

# Issue 318 Documentation

version 0.0

**Roberto Alsina**

April 29, 2018



# Contents

<a href="#">Welcome to Issue 318's documentation!</a>	<a href="#">1</a>
<a href="#">Indices and tables</a>	<a href="#">1</a>
<a href="#">Index</a>	<a href="#">3</a>
<a href="#">Python Module Index</a>	<a href="#">5</a>



# Welcome to Issue 318's documentation!

Contents:

`bool` `namespaced::theClass::method(int arg1, std::string arg2)`  
Describes a method with parameters and types.

`bool` `namespaced::theClass::method2(arg1, arg2)`  
Describes a method without types.

`const T &array::operator[] (int index) const`  
Describes the constant indexing operator of a templated array.

`operator bool () const`  
Describe a casting operator here.

`std::string theClass::name`

`type theClass::const_iterator`

`format_exception(etype, value, tb[, limit=None])`  
Format the exception with a traceback.

**Parameters:**

- **etype** – exception type
- **value** – exception value
- **tb** – traceback object
- **limit** (*integer or None*) – maximum number of stack frames to show

**Return type:** list of strings

`parrot.spam(eggs)`  
`parrot.ham(eggs)`  
Spam or ham the foo.

## Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)



# Index

## A

[array::operator\[\]](#) (C++ function)

## B

### built-in function

[format\\_exception\(\)](#)

## F

### format\_exception()

built-in function

## H

[ham\(\)](#) (in module [parrot](#))

## M

### module

[parrot](#)

## N

[namespaced::theclass::method](#) (C++ function)

[namespaced::theclass::method2](#) (C++ function)

## O

[operator bool](#) (C++ function)

## P

### parrot

module

## S

[spam\(\)](#) (in module [parrot](#))

## T

[theclass::const\\_iterator](#) (C++ type)

[theclass::name](#) (C++ member)



# Python Module Index

## p

### **parrot (Unix, Windows)**

Analyze and reanimate dead parrots.