

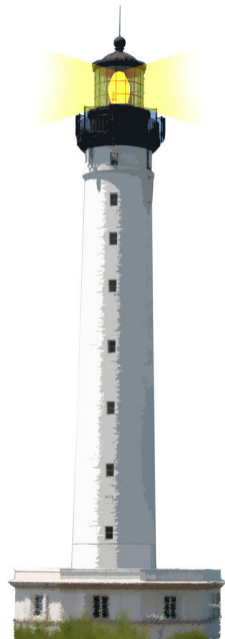
# Understanding Return

Damien Cassou, Stéphane Ducasse and Luc Fabresse

W3S11



<http://www.pharo.org>



# What You Will Learn

- How to return a value from a method and a block
- The default return values



# 4 Cases

- Method with a return statement
- Method without a return statement
- Block without a return statement
- Block with a return statement



# Returning a Value From a Method

Use the caret `^` to return a value from a method

```
Number >> squared
```

```
"Answer the receiver multiplied by itself."
```

```
^ self * self
```

# Default Method's Return Value

A method with no caret ^ returns self

```
Game >> initializePlayers
self players
  at: 'tileAction'
  put: ...
```

is equivalent to

```
Game >> initializePlayers
self players
  at: 'tileAction'
  put: ...
  ^ self    "<-- optional"
```

# Blocks Return Value

Blocks return the value of their last expression

```
[ :x |  
  x + 33.  
  x + 2 ] value: 5  
> 7
```

The caret ^ in a block has a special meaning...

# A Caret in a Block Returns from the Method

A caret `^` in a block quits the enclosing method

```
Integer>>factorial
"Answer the factorial of the receiver."

self = 0 ifTrue: [ ^ 1 ].
self > 0 ifTrue: [ ^ self * (self - 1) factorial ].
self error: 'Not valid for negative integers'
```

- When returning (with caret `^`) from a block, the method defining the block is terminated
- Further readings: <http://deepintopharo.org>



# What you Should Know

- The caret `^` always terminates the method
- A method returns `self` by default
- A block returns the result of its last expression





A course by



and



in collaboration with



Inria 2016

Except where otherwise noted, this work is licensed under CC BY-NC-ND 3.0 France

<https://creativecommons.org/licenses/by-nc-nd/3.0/fr/>