Parentheses Vs. Square Brackets

Damien Cassou, Stéphane Ducasse and Luc Fabresse

W2S09

( ) vs. [ ]

- ( ) just changes the priority of an execution but the program is executed
- [ ] blocks program execution: the program is NOT executed

Use [ ] for 'Unknown' Execution Occurrence

Use [expressions] when:

- expressions **may not be executed at all (if, and, ...)**
- expressions **may be executed multiple times (while, ...)**

Example

n timesRepeat: [self doSomething]

timesRepeat: executes a number of times its argument, therefore the argument is a block

( ) vs. [ ] Example

```
x isNil ifTrue: [ self doSomething ]
```

ifTrue: may execute or not its argument, therefore the argument is a block

( ) vs. [ ] Example

[self start] whileTrue: [self doSomething]

whileTrue: may execute both its receiver and argument
multiple times, therefore they are both a block

Quiz

1 to: n do: ... self doSomething ...

x ifEmpty: ... self doSomething ...

Quiz

```
1 to: 100 do: [ :i | self doSomething ]
```

```
x ifEmpty: [ self doSomething ]
```

Summary

- ( ) is about changing the order of a computation
- [ ] is freezing the computation and controlling it

A course by

and

in collaboration with