

# Languages-beta: SL-3-Statements

The PPlanCompS Project

Languages-beta/SL/SL-3-Statements/SL-3-Statements.cbs\*

*Language* "SL"

## 3 Statements

```
Syntax Stmt : stmt ::= expr ;  
                | return expr ;  
                | return ;  
                | if ( expr ) block  
                | if ( expr ) block else block  
                | while ( expr ) block  
                | break ;  
                | continue ;  
                | block  
Block : block ::= { stmt* }
```

```
Rule [ [ if ( Expr ) Block ] : stmt =  
      [ [ if ( Expr ) Block else { } ] ]
```

---

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

Semantics  $\text{exec} \llbracket \text{Stmt}^* : \text{stmt}^* \rrbracket : \Rightarrow \text{null-type}$

Rule  $\text{exec} \llbracket \text{Expr} ; \rrbracket =$   
 $\text{effect}(\text{eval} \llbracket \text{Expr} \rrbracket)$

Rule  $\text{exec} \llbracket \text{return Expr} ; \rrbracket =$   
 $\text{return}(\text{eval} \llbracket \text{Expr} \rrbracket)$

Rule  $\text{exec} \llbracket \text{return} ; \rrbracket =$   
 $\text{return}(\text{null-value})$

Rule  $\text{exec} \llbracket \text{if ( Expr ) Block}_1 \text{ else Block}_2 \rrbracket =$   
 $\text{if-true-else}(\text{bool eval} \llbracket \text{Expr} \rrbracket,$   
 $\text{exec} \llbracket \text{Block}_1 \rrbracket,$   
 $\text{exec} \llbracket \text{Block}_2 \rrbracket)$

Rule  $\text{exec} \llbracket \text{while ( Expr ) Block} \rrbracket =$   
 $\text{handle-break}(\text{while-true}(\text{bool eval} \llbracket \text{Expr} \rrbracket,$   
 $\text{handle-continue}(\text{exec} \llbracket \text{Block} \rrbracket)))$

Rule  $\text{exec} \llbracket \text{break} ; \rrbracket =$   
 $\text{break}$

Rule  $\text{exec} \llbracket \text{continue} ; \rrbracket =$   
 $\text{continue}$

Rule  $\text{exec} \llbracket \{ \text{Stmt}^* \} \rrbracket =$   
 $\text{exec} \llbracket \text{Stmt}^* \rrbracket$

Rule  $\text{exec} \llbracket \rrbracket =$   
 $\text{null-value}$

Rule  $\text{exec} \llbracket \text{Stmt Stmt}^+ \rrbracket =$   
 $\text{sequential}(\text{exec} \llbracket \text{Stmt} \rrbracket,$   
 $\text{exec} \llbracket \text{Stmt}^+ \rrbracket)$