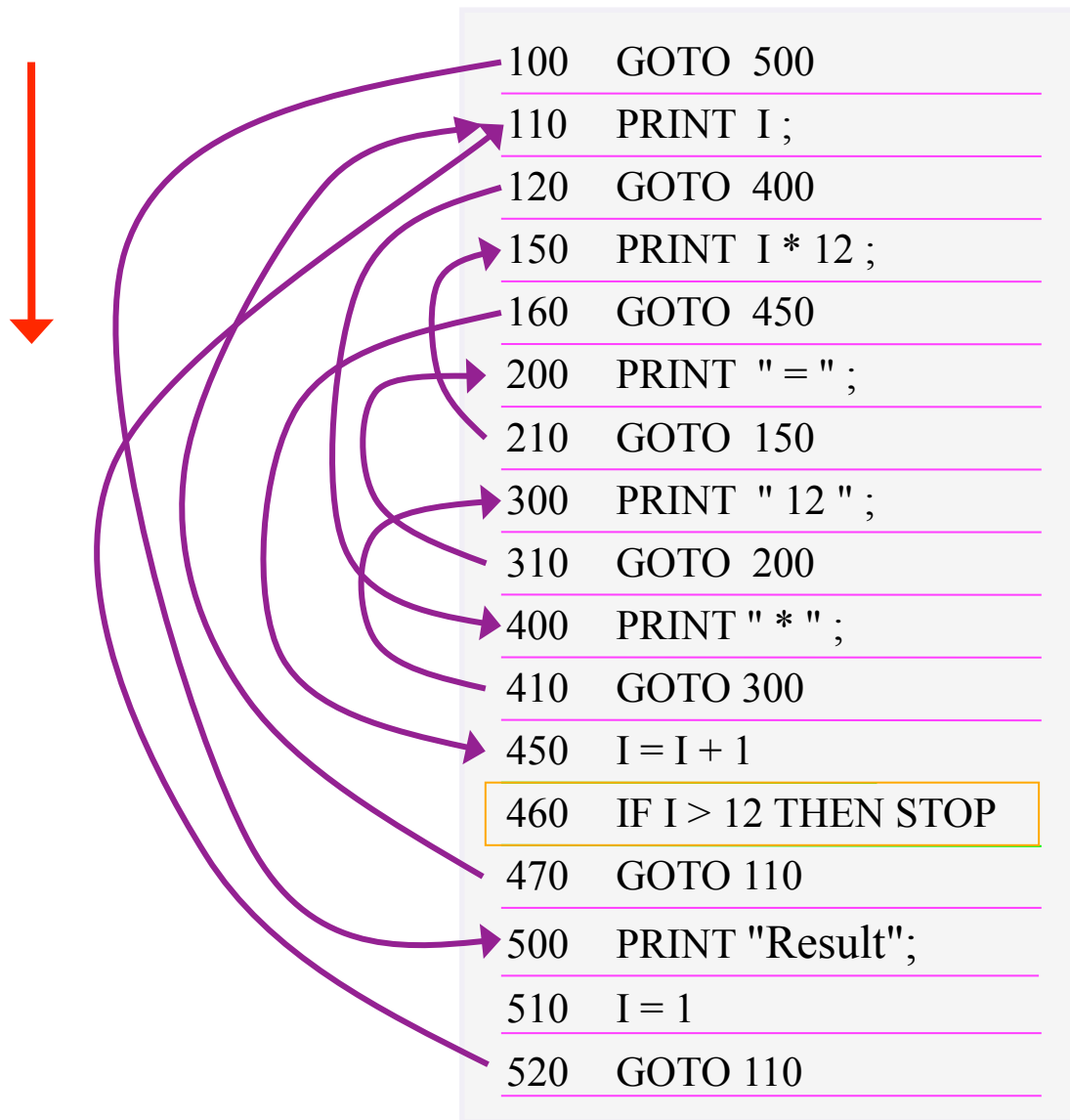
- Speicherverbrauch ist größer
- langsamer
- Keine statische Typkontrolle
- Für bestimmte Anwendungen nicht geeignet

**Jedoch: große Unterschiede zwischen den einzelnen Skript-Sprachen!**

# Die ABC Programmiersprache

- Speziell für die **Lehre** entwickelt
- als Ersatz für **Basic** und **Pascal** konzipiert
- die **Programme** sind sehr kompakt
- von Anfang an **strukturiertes Programmieren**
- **sehr interaktiv**
- wird **interpretiert**

# „GOTOs und Spaghetti code“



```
Result
1 * 12 = 12
2 * 12 = 24
3 * 12 = 36
...
```

## GOTOs in Fortran

```
10 IF (X .GT. 0.00001) GO TO 30
20 X = -X
   IF (X .LT. 0.00001) GO TO 50
30 IF (X*Y .LT. 0.001) GO TO 40
   X = X-Y-Y
40 X = X+Y
   ...
50 CONTINUE
   X = A
   Y = B-A
   GO TO 20
   ...
```

## Die GOTO-Anweisung

- die GOTO-Anweisung ist die **meist umstrittene Kontroll-Anweisung** in der Welt der höheren Programmiersprachen.
- **theoretisch** kann jeder GOTO nach vorne mit einer *if-else*-Anweisung und jeder GOTO nach hinten mit einer *while*-Schleife ersetzt werden.
- **Assembler-Befehle** haben keine *if-else*- oder *while*-Anweisungen sondern nur bedingte oder unbedingte GOTOs.
- Die Äquivalenz ist klar, sonst gäbe es keine Übersetzer.
- **C hat GOTOs, und Java hat beschränkte GOTOs.**
- **Python hat keine GOTO-Anweisung.**



# Lesbarkeit

## Pseudocode

```
quicksort( A, p, r )
  if p < r
    then q ← partition( A, p, r )
         quicksort( A, p, q-1 )
         quicksort( A, q+1, r )

partition( A, p, r )
  x ← A[r]
  i ← p-1
  for j ← p to r-1
    do if A[j] ≤ x
       then i ← i+1
          exchange A[i] ↔ A[j]
  exchange A[i+1] ↔ A[r]
  return i+1
```

## Python

```
def quicksort( A, p, r ):
    if p < r:
        q = partition( A, p, r )
        quicksort( A, p, q-1 )
        quicksort( A, q+1, r )

def partition( A, p, r ):
    x = A[r]
    i = p-1
    for j in range( p, r ):
        if A[j] <= x:
            i = i+1
            exchange( A, i, j )
    exchange( A, i+1, r )
    return i+1
```

## Einrücken anstatt Klammern

- keine *begin ... end* wie in Pascal oder `{ ... }` wie in C
- Die Anweisungen innerhalb eines Blocks beginnen immer an der gleichen Zeilenspalte

Beispiel:

```
x = int ( input() )
if x <= 0:
    x = 0
    print ('zero')
else:
    x = 1
    print ('one')
```

## Python unterstützt mehrere Paradigmen

- flexibel in der Handhabung verschiedener Programmier-Paradigmen
  - **Imperative Programmierkonzepte**
  - Objektorientierte Programmierung
  - Funktionale Programmierung
  - Aspektorientierte Programmierung
- einfaches Einsetzen verschiedener Programmier-Techniken
  - Strukturierte Programmierung
  - Entwurf gemäß Vertrag (DBC)

## Python-Interpreter

Linux und Mac OS



Der Python-Interpreter  
ist Teil der Standard-  
Installation.



python

Windows und Mac OS



IDLE

Integrierte DeveLopment  
Environment

[www.python.org/idle](http://www.python.org/idle)

python 2.x    python 3.x

Wir werden python 3.x verwenden.