

```
/* control flow */

/* BRANCHES
 * if | elif | else
 */
if condition {
    ...
} elif condition { // [optional] no limits per if, equivalent to ‘else if’
    ...
} else { // [optional] 1 per if
    ...
}

/* LOOPS
 */

/*
 * while loop
 */
loop condition {
    ...
}

/*
 * do while loop
 */
loop true {
    ...
    if condition {
        break
    }
}

/*
 * for each loop
 * it: value of the iteration
 * array: the array being iterated
 * index: [optional] the index of the item in the array
 * a: [optional] increment of ‘index’, default to 1, set to negative for reverse iteration
 * NOTE: what defines an array?
 */
loop it : array ; index, a {
    ...
}

loop item : array ; a {
    ...
}

loop item : array {
    ...
}

/*
 * range loop
 * m: starting value of i
 * n: value of i that breaks the loop
 * a: [optional] incremental value of ‘i’, default to 1, set to negative for reverse iteration to -1
 */
loop i = m : n ; a {
    ...
}

loop i = m : n {
    ...
}

loop i : n {
    ...
}

/*
 * ignore an iteration of a loop
 */
loop true {
    ...
    if condition {
        continue
    }
    ...
}

/*
 * nested loops
 */
#loop LOOP_NAME
loop condition {
    ...
    loop condition {
        ...
        if condition {
            continue // affect the inner loop; same as continue 0
        }
        ...
        if condition {
            break 1 // affect the parent of inner loop and so on
        }
        ...
        if condition {
            break LOOP_NAME // affect the loop of name LOOP_NAME
        }
        ...
    }
    ...
}

/* GOTO
 * NOTE: experimental!!
 */
#line LINE_NAME
...
goto LINE_NAME
```