

Asciidoctor LoFTE

# Table of Contents

Table of Contents .....	ii
List of Figures.....	iii
List of Tables.....	iv
List of Examples .....	v
List of Listings .....	vi
Content .....	1

# List of Figures

Figure 1. Figurer .....	1
-------------------------	---

# List of Tables

Table 1. Tabler ..... 1

# List of Examples

Example 1. Exempler..... 1

# List of Listings

Listing 1. Ruby Example .....	2
-------------------------------	---

# Chapter 1. Content

Didn't the LoF, LoT, LoE, and LoL look nice?

Build command:

```
asciidoctor-pdf -r ./asciidoctor-pdf-lofte.rb -a pdf-theme=runningcontent -a pdf-themesdir=resources/themes -a allow-uri-read asciidoctor-pdf-lofte-example.adoc
```

On with content:

- [Table 1](#) shows a table example.
- [Figure 1](#) shows a figure example.
- [Example 1](#) shows an example.
- [Listing 1](#) shows a listing example.



*Figure 1. Figurer*

*Table 1. Tabler*

Header 1	Header 2
Body 1	Body 2

*Example 1. Exempler*

```
puts "Hello World"
```

### Listing 1. Ruby Example

```
## The caption (signifier and counter) is inserted on each line, for example "Figure
1."
class FormatTOC < AsciiDoctor::Converter.for('pdf')
  register_for 'pdf'

  def ink_toc_level entries, num_levels, dot_leader, num_front_matter_pages
    # NOTE: font options aren't always reliable, so store size separately
    toc_font_info = theme_font :toc do
      { font: font, size: @font_size }
    end
    hanging_indent = @theme.toc_hanging_indent
    entries.each do |entry|
      next if (num_levels_for_entry = (entry.attr 'toclevels', num_levels).to_i) <
        (entry_level = entry.level + 1).pred ||
        ((entry.option? 'notitle') && entry == entry.document.last_child && entry
        .empty?)
    end
  end
end
```