

1

start

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

Lines status
italic: already executed
bold: current execution
plain: future

8

↑ sender

2

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
```

homeContext

3

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

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

↑ sender

4

arg: aBlock

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

↑ sender

5

executeBlock: aBlock

```
/ res /  
self traceCr: 'executeBlock: start'.  
res := self executeBlock: aBlock value. 6  
self traceCr: 'executeBlock: should never get here'.  
^ res
```

6

```
[ self traceCr: 'block start'.  
  1 isZero ifFalse: [ ^ 33 ]. 7  
  self traceCr: 'block end'. ]
```

7