

# Languages-beta: SL-3-Statements \*

The PPlanCompS Project

SL-3-Statements.cbs | PLAIN | PRETTY

---

Language "SL"

## 3 Statements

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

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

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

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

---

\*Suggestions for improvement: [plancomps@gmail.com](mailto:plancomps@gmail.com).  
Reports of issues: <https://github.com/plancomps/CBS-beta/issues>.