Syntax

Statement	Syntax	Reason
begin of scope	{	Uses the syntax of the most common modern programming languages - fast to write (one char)
end of scope	}	

while	The while statement executes an Expression and a Statement repeatedly until the value of the Expression is false
	`int a`
	`a := 0`
	`while (expression) {`
	`  statement`
	`}`
		Makes sense in form of english gramma notation and used in most languages today

switch	The switch statement transfers control to one of several statements depending on
	the value of an expression. When one case been executed the switch is exited.
	`switch(expression) {`
	`  case a: statement1`
	`(implicit break)`
	`  case b:  statement2`
	`(implicit break)`
	`  default statement3`
	`(implicit break)`
	`}`
		Switch statements are very useful in the Arduino environment - automatically provides a break as you don't want users to accidentally fall-through.

AND	The conditional-and operator AND evaluates true if both values are true. `true AND true`	Common logical statement - must have

OR	The conditional-or operator OR evaluates true if either of the values are true or false.
	`true OR false`
	`true OR true`
		Common logical statement - must have

else	If the value is true, then the first contained Statement (the one before the else keyword) is executed.
	`if (expression) {`
	`  statement`
	`}`
	`else {`
	`  statement`
	`}`
		Common logical statement - must have