Syntax und Semantik von Python im Rahmen von Inf-I1-2FNF

Version 3

(Änderungen zu Version 2 blau)

Datentypen

Bezeichner	Wertebereich	Bemerkung
int	Ganze Zahlen	Prinzipiell ohne Beschränkung des Wertebereichs!
float	Rationale Zahlen	Nicht alle rationalen Zahlen sind tatsächlich darstellbar!
str	Zeichenketten	Prinzipiell beliebige Länge
bool	True / False	
list	Zusammenfassung von Elementen	

Zu jedem Datentyp gibt es die Möglichkeit einer expliziten Typumwandlung durch die Verwendung des Befehls X(…), wobei X für den Datentyp steht in den umgewandelt werden soll (z.B. str(12))

Operatoren

Bezeichner	Bedeutung	Möglich für Datentyp
+	Addition Konkatenation	int, float str, list
-, *, /, **	Subtraktion, Multiplikation, Division, Potenz	int, float
%	Ganzzahliger Divisionsrest	int
and, or, not	Logisches „Und“, „Oder“, „Nicht“	bool
^	Logisches „Exklusives Oder“ (XOR) Bitweises „Exklusives Oder“	bool int
[n]	Zugriff auf das (n+1)-te Element einer Liste	list
[a:b]	Kopieren der Element zwischen den Positionen a und (b-1) einer Liste. Fehlen a und/oder b wird 0 bzw. die Länge der Liste verwendet.	list

Zuweisung

Syntaktische Form

Bezeichner ‘=‘ Ausdruck

Semantische Regel

Falls „Bezeichner“ eine neue Variable im aktuellen Lebensbereich ist, reserviere einen Speicherplatz und speichere dort den Wert der rechten Seite des Zuweisungsoperators.

Andernfalls finde im aktuellen Lebensbereich den Speicherbereich, der der Variable bereits zugeordnet wurde und überschreibe den Wert mit dem Wert der rechten Seite des Zuweisungsoperators.

Hinweise

- Der Datentyp einer Variablen wird von Python automatisch festgelegt und kann sich mit jeder Zuweisung implizit oder explizit (Typumwandlung) ändern.
- Der Datentyp bestimmt, wie die Variable verwendet werden kann.

Lesender Zugriff auf Variable

Syntaktische Form

Bezeichner

Semantische Regel

Finde den Speicherbereich, der der Variablen im aktuellen oder globalen Lebensbereich zugeordnet ist und verwende den dort gespeicherten Wert.

Falls eine Variable im aktuellen Lebensbereich verschattet werden kann, so verwende ausschließlich den aktuellen Lebensbereich.

(Kann der Speicherbereich nicht gefunden werden, erzeuge einen Laufzeitfehler)

Hinweise

- Kann als eigener Ausdruck oder als Teil eines Ausdrucks (z.B. auf der rechten Seite einer Zuweisung) vorkommen

Bedingung

Syntaktische Form

‘if‘ Bedingung ‘:‘ Anweisung

Semantische Regel

Falls Bedingung den Wert „True“ hat, führe „Anweisung“ aus.
Andernfalls setze Ausführung direkt nach der bedingten Anweisung fort.

Hinweise

- Als Bedingung kann jeder Ausdruck verwendet werden, der einen Wahrheitswert zurückliefert
- Anweisung kann auch ein (eingerückter) Block sein!

Bedingung mit Alternative

Syntaktische Form

‘if‘ Bedingung ‘:‘ Anweisung

‘else‘ ‘:‘ Anweisung

Semantische Regel

Falls Bedingung den Wert „True“ hat, führe die erste Anweisung aus, andernfalls die zweite.

Hinweise

- Als Bedingung kann jeder Ausdruck verwendet werden, der einen Wahrheitswert zurückliefert
- Anweisung kann auch ein (eingerückter) Block sein!
- Anstelle else: if x: … kann die Kurzform elif x: … verwendet werden.

Wiederholung mit fester Anzahl an Wiederholungen

Syntaktische Form

‘for‘ Bezeichner ‘in‘ Bezeichner ‘:‘ Anweisung

Semantische Regel

Führe die Anweisung für jeden Eintrag in der durch den zweiten Bezeichner gegebenen Datenstruktur aus und setze vor der Ausführung die durch den ersten Bezeichner gegebene Variable auf diesen Eintrag.

Hinweise

- Anweisung kann auch ein (eingerückter) Block sein!

Wiederholung mit Abbruchbedingung

Syntaktische Form

‘while’ Bedingung ‘:’ Anweisung

Semantische Regel

Falls Bedingung wahr, führe Anweisung aus und setze Ausführung danach unmittelbar vor der Wiederholung fort. Andernfalls setze die Ausführung unmittelbar nach der Wiederholung fort.

Hinweise

- Anweisung kann auch ein (eingerückter) Block sein!
- Erst nach der kompletten Ausführung der Anweisung (bzw. des Blocks!) wird die Bedingung erneut überprüft!

Unterprogrammaufruf

Syntaktische Form

Bezeichner ‘(‘ [Ausdruck] {‘,‘ Ausdruck} ‘)‘

Semantische Regel

Beim Aufruf eines Unterprogramms, führe den Code der Definition des Unterprogramms aus und lege einen neuen Lebensbereich an. Für jeden Parameter, lege eine lokale Variable an und belege diese mit dem Wert des Parameters in dem Aufruf.

Am Ende der Ausführung des Unterprogramms, löschen den angelegten Lebensbereich und setze die Ausführung nach der Stelle des Aufrufs fort. Verwende den mit „return“ angegebenen Wert (oder „None“) als Wert des Unterprogrammaufrufs.

Hinweise

-

Unterprogrammdefinition

Syntaktische Form

‘def‘ Bezeichner ‘(‘ [Ausdruck] {‘,‘ Ausdruck} ‘)‘ ‘:‘

Semantische Regel

Bei der Definition eines Unterprogramms, speichere den Beginn des Unterprogramms zusammen mit dem Bezeichner. Setzte die Programmausführung unmittelbar nach dem Unterprogramm fort.

Hinweise

- Unterprogramm muss im Programmtext vor der Verwendungsstelle definiert werden

„Basisbefehle“

Bezeichner	Bedeutung	Beispiel	Import nötig?
print	Ausgabe auf dem Bildschirm	print(“Hello world!“)	Nein
input	Benutzereingabe einlesen (als Zeichenkette!)	input(“Bitte Namen eingeben:“)	Nein
min, max	Minimum / Maximum einer Liste von Werten bestimmen	min(1,2,3) oder min([1,2,3])	Nein
range(a, b)	Liste aller ganzer Zahlen zwischen a und b-1	range(1, 10), range(-3, 5)	Nein
quit	Beendet die Programmausführung	quit()	Nein

Arbeiten mit Dateien

Bezeichner	Bedeutung	Beispiel	Import nötig?
isfile	Prüft ob eine Datei existiert	if not isfile(“landscape.jpg“): quit()	os.path
open	Öffnen einer Datei zum Lesen Öffnen einer Datei zum Schreiben	datei = open(“template.html“) datei = open(“ausgabe.html“, “w+“)	Nein
read	Liest den Inhalt einer vorher zum Lesen geöffneten Datei als Zeichenkette ein	inhalt = datei.read()	Nein
write	Schreiben von Daten in eine vorher mit zum Schreiben geöffneten Datei	datei.write(“Hello world“)	Nein
close	Beendet den Zugriff auf eine Datei	datei.close()	Nein
reader	Liest die Zeilen einer vorher zum Lesen geöffneten CSV Datei ein	zeilen = reader(daten)	csv
imread, imwrite	Bilddaten aus einer Datei lesen/speichern	image = imread('landscape.jpg') imwrite('landscape.jpg', image)	imageio

Arbeiten mit Zeichenketten

Bezeichner	Bedeutung	Beispiel	Import nötig?
match	Regulären Ausdruck auf einer Zeichenkette auswerten	match(r“[0-9+]“, eingabe)	re
replace	Ersetzen aller Vorkommen einer Zeichenkette in einer Zeichenkette, durch eine andere Zeichenkette	ausgabe.replace(“{content}“, html)	Nein