

# Das `algorithm` Paket v1.2b

Jochen Rüttschlin

19. März 1998

## Zusammenfassung

Makros für die Darstellung von Algorithmen in einer Pseudo-Programmiersprache.

---

`\begin{algorithm}[Name] ... \end{algorithm}`

Die Umgebung, innerhalb der die folgenden Befehle angewendet werden können und die die korrekte Formatierung (Einrückung) der Befehle steuert. Optional kann ein *Name* für den Algorithmus bzw. die Prozedur angegeben werden, der in Capitälchen über den Algorithmus gesetzt wird. Die Zeilen der Algorithmusdarstellung werden fortlaufend durchnummeriert.

`\keyword{Schlüsselwort}`

Darstellung des *Schlüsselwortes* in der Schriftart für Schlüsselwörter.

`\assign{Variable}{Wert}`

Darstellung einer Wertzuweisung ohne Zeilenvorschub zur Verwendung innerhalb eines anderen Befehls (z. B. als Initialisierung einer `for`-Schleife). Die Verarbeitung von *Variable* und *Wert* erfolgt im Mathematikmodus.

`\assignment{Variable}{Wert}`

Darstellung einer Wertzuweisung als eigenständige Anweisung (wie `\assign`, nur mit einem Zeilenvorschub am Ende der Zeile). Die Verarbeitung von *Variable* und *Wert* erfolgt im Mathematikmodus.

`\function{Funktionsname}[Argument]`

Darstellung eines Funktionsaufrufes ohne Zeilenvorschub zur Verwendung innerhalb eines anderen Befehls (z. B. als rechte Seite einer Wertzuweisung). Die Verarbeitung von *Argument* erfolgt im Mathematikmodus.

`\functioncall{Funktionsname}[Argument]`

Darstellung eines Funktionsaufrufes als eigenständige Anweisung (wie `\function`, nur mit einem Zeilenvorschub am Ende der Zeile). Die Verarbeitung von *Argument* erfolgt im Mathematikmodus.

`\command{Anweisung}{Argument}`

Darstellung einer Anweisung mit der Bezeichnung *Anweisung* in der Schlüsselwortschriftart mit zusätzlichem *Argument* (z. B. zur Darstellung einer `return`-Anweisung). Die Verarbeitung von *Argument* erfolgt im Mathematikmodus.

`\command*{Anweisung}{Argument}`

Darstellung einer Anweisung mit der Bezeichnung *Anweisung* in der Schlüsselwortschriftart mit zusätzlichem *Argument* in Anführungszeichen (z. B. zur Darstellung einer Fehlermeldung).

`\metacommand{Text}`

Darstellung einer Anweisung, die aus erklärendem *Text* besteht.

`\whileloop{boolean}{Rumpf}`

Darstellung einer `while`-Schleife mit der Ausführungsbedingung *boolean* und dem Schleifenrumpf *Rumpf*. Die Verarbeitung von *boolean* erfolgt im Mathematikmodus.

`\repeatloop{Rumpf}{boolean}`

Darstellung einer `repeat-until`-Schleife mit der Abbruchbedingung *boolean* und dem Schleifenrumpf *Rumpf*. Die Verarbeitung von *boolean* erfolgt im Mathematikmodus.

`\doloop`

Darstellung einer `do`-Schleife (Endlosschleife) mit dem Schleifenrumpf *Rumpf*.

`\forloop{von}{bis}[Schritte]{Rumpf}`

Darstellung einer `for`-Schleife mit der Initialisierungsanweisung *von*, der Abbruchbedingung (obere oder unter Grenze der Laufvariablen) *bis*, einer Optionalen Angabe der Schrittweite *Schritte* und dem Schleifenrumpf *Rumpf*. Die Verarbeitung von *von*, *bis* und *Schritte* erfolgt im Mathematikmodus.

`\forloop*{von}{bis}[Schritte]{Rumpf}`

Eine Sternform des vorherigen Befehls stellt ebenfalls eine `for`-Schleife dar, fügt jedoch zwei

den Werten *von* und *bis* anstelle des Schlüsselwortes „to“ das Wort „downto“ ein.

`\foreachloop{Menge}{Rumpf}`

Darstellung einer `for`-Schleife, die über die Elemente einer *Menge* läuft und den Schleifenrumpf *Rumpf* hat. Die Verarbeitung von *Menge* erfolgt im Mathematikmodus.

`\conditional{boolean}{then-Zweig}[else-Zweig]`

Darstellung einer `if`-Anweisung, deren Wahrheitswert *boolean* ist und die einen (zwingenden) *then-Zweig* und einen optionalen *else-Zweig* hat. Die Verarbeitung von *boolean* erfolgt im Mathematikmodus.