

Languages-beta: SL-3-Statements *

The PPlanCompS Project

SL-3-Statements.cbs | PLAIN | PRETTY

Language "SL"

3 Statements

Syntax $Stmt : stmt ::= expr\ ;\ ;$
 $| \text{'return' } expr\ ;\ ;$
 $| \text{'return' } ;\ ;$
 $| \text{'if' ' (' } expr\)\ ' block$
 $| \text{'if' ' (' } expr\)\ ' block\ \text{'else' } block$
 $| \text{'while' ' (' } expr\)\ ' block$
 $| \text{'break' } ;\ ;$
 $| \text{'continue' } ;\ ;$
 $| block$

$Block : block ::= \{ ' stmt^* ' \}$

Rule $\llbracket \text{'if' ' (' } Expr\)\ ' Block \rrbracket : stmt =$
 $\llbracket \text{'if' ' (' } Expr\)\ ' Block\ \text{'else' ' (' } ' \rrbracket$

Semantics $exec \llbracket Stmt^* : stmt^* \rrbracket : \Rightarrow \text{null-type}$
Rule $exec \llbracket Expr\ ;\ ; \rrbracket = \text{effect}(\text{eval} \llbracket Expr \rrbracket)$
Rule $exec \llbracket \text{'return' } Expr\ ;\ ; \rrbracket = \text{return}(\text{eval} \llbracket Expr \rrbracket)$
Rule $exec \llbracket \text{'return' } ;\ ; \rrbracket = \text{return}(\text{null-value})$
Rule $exec \llbracket \text{'if' ' (' } Expr\)\ ' Block_1\ \text{'else' } Block_2 \rrbracket =$
 $\text{if-true-else}(\text{bool eval} \llbracket Expr \rrbracket, exec \llbracket Block_1 \rrbracket, exec \llbracket Block_2 \rrbracket)$
Rule $exec \llbracket \text{'while' ' (' } Expr\)\ ' Block \rrbracket =$
 $\text{handle-break}(\text{while-true}(\text{bool eval} \llbracket Expr \rrbracket, \text{handle-continue}(exec \llbracket Block \rrbracket)))$
Rule $exec \llbracket \text{'break' } ;\ ; \rrbracket = \text{break}$
Rule $exec \llbracket \text{'continue' } ;\ ; \rrbracket = \text{continue}$
Rule $exec \llbracket \{ ' Stmt^* ' \} \rrbracket = exec \llbracket Stmt^* \rrbracket$
Rule $exec \llbracket \rrbracket = \text{null-value}$
Rule $exec \llbracket Stmt\ Stmt^+ \rrbracket =$
 $\text{sequential}(exec \llbracket Stmt \rrbracket, exec \llbracket Stmt^+ \rrbracket)$

*Suggestions for improvement: plancomps@gmail.com.
Reports of issues: <https://github.com/plancomps/CBS-beta/issues>.