

# Erudite output test

Mariano Montone

April 12, 2015



# Contents

0.1	Introduction . . . . .	1
0.2	Chunks . . . . .	1
0.3	Extracts . . . . .	1
0.4	Includes . . . . .	1
0.5	Ignore . . . . .	2
0.6	Erudite syntax . . . . .	2
0.6.1	Subsection . . . . .	2
0.6.2	Subsubsection . . . . .	2
0.6.3	Verbatim . . . . .	2
0.6.4	Code . . . . .	2
0.6.5	List . . . . .	2
0.6.6	Emphasis . . . . .	2
0.6.7	Inline verbatim . . . . .	2
0.6.8	Link . . . . .	2
0.6.9	Reference . . . . .	2
0.6.10	Label and index . . . . .	3
1	<b>Index</b>	<b>5</b>



## 0.1 Introduction

This is a test of Erudite output rendering, commands and syntax elements

## 0.2 Chunks

This is a good chunk

```
<<<chunk1>>>
```

This is a good chunk

```
<<<chunk2>>>
```

This is the chunk:

```
<<chunk2>>=
```

```
(+ 1 1)
```

```
<<chunk4>>=
```

```
(print "Start")
```

The end

```
<<<chunk4>>>
```

This is the factorial function:

```
(defun factorial (n)
```

```
  (if (<= n 1)
```

```
<<<base-case>>>
```

```
<<<recursive-case>>>
```

```
  ))
```

The base case is simple, just check for  $n=1$  less:

```
<<base-case>>=
```

```
  1
```

The recursive step is  $n \times n - 1$ :

```
<<recursive-case>>=
```

```
  (* n (factorial (1- n)))
```

## 0.3 Extracts

Extract test This has been extracted

```
(+ 1 2)
```

Extract 3

Start End

## 0.4 Includes

Include test This is includeA

```
(print "include A")  
(print "Include")
```

This is includeB

```
(print "include B")
```

## 0.5 Ignore

## 0.6 Erudite syntax

### 0.6.1 Subsection

### 0.6.2 Subsubsection

### 0.6.3 Verbatim

This is in verbatim

### 0.6.4 Code

```
(defun hello-world ()  
(print "Hello world"))
```

### 0.6.5 List

- First item
- Second item

### 0.6.6 Emphasis

*This is emphasized*

**This is in bold**

*This is in italics*

### 0.6.7 Inline verbatim

This is in inline verbatim

### 0.6.8 Link

Erudite

### 0.6.9 Reference

hello-world

### 0.6.10 Label and index





# 1 Index



# Index

factorial, 1

label-test, 3