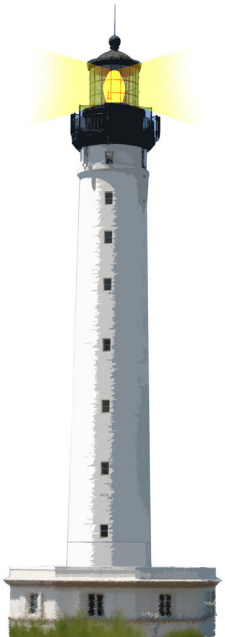
Booleans and Conditions

Damien Cassou, Stéphane Ducasse and Luc Fabresse

W2S08

Pharo

http://www.pharo.org



Booleans

- true is the unique instance of class True
- false is the unique instance of class False

In Pharo, booleans have nothing special

- & | not
- or: and: (lazy)
- xor:
- ifTrue:ifFalse:
- ifFalse:ifTrue:
- ...

Eager and Lazy Logical Operators

```
false & (1 error: 'crazy')
—> an error
```

- the argument (1 error: 'crazy') is executed because this is a non lazy operator

```
false and: [ 1 error: 'crazy' ]
—> false "no error!"
```

- the argument [1 error: 'crazy'] is not executed because it is not necessary

Conditionals

In Pharo, traditional conditional (if, else, while) are messages sent to boolean or block objects

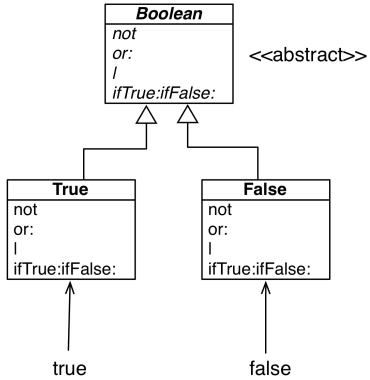
Yes ifTrue:ifFalse: is a message!

```
Weather isRaining
  ifTrue: [ self takeMyUmbrella ]
  ifFalse: [ self takeMySunglasses ]
```

- Conceptually ifTrue:ifFalse: is a message sent to an object: a boolean!
- Heavily optimised by the compiler

Boolean Implementation

- true is the unique instance of the class True
- false is the unique instance of the class False



More details in a future lecture (The Essence of Dispatch)

Conditionals: ifTrue: and ifTrue:ifFalse:

ifTrue: [ ] and ifTrue: [ ] ifFalse: [ ] are two different messages

```
forceItalicOrOblique
  self slantValue = 0
    ifTrue: [ slantValue := 1 ]
```

```
fullName isEmptyOrNil
  ifTrue: [ 'FirstnameLastname' translated ]
  ifFalse: [ fullName ].
```

Conditionals: ifFalse: and ifFalse:ifTrue:

ifFalse: [ ] and ifFalse: [ ] ifTrue: [ ] are two different messages

Conditionals: ifEmpty: ifNotEmpty:

```
myProtocol
  ifEmpty: [ 'As yet unclassified' ]
```

```
self listItems
  ifNotEmpty: [ :aList | aList at: index ]
```

- Notice that when the receiver is not empty we get it as argument
- No need to ask it again

Summary

- Booleans are real objects
- Some conditionals are messages sent to Booleans

A course by

and

in collaboration with