

start

/ res /
self traceCr: 'start start'.
*res := self **defineBlock**.*
self traceCr: 'start end'.
^ res

Step 1

Context

↑ sender

defineBlock

/ res /
self traceCr: 'defineBlock start'.
*res := self **arg**: [self traceCr: 'block start'.*
1 isZero ifFalse: [^ 33].
self traceCr: 'block end'.].
self traceCr: 'defineBlock end'.
^ res

Step 2

Context

↑ sender

arg: aBlock

/ res /
self traceCr: 'arg start'.
*res := self **executeBlock: aBlock**.*
self traceCr: 'arg end'.
^ res

Step 4

Context

↑ sender

executeBlock: aBlock

/ res /
self traceCr: 'executeBlock: start'.
*res := self executeBlock: **aBlock value**.*
self traceCr: 'executeBlock: loop so should never print that one'.
^ res

Step 5

Context

Lines status

italic: already executed

bold: current execution

plain: future

Step 8

Step 3

homeContext

[self traceCr: 'block start'.
1 isZero ifFalse: [^ 33].
self traceCr: 'block end'.]

returns to the method that
invoked
the block's home
context method

Step 6

[self traceCr: 'block start'.
1 isZero ifFalse: [^ 33].
self traceCr: 'block end'.]

Step 7