

How To Use Scribble to Write your Academic Papers: A Reference Tutorial

Morgan Lemmer-Webber

March 16, 2018

Abstract

This tutorial gives examples of the common functions and formatting tags you will need to write an academic paper using Dr. Racket and scribble. This document should be used as a reference guide when you are ready to set up your own academic paper.

This document itself is written in Scribble, and will be most useful to you if you look at both the scribble files (HowTo.scrbl, Basics.scrbl, DocumentStructure.scrbl, Lists.scrbl, Tags.scrbl, Links.scrbl, Tables.scrbl, Citations.scrbl, Images.scrbl, Export.scrbl) and the pdf (HowTo.pdf) or html output (HowTo.html) at the same time. You can find these documents on our git repository <https://github.com/mlemmer/DigitalHumanities>.

Both Racket and Scribble have wonderful documentation and tutorials available, see below for useful references. Often times if you are stuck and can't figure out how to write a function or debug your code on your own you can find an answer by simply googling it (be sure to include some combination of Racket/Scribble in your search because there are a lot of programming languages out there). Racket also has a thriving community of programmers, so you can post a question on the Racket Users forum.

Introduction to Racket with Pictures: <https://docs.racket-lang.org/quick/index.html>

Web Applications in Racket: <https://docs.racket-lang.org/continue/index.html>

Racket Guide: <https://docs.racket-lang.org/guide/index.html>

Racket Reference: <https://docs.racket-lang.org/reference/index.html>

Racket Documentation: <https://docs.racket-lang.org/continue/index.html>

Scribble tutorial: "<https://docs.racket-lang.org/scribble/getting-started.html>"

Scribble Manual: <https://docs.racket-lang.org/scribble/>

1 License

The documentation and content of this tutorial are licensed under the Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0)

All code for this project is licensed under the GNU Lesser General Public License version 3.0 (LGPL) or (at your option) any later version of the GNU LGPL as published by the Free Software Foundation.

2 Basics of the Scribble/Racket language

The first line of your document should indicate the language you are using `"#lang scribble/doc"`. There should be nothing else on that line. The next element on your document should be any libraries you will

require. Many of the functions covered here, including bibliography and footnotes, require individual libraries. If you copy these first few lines from this document, you should be able to run any of the functions I demonstrate here.

Scribble allows us to use Dr. Racket as a text editor, so when your language is set to "`#lang scribble/doc`", it is in "text mode." In programming terms, everything you type is by a string by default. A string, or string of characters, is how text appears in a computer program. The `@` symbol indicates that you are calling a function rather than typing text. Sometimes, you will need to use strings when in this programming mode, in which case you wrap the words in quotation marks. You can use these functions to format your text. Some of these functions can be modified by arguments (such as changing your list from bullet points to numbers, or increasing the space between columns in a table). If you need to use the symbols used for functions in your text, such as `@ # { } []` use the function:

```
@literal|{ }|
```

2.1 Spell Check

You can turn on the spell check by going to "Edit" -> "Spell Check Text (between {} in Scribble)" or by typing Shift+Ctrl+T. In this setting, the majority of your text will appear green in the text editor and words that are not in the standard dictionary appear black (all will appear black once the file is exported). To get spelling suggestions on a word, put your mouse cursor within the word and hit Shift+CTRL+k.

2.2 Margin notes and comments

See section 3.1.2 'margin-note' in the Racket Documentation

`@margin-note{}` places a note to the side

```
@margin-note{like this}
```

like this

Comments: Since scribble is a markup language, you can include comments in the `.scrbl` file that do not render when the file is exported. Just type `@;` and put curly brackets around the text you want hidden. This allows you to leave notes to yourself without publishing them.

```
Now you see it ... @;{Now you don't}
```

Now you see it ...

3 Document Structure

See section 3.1.1 in the Scribble Guide

Scribble makes it easy to structure your document consistently. There are pre-formatted tags for standard aspects of a document. Like the other tags demonstrated in this tutorial, Scribble has coded these tags to map to the equivalent html and LaTeX tags so that the document can then be exported with its formatting intact. Further customization can then be applied with CSS or LaTeX at the export stage

- `@title` designates the title. The text appears in large, bold font (the equivalent of `<title>` in html).
- `@author` designates the author(s) of the document and is typically placed after the title.
- `@abstract` formats your abstract in blockquote format, typically placed after the author.

- @section can be used to separate the document into sections or chapters. This automatically numbers the section (and re-numbers it if you edit or re-arrange your document), and uses a bold font that is smaller than the title, but larger than the text (the equivalent of <h3> in html). Sections can be further broken down into @subsection (the equivalent of <h4> in html) and @subsubsection (the equivalent of <h5> in html) which will continue to be numbered and formatted.
- @subsubsub*section is formatted the same as subsubsection but is unnumbered.

3.1 Section Heading

3.1.1 First Subsection

Second Subsection

Unnumbered Subsection

3.2 Combining Multiple Documents

See section 1.3 in the Scribble Guide

With large documents, such as this tutorial or a dissertation, containing everything within a single document can become cumbersome. Scribble allows you to create multiple documents that will export into a single pdf or a set of linked html files.

For this you set one document as the master for the project (See `HowTo.scrbl` for an example). This will include your main title page, abstract, and perhaps introduction. After the content you wish to include on the master document, you will then use the command `@include-section["filename.scrbl"]` to make a list of the subsequent files you wish to include. These will appear in the document in the order that you list the files.

It will look something like this:

```
@include-section["Basics.scrbl"]
@include-section["DocumentStructure.scrbl"]
@include-section["Lists.scrbl"]
@include-section["Tags.scrbl"]
@include-section["Links.scrbl"]
@include-section["Tables.scrbl"]
@include-section["Citations.scrbl"]
@include-section["Notes.scrbl"]
@include-section["Images.scrbl"]
```

When you combine documents like this, the nested structure of title, section, subsection etc. will all move down one for the subordinate files. At the top of this document (`DocumentStructure.scrbl`), I used `@title{Document Structure}`. If you were to export this file on its own, that would be formatted as a title and all of the sections would nest down accordingly. However if I export `HowTo.scrbl`, that title will appear as a section heading and all of the subsequent sections would be nested another layer in.

4 Exporting From Scribble

See section 1.1 in the Scribble Guide

Exporting simple html from Scribble is easy. Dr. Racket has a built in exporter, so you simply hit the "Scribble HTML" button at the top of the page. This generates an html file of the same filename saved in

the same directory as the original scribble file and opens the html file in your browser. I recommend that you use that simple export option often when you're writing to make sure that everything works (if you export and get an error message but you've only added five tags or functions since the last time you exported successfully, that's only five things you need to debug to find the problem).

In order to export to pdf, tex, or to export a multi-page document, you will need to use the command line. The commands below apply out of the box to OSX and GNU/Linux. If you are using Windows, you can install Cygwin, which will allow you to use a terminal with GNU/Linux commands.

4.1 Exporting to HTML via the Command Line

All required files including images need to be stored in the same directory or folder. Scribble has base CSS and LaTeX templates that they generate from, but if you wish to export with custom CSS or LaTeX files, those should be stored in the same directory as well. In a terminal, use the cd command to change to the directory where your files are stored.

```
cd Desktop/Documents/Racket/InstructionMaterials/
```

To export the same simple html file as the "Scribble HTML" button generates, type the command 'scribble' (i.e. the program that you're using) + the filename

```
mlemmer@rooibos:~/Desktop/Documents/Racket/InstructionMaterials$  
scribble HowTo.scrbl
```

If it successfully exported, it will return the message

```
[Output to HowTo.html]
```

If you are using multiple files to create one document and wish to have separate html files (if you want to be able to click from one page to the next etc), type the command 'scribble' then '-htmls' (to indicate you want multiple html subsections) + the filename

```
mlemmer@rooibos:~/Desktop/Documents/Racket/InstructionMaterials$  
scribble --htmls HowTo.scrbl
```

If it successfully exported, it will return the message

```
[Output to HowTo/index.html]
```

This will create a new file or directory with multiple html files. If you open the index file in your browser, you'll be able to navigate through all of the files in the directory.

4.2 Exporting to pdf via the Command Line

Scribble uses LaTeX to export to pdf, so you will need to first install LaTeX and any applicable packages.

To export to pdf from the command line, type the command 'scribble' then '-pdf' (to indicate that you want the output to be pdf) + the filename

```
mlemmer@rooibos:~/Desktop/Documents/Racket/InstructionMaterials$  
scribble --pdf HowTo.scrbl
```

LaTeX is notoriously fussy and it is likely that you will get some long and confusing error messages whether or not your document successfully exported. If it successfully exported, somewhere in the message it will say

[Output to HowTo.pdf]

If you want to modify the pdf output via a LaTeX template, first create a basic template. Keep in mind that Scribble has a default template and you don't need to re-create everything, and if your template is too involved it will cause an error message. It is fairly straightforward to add a template that indicates font size, margins, and line spacing (such as `LatexHeader.tex`, which we've included here). In the command line, type the command `'scribble'` then `'-prefix LatexFilename.tex'` (to indicate which LaTeX file you want to modify your output) then `'-pdf'` (to indicate that you want the output to be pdf) + the filename

```
mlemmer@rooibos:~/Desktop/Documents/Racket/InstructionMaterials$  
scribble --prefix LatexHeader.tex --pdf HowTo.scrbl
```

If it successfully exported, somewhere in the message it will say

[Output to HowTo.pdf]

4.3 Exporting to LaTeX via the Command Line

If you want to export to a LaTeX file, type the command `'scribble'` then `'-latex'` (to indicate that you want the output to be a LaTeX file) + the filename.

```
mlemmer@rooibos:~/Desktop/Documents/Racket/InstructionMaterials$  
scribble --latex HowTo.scrbl
```

If it successfully exported, it will return the message

[Output to HowTo.tex]

4.4 Editing the Exported Files

This tutorial focuses on ways to format your documents using Scribble that you can then export to both html and pdf. For this, scribble acts as the middle man, bridging the syntax of the two markup languages so that you don't have to do double the work. However, even so there are bound to be some glitches in the output that you would like to tweak. Since your output via the command line can be `.html` or `.tex` files, you can do most of the leg work using scribble, then manually clean up the last parts that are not formatting properly based off of those files if necessary.

5 Lists

See section 3.1.2 'itemlist' in the Scribble guide

`@itemlist` creates indented bullet points for lists

- First bullet point
- Second bullet point

`@itemlist[#:style'ordered]` creates numbered lists

1. First bullet point
2. Second bullet point

6 Common Formatting Tags

See section 3.1.3 in the Scribble Guide

- `@smaller` renders the font in a smaller size. These can be nested to decrease incrementally
- `@larger` renders the font in a larger size. These can be nested to increase incrementally
- `@italic` can be used for book titles or words in foreign languages.
- `@emph` can be used instead of italic.
- `@bold` can be used for emphasis.
- `---` turns into an em dash (—)
 - `@centered` centers the text
- `@subscript` can be used to create subscripts
- `@superscript` can be used to create superscripts

All of these formatting functions can be nested.

`@nested[#:style 'inset]` can be used to place text in blockquotes

See section 3.1.2 in the Scribble Guide

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur in egestas nulla, eu sodales purus. Quisque quis mi eleifend, malesuada libero a, tempor diam. Phasellus accumsan neque faucibus ornare rutrum. Phasellus ac nulla finibus, blandit mi sit amet, condimentum ante. Mauris dapibus ornare hendrerit. Cras eget neque ut nibh semper rhoncus. Pellentesque in orci quam. Nunc est ligula, tempor eget cursus quis, tempus vel dui. Vivamus vestibulum enim a elit commodo sagittis. Mauris scelerisque dui eu aliquet pretium. In ut augue eget enim tristique accumsan sit amet a enim. Sed sed ullamcorper risus.

7 Links

See section 3.1.6 in the Scribble Guide Embedding a hyperlink requires three components. You need to call the `@hyperlink` function, you need to place the link in quotes between brackets [""], and you need to place the text you want visible in curly brackets {}.

So if you type:

```
@hyperlink["https://youtu.be/LNCC6ZYI3SI"]{What sound does a rhinoceros make?}
```

It will export as: What sound does a rhinoceros make?

8 Tables

See section 3.1.2 'tabular' in the Scribble Guide

@tabular allows you to create tables from a list of lists.

```
@tabular[#:sep @hspace[1]
(list (list @bold{Animal} @bold{Sound})
      (list "cat" "meow")
      (list "dog" "bark")
      (list "duck" "quack")
      (list "rhinoceros" "squee")))]
```

Animal	Sound
cat	meow
dog	bark
duck	quack
rhinoceros	squee

In Scribble, #: indicates that there is a keyword-value argument. These are variables that will modify how your final product renders. The argument #:sep @hspace[] in this table is a variable that controls the amount of space between columns, so increasing that number increases the space between columns. You can also include other features/formatting into your table

```
@tabular[#:sep @hspace[5]
(list (list @bold{Animal} @bold{Sound})
      (list "cat" "meow")
      (list @italic{dog} "bark")
      (list "duck" "quack")
      (list "rhinoceros" @hyperlink["https://youtu.be/LNCC6ZYI3SI"]{squee})))]
```

Animal	Sound
cat	meow
<i>dog</i>	bark
duck	quack
rhinoceros	squee

9 Citations and Footnotes

9.1 Citations

See section 4.2.9 in the Scribble Guide

The formatting for the bibliography in scribble may look intimidating, but really you are filling in the same general fields you would in any citation manager. This feature requires access to the "[scrib-lib/autobib](#)" library, so include that in the list of required libraries at the top of your page. Citations use keyword arguments "(#:title #:url etc)" that allow you to construct your citation with the components you need. For ease of data entry, you can just copy and paste the from the templates below for each citation. However, if you do so remember to **remove any keyword-arguments you do not use** because a blank argument will result in an error message.

Basic components of a citation:

```
@(define bib-Example2018
(make-bib
#:title "Example Book Title"
#:author (authors "Author One" "Author Two")
#:is-book? yes
#:date 2018
#:location (book-location #:publisher "City:Publisher")
#:url "exampleurl.com"
#:note "This is where you would annotate your citation"))
```

this is a nuts and bolts example of a citation entry with the types of information required in each field. I will now go through and cut that entry into pieces to explain each of the components you need to create your bibliography. Following that I will provide a template and examples of the main types of bibliographic entry.

```
(require scriblib/autobib)
```

Your bibliography will draw on the "`scriblib library`", so it is necessary to include it in the libraries called at the top of your page.

```
@(define-cite ~cite citet-id generate-bibliography #:style author+date-style)
```

This function is what allows scribble to place citations in-line in your document and generate a bibliography from those citations. It also determines what type of citations you will have using the keyword-argument `#:style`. Out of the box, scribble only has two citation options: `author+date-style` (Flatt and PLT 2010) and `number-style` (Flatt and PLT 2010). However, you can use a BibTeX file to modify the LaTeX output to any citation style you want. You only need to define this function once for the whole bibliography.

```
define bib-NameDate
(make-bib
)
```

This has two components. You are essentially defining a new function for each citation. You can use whatever naming convention you want but for your own sanity, stay consistent. For this tutorial, I'm defining each citation with the prefix `bib-` and the author(s) last name and the date (`bib-Barber1994`). The second component (`make-bib`) is what defines this function as a citation.

```
#:title "Title of Book Here"
```

This section is for the title of the book or article. This argument requires a string, so place the title in quotation marks. This can lead to some formatting problems when quotation marks are used in the code *and* you are creating an entry for an article and the title needs to be in parentheses. Some BibTeX styles may automatically fix this. If not, use a method called string escaping by typing forward-slash before the set of quotation marks you wish to be visible (`"This is a string." "\The Evolution of String Theory\`"`)

```
#:author (authors "Author One" "Author Two")
```

Since there can be multiple authors, this argument requires a list of strings. Place the author(s) name(s) in quotation marks within the parentheses after `authors`. If there are multiple authors, leave only a space between the two strings (`authors "Carole Gillis" "Marie-Louise Nosch"`). The program recognizes the structure of `"firstname lastname"` in these strings and separates them automatically for whatever citation format you use. If the author name does not fit in this format, use `@literal` instead of parentheses. Ex: (`authors @literal{Cato the Elder}|`), (`authors @literal{Kelsey Museum of Archaeology}|`)


```
#:date "2018"
```

This argument requires a string, so place the date in quotation marks.

```
#:is-book? "yes"
```

This argument italicizes the title component of the citation (as opposed to journal articles, sections of books, or tech reports). **Only use this argument if the answer is yes.** This argument requires a string, so place yes in quotation marks.

```
#:url "exampleurl.com"
```

This argument takes a string, place the url in quotation marks.

```
#:note "This book was really useful..."
```

You can annotate your bibliography with this argument. While most of the strings we've used so far are short, in this context your string can be as long as a paragraph.

```
#:location (citation keyword [])
```

The location field is where you distinguish between citation types. Below are citation types and the additional keyword-arguments that each takes in the location portion of the above template.

9.1.1 Book

```
#:location (book-location #:edition #:publisher )
```

Both of these arguments require strings. For publisher, enter both fields "city: publisher". When a keyword-argument takes multiple subsequent arguments, you can either place them in-line with a single space between as shown here, or return once between arguments.

Template:

```
@(define bib-NameDate
  (make-bib
    #:title
    #:author (authors )
    #:is-book?
    #:date
    #:location (book-location #:edition #:publisher )
    #:url
    #:note ))
```

Example (single author):

```
@(define bib-Barber1994
  (make-bib
    #:title "Women's Work: The First 20,000 Years: Women, Cloth, and Society
in Early Times."
    #:author (authors "E.J.W Barber")
    #:is-book? "yes"
```

```

#:date "1994"
#:location (book-location #:publisher "New York: Norton")
#:note "In this seminal book, Barber looks at the association between
women and textile production spanning the time period from 20,000-
500 BCE in various cultures in the Mediterranean, Egypt, and Near East.
While Barber does not cover Roman evidence, this book is a foundational
resource for how to approach the topic of textile production as a gendered
task.")

```

Example (multiple authors):

```

@(define bib-GillisNosch2007
  (make-bib
    #:title      "Ancient Textiles. Production, Craft and Society. Proceedings
of the First International Conference On Ancient Textiles. Lund, Sweden, and
Copenhagen, Denmark, On March 19-23 2003"
    #:author     (authors "Carole Gillis" "Marie-Louise Nosch")
    #:is-book?   "yes"
    #:date       "2007"
    #:location   (book-location #:publisher "Oxford: Oxbow")))

```

9.1.2 Journal Article

```

#:location (journal-location "JournalTitle"
  #:pages #:number #:volume)

```

The journal-location argument automatically takes an argument of the title of the journal as a string. The pages argument requires a numerical range in the form of a list. Create this by putting the two numbers in parentheses separated by a single space and place a single quote before the parentheses '(1 10)

Template:

```

@(define bib-NameDate
  (make-bib
    #:title
    #:author (authors )
    #:date
    #:location (journal-location "JournalTitle" #:pages #:number #:volume)
    #:url
    #:note ))

```

Example:

```

@(define bib-Cottica2006
  (make-bib
    #:title      "\"The Symbolism of Spinning in Classical Art and Society\""
    #:author     (authors "Daniela Cottica")
    #:date       "2006"
    #:location   (journal-location "Cosmos" #:pages '(185 209) #:volume "20")
    #:note       "In this brief article, Cottica summarizes the association

```

between women and spinning in the Greek and Roman cultures. She approaches the topic thematically, covering religious associations, everyday life, and funerary topics."))

9.1.3 Conference Proceedings

```
#:location (proceedings-location "ProceedingsTitle" #:pages #:number #:volume)
```

This format can be used for conference proceedings or for chapters or sections of books. The proceedings-location argument automatically takes an argument of the title of the proceedings/book as a string. The pages argument requires a numerical range in the form of a list. Create this by putting the two numbers in parentheses separated by a single space and place a single quote before the parentheses '(1 10)

Template:

```
@(define bib-NameDate
  (make-bib
    #:title
    #:author (authors )
    #:date
    #:location (proceedings-location "ProceedingsTitle" #:pages
                                     #:number #:volume)
    #:url
    #:note ))
```

Example:

```
@(define bib-Trinkl2007
  (make-bib
    #:title      "\"Artifacts Related to Preparation of Wool and Textile
Processing Found Inside the Terrace Houses of Ephesus, Turkey\""
    #:author     (authors "Elisabeth Trinkl")
    #:date       "2007"
    #:location   (proceedings-location "Ancient Textiles. Production, Craft and
Society. Proceedings of the First International Conference On Ancient Textiles.
Lund, Sweden, and Copenhagen, Denmark, On March 19-23 2003" #:pages '(80 86)
#:volume "1")))
```

9.1.4 Dissertation/Thesis

```
#:location (dissertation-location #:institution #:degree)
```

Both of these arguments require strings, enter the answers in quotation marks.

Template:

```
@(define bib-NameDate
  (make-bib
    #:title
    #:author (authors )
```

```

#:is-book?
#:date
#:location (dissertation-location #:institution #:degree)
#:url
#:note ))

```

Example:

```

@(define bib-LarssonLoven2002
  (make-bib
    #:title      "The Imagery of Textile Making. Gender and Status in the
    Funerary Iconography of Textile Manufacture in Roman Italy and Gaul"
    #:is-book?   "yes"
    #:author     (authors "Lena Larsson Lovén")
    #:date       "2002"
    #:location   (dissertation-location #:institution "University of Gothenburg"
    #:degree     "PhD")

    #:note "Larsson Lovén analyzes funerary iconography related to textile
    production from Italy and Roman Gaul. Textile imagery was found on both male
    and female funerary monuments and she analyzes the gendered implications
    of how this imagery differs between the two.))

```

9.1.5 Tech Report

```

#:location (techrpt-location #:institution #:number)

```

Template:

```

@(define bib-NameDate
  (make-bib
    #:title
    #:author (authors )
    #:date
    #:location (techrpt-location #:institution #:number)
    #:url
    #:note ))

```

Example:

```

@(define bib-Flatt2010
  (make-bib
    #:title      "Reference: Racket"
    #:author     (authors "Matthew Flatt" "PLT")
    #:date       "2010"
    #:location   (techrpt-location #:institution "PLT Inc."
    #:number     "PLT-TR-2010-1")
    #:url        "http://racket-lang.org/tr1/"))

```

If you are following along and testing this as you go, you've now got several functions defined for specific citations. However, if you were to export these citations to either html or pdf, nothing would appear because so far we've created only the back end. Just like most citation management programs, two more steps are required before you get any output. You must cite the source in your text, then you must generate the bibliography.

Use `@~cite[bib-NameDate]` to enter your citations in-line of your text. These will either be author+date-style or number-style depending on which option you determined above. Use `@(generate-bibliography)` to create the full bibliography based on sources you've cited in your text. For this example, I'm just using the citation and bibliography styles that come out of the box with scribble. You can change the citation style and formatting with LaTeX.

```
I am including this brief literary review as an example of
citations. The Racket Reference has been invaluable
in creating this guide on using Racket and Scribble for academic
papers.@~cite[bib-Flatt2010] The Racket Reference is the
only citation here that was used in creating this file, the rest
of the titles here are drawn from my dissertation research.
Elizabeth Barber efficiently breaks down the central role that
women played in the early development of textile production in
her 1994 book Women's Work.@~cite[bib-Barber1994] Daniella
Cottica rather deftly analyzes this topic thematically utilizing
both Greek and Roman evidence. However, the breadth of the topic
exceeds the bounds of an article spanning only 20 pages including
copious images.@~cite[bib-Cottica2006] Lena Larsson Lovén's work
focuses on the symbolic relationship between women and textile
production in the Roman Empire through literary, epigraphic, and
iconographic sources.@~cite[bib-LarssonLoven2002] Conference
proceedings from the International Conference On Ancient
Textiles provide interesting case studies of textile research
@~cite[bib-Trinkl2007] as well as current research methods, testing,
and synthesis between sites.@~cite[bib-GillisNosch2007]
```

```
@(generate-bibliography)
```

Put this function to generate the bibliography wherever you want the bibliography to appear in your paper.

I am including this brief literary review as an example of citations. The *Racket Reference* has been invaluable in creating this guide on using Racket and Scribble for academic papers. (Flatt and PLT 2010) The *Racket Reference* is the only citation here that was used in creating this file, the rest of the titles here are drawn from my dissertation research. Elizabeth Barber efficiently breaks down the central role that women played in the early development of textile production in her 1994 book *Women's Work.* (Barber 1994) Daniella Cottica rather deftly analyzes this topic thematically utilizing both Greek and Roman evidence. However, the breadth of the topic exceeds the bounds of an article spanning only 20 pages including copious images. (Cottica 2006) Lena Larsson Lovén's work focuses on the symbolic relationship between women and textile production in the Roman Empire through literary, epigraphic, and iconographic sources. (Lovén 2002) Conference proceedings from the *International Conference On Ancient Textiles* provide interesting case studies of textile research (Trinkl 2007) as well as current research methods, testing, and synthesis between sites. (Gillis and Nosch 2007)

Bibliography

- E.J.W Barber. *Women's Work: The First 20,000 Years: Women, Cloth, and Society in Early Times*. New York: Norton, 1994. In this seminal book, Barber looks at the association between women and textile production spanning the time period from 20,000-500 BCE in various cultures in the Mediterranean, Egypt, and Near East. While Barber does not cover Roman evidence, this book is a foundational resource for how to approach the topic of textile production as a gendered task.
- Daniela Cottica. "The Symbolism of Spinning in Classical Art and Society". *Cosmos* 20, pp. 185–209, 2006. In this brief article, Cottica summarizes the association between women and spinning in the Greek and Roman cultures. She approaches the topic thematically, covering religious associations, everyday life, and funerary topics.
- Matthew Flatt and PLT. Reference: Racket. PLT Inc., PLT-TR-2010-1, 2010. <http://racket-lang.org/tr1/>
- Carole Gillis and Marie-Louise Nosch. *Ancient Textiles. Production, Craft and Society. Proceedings of the First International Conference On Ancient Textiles. Lund, Sweden, and Copenhagen, Denmark, On March 19-23 2003*. Oxford: Oxbow, 2007.
- Lena Larsson Lovén. *The Imagery of Textile Making. Gender and Status in the Funerary Iconography of Textile Manufacture in Roman Italy and Gaul*. PhD dissertation, University of Gothenburg, 2002. Larsson Lovén analyzes funerary iconography related to textile production from Italy and Roman Gaul. Textile imagery was found on both male and female funerary monuments and she analyzes the gendered implications of how this imagery differs between the two.
- Elisabeth Trinkl. "Artifacts Related to Preparation of Wool and Textile Processing Found Inside the Terrace Houses of Ephesus, Turkey". In *Proc. Ancient Textiles. Production, Craft and Society. Proceedings of the First International Conference On Ancient Textiles. Lund, Sweden, and Copenhagen, Denmark, On March 19-23 2003* volume 1, pp. 80–86, 2007.

9.2 Footnotes

See section 5 in the Scribble Guide

```
"(require scrib/lib/footnote)"
```

Your bibliography will draw on the "scrib/lib/footnote" library, so it is necessary to include it in the libraries called at the top of your page. There is a built-in note function, @note.¹ In your html output, notes made using this function appear to the side of the text. In the pdf output, they appear as footnotes.

If you want your footnotes to appear as endnotes in the html output, you must define the footnote function using @define-footnote. This requires two arguments separated by a space — the first is the tag you will use to place a footnote in-line of your text, the second is the tag you will use to generate your footnotes. You can use whatever naming convention you want. For this tutorial, I'm keeping it simple and using 'footnote' and 'make-footnote'.

```
@define-footnote [footnote make-footnote]
```

You then will call your footnotes by typing @footnote{Whatever text you want in your footnote in-line of your text. Other than html output, @note and @footnote are now interchangeable. If you use both — as I have here — The numbers will sync up in the pdf footnotes even.

You can nest footnotes and citations by typing @footnote{@~cite[bib-NameDate]}. If you have multiple citations in one footnote, you simply call each citation: @footnote{@~cite[bib-NameDate1]; @~cite[bib-NameDate2]}. You can also combine both text and a citation in a footnote: @footnote{A similar representation can be found in @~cite[bib-NameDate]}.

```
I am including this brief literary review as an example of  
footnotes and citations.@footnote{Footnotes can be text only, or you
```

¹You'll notice that in the html this appears on the side and is unnumbered, in the pdf it is a numbered footnote

can include a citation.} The *Racket Reference* has been invaluable in creating this guide on using Racket and Scribble for academic papers. The *Racket Reference* is the only citation here that was used in creating this file, the rest of the titles here are drawn from my dissertation research.} Elizabeth Barber efficiently breaks down the central role that women played in the early development of textile production in her 1994 book *Women's Work*.} Daniella Cottica rather deftly analyzes this topic thematically utilizing both Greek and Roman evidence. However, the breadth of the topic exceeds the bounds of an article spanning only 20 pages including copious images.} Lena Larsson Lovén's work focuses on the symbolic relationship between women and textile production in the Roman Empire through literary, epigraphic, and iconographic sources.} Conference proceedings from the *International Conference On Ancient Textiles* provide interesting case studies of textile research as well as current research methods, testing, and synthesis between sites.}

@make-footnote []

I am including this brief literary review as an example of footnotes and citations.² The *Racket Reference* has been invaluable in creating this guide on using Racket and Scribble for academic papers.³ Elizabeth Barber efficiently breaks down the central role that women played in the early development of textile production in her 1994 book *Women's Work*.⁴ Daniella Cottica rather deftly analyzes this topic thematically utilizing both Greek and Roman evidence. However, the breadth of the topic exceeds the bounds of an article spanning only 20 pages including copious images.⁵ Lena Larsson Lovén's work focuses on the symbolic relationship between women and textile production in the Roman Empire through literary, epigraphic, and iconographic sources.⁶ Conference proceedings from the *International Conference On Ancient Textiles* provide interesting case studies of textile research⁷ as well as current research methods, testing, and synthesis between sites.⁸

10 Images

See section 3.1.4 in the Scribble Guide

Entering images in-line is simple. Use `@image["filename.jpg"]` to insert an image file into your document. The image file must be stored in the same directory/folder as your scribble file. If you want the image centered with your text, you can use `@centered{@image[filename.jpg]}`

²Footnotes can be text only, or you can include a citation.

³ (Flatt and PLT 2010) The *Racket Reference* is the only citation here that was used in creating this file, the rest of the titles here are drawn from my dissertation research.

⁴ (Barber 1994)

⁵ (Cottica 2006)

⁶ (Lovén 2002)

⁷ (Trinkl 2007)

⁸ (Gillis and Nosch 2007)

The image will appear at the size of the image file. The size and format of your image will effect its placement in your document. If the image is too large to fit on the remaining space of the page below the text above it, it will shift to the next page. If you have nice, high-res images, that is great for research, but bad for your pdf or html output:

```
@image["KaranisKM3338_1.JPG"]
```


There are a few ways to modify size. The most stable method is to scale the image using your preferred image manipulation program. If you do this, remember to export the scaled image with a new file name — You don't want to save over your original high-res image, and you want to be able to distinguish between the two files.

```
@centered{@image["KaranisKM3338_1scaled.JPG"]}
```



Another method is to use CSS or LaTeX to scale the images. This requires more effort in coding, but it can be universally applied to all of the images you use so you don't have to manually re-size all 50+ images in your dissertation and you don't need to either scale the original image files or have double copies of all of your images. Setting up a CSS or LaTeX style sheet is outside the bounds of this tutorial, but I will cover how to incorporate it into your scribble file. First you need to first make a CSS/LaTeX stylesheet and store it in the same directory. Then you can define a function to apply the added style.

*This is not pictured in the pdf because the example is for CSS and therefore does not work with pdf export.

Note that this is the same file as at the top of this section, the image file itself has not been altered.

10.1 Figures

See section 2 in the Racket Documentation

In an academic paper, you will likely have more than one image, and perhaps other types of figures such as tables and graphs. Additionally, your images and figures will need to be numbered and tagged in such a way that you can refer to them in your text. For this we use the @figure function.

```
@figure["tag" @elem{Image identification} @image["filename.jpg"]]
```

This function has three primary arguments:

The "tag" is the what you will use to reference the figure when you reference it in-line. This argument requires a string, so enter it in quotation marks. You can use any tag convention you prefer, but each tag must be unique. I am using the Site + Museum number convention here because the artifacts I'm discussing already have unique ID numbers and that's the convention that I've used for naming the image files.

The @elem{} argument is for any text you want visible after the figure number. This will likely include a description of the image, date, culture, materials, photo credits, etc.

The @image["filename.jpg"] argument is to place the image.

```
@figure["KaranisKM3338" @elem{Loom weight from a house in Karanis,  
unfired clay, 1st through 5th centuries CE, Kelsey Museum of Art, KM3338.  
Image by Morgan Lemmer-Webber with permission of the Kelsey Museum of  
Art.}]{@image["KaranisKM3338_1.JPG"]}
```



Figure 1: Loom weight from a house in Karanis, unfired clay, 1st through 5th centuries CE, Kelsey Museum of Art, KM3338. Image by Morgan Lemmer-Webber with permission of the Kelsey Museum of Art.

```
@figure["KaranisKM3352" @elem{Weavers comb from a house in Karanis,  
wood, 1st through 5th centuries CE, Kelsey Museum of Art, KM3352. Image  
by Morgan Lemmer-Webber with permission of the Kelsey Museum of  
Art.}]{@image["KaranisKM3352_1.JPG"]}
```

To reference a figure in-line, use `@Figure-ref[""]` if you want the 'F' capitalized or `@figure-ref[""]` if you want the 'f' lower-case. Since each figure has a unique tag, you can reference a figure multiple times. The figure number that appears in-line will have a link to take you to the figure itself. Unfortunately, since a figure can be referenced multiple times, it's impossible to have a link to reference back to your exact location within the text. Unlike the bibliography, the figure list will render whether or not the figures have been referenced in your text.

The figures themselves will appear wherever in the order you define the figure functions. Depending on your preferences, you can intersperse the images within the text or group them together. The Racket/Scribble code for figures generates the figure numbers based on the order the functions are defined, not the order in which they are referenced in the text. If you edit your paper and end up referencing figures in a different order, you can update the figure numbers by re-organizing the order in which they are listed.

Example:

```
Given the arid conditions of Karanis, many textile implements  
of perishable or delicate materials survive that are rare elsewhere. These  
include loom weights of unfired clay (@Figure-ref["KaranisKM3338"]), or  
wooden weavers combs (@figure-ref["KaranisKM3352"])
```

Given the arid conditions of Karanis, many textile implements of perishable or delicate materials survive that are rare elsewhere. These include loom weights of unfired clay (Figure 1), or wooden weavers combs (figure 2).

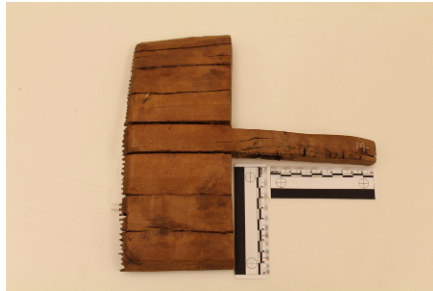


Figure 2: Weavers comb from a house in Karanis, wood, 1st through 5th centuries CE, Kelsey Museum of Art, KM3352. Image by Morgan Lemmer-Webber with permission of the Kelsey Museum of Art.