



### Laufzeitfehler (Semantische Fehler)

- Programme mit Syntaxfehlern lassen sich nicht starten.
- Semantische Fehler sind dagegen Fehler, die die Bedeutung des Programmcodes betreffen

```

a = "String"
print a[10]
  
```

a besteht nur aus 6 Zeichen

- Solche Fehler können das Programm zum Absturz bringen, das Programm friert ein (Endlosschleifen) oder verhält sich merkwürdig

Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 7

### Designfehler

- Designfehler sind Fehler im Grundkonzept einer Software
- Fehlerarten:
  - Fehler in der Anforderungsdefinition / Spezifikation
    - Das Programm tut nicht das, was der Kunde wollte
    - beruhen meist auf mangelnder Kenntnis des Fachgebietes, für das die Software entwickelt wird
  - Fehler im Softwaredesign
    - Das Programm läßt sich später nicht mehr verstehen / warten / verbessern
    - beruhen oft auf mangelnder Erfahrung des Softwareentwicklers

Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 8

### Vermeidung von Fehlern

- Je früher in einem Entwicklungsprozeß ein Fehler auftritt und je später er entdeckt wird, desto aufwendiger ist es, ihn zu beheben.
- Designfehler lassen sich oft durch gute Planung und Wahl eines geeigneten Softwaredesigns weitestgehend vermeiden
- Syntaktische Fehler erkennt meist der Compiler oder der Interpreter
- Beim Auffinden semantischer Fehler helfen (für bestimmte Fälle / Klassen von Fehlern) spezielle Programme

Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 9

### Debugger

- Nicht immer sind semantische Fehler so einfach zu finden wie in dem Beispiel oben
- Deswegen hilft ein spezielles Programm, der sogenannte Debugger, dem Programmierer bei der Fehlersuche
- Ein Debugger ermöglicht die Ablaufverfolgung des Programms in einzelnen Schritten
- Mit Python wird ein Debugger mitgeliefert

Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 10

### Python-Debugger

- Die Python-IDE Idle starten
- Das Menü „Debugger“ wählen
- Auf Menüpunkt „Debugger“ drücken

Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 11

### Python-Debugger

- Das Debugger-Fenster wird geöffnet

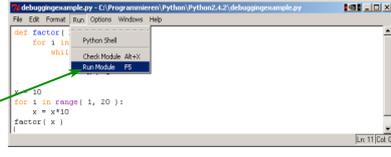
Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 12

### Python-Debugger

- Das betreffende Programm mit Idle öffnen



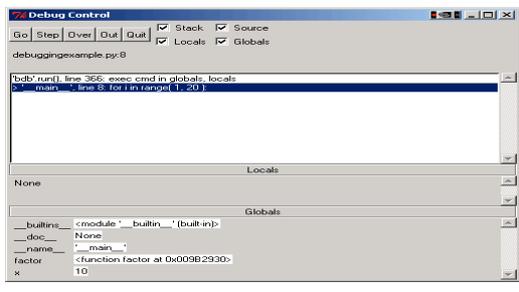
- Und starten



Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 13

### Python-Debugger

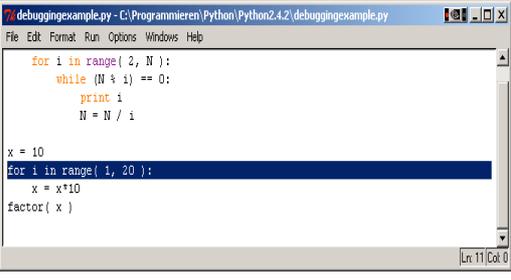
- Im Debugger-Fenster sieht man den Stack, sowie die Werte der lokalen und globalen Variablen



Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 14

### Python-Debugger

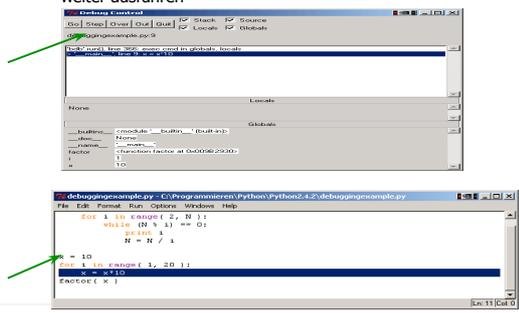
- Im Sourcecode-Fenster wird die aktuelle Position, an der der Debugger das Programm gerade hält, angezeigt



Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 15

### Python-Debugger

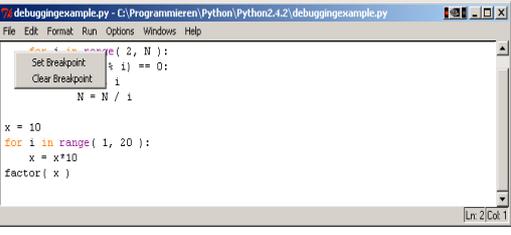
- Mit dem Step-Button kann man das Programm schrittweise weiter ausführen



Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 16

### Python-Debugger

- Wenn man nur an bestimmten Stellen des Source interessiert ist kann man per Rechtsklick im Sourcecode-Fenster einen Breakpoint setzen und mittels **Go**-Taste im Debugger-Fenster bis zu dieser Stelle laufen lassen



Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 17