
ANATOMY OF “ANATOMY OF MELANCHOLY” IN X_YL^AT_EX

*Making a large and complex book with
free/libre software*

Contents

I	Introduction	5
1	Selecting a typeface (Junicode)	6
2	Selecting a paper layout and margins	8
3	Selecting a design style, choices	12
4	Organizing the artwork with biblatex	14
II	Preliminary pages	19
5	Designing a cover with Inkscape	20
6	Designing a diagram for the frontispiece with TikZ	23
7	Including the cover, title pages and frontispiece	26
8	Adding a list of figures and a list of prescriptions	28

<i>CONTENTS</i>	3
III Main Part	31
9 Creating formatting and helpful macros	32
9.1 URLs	34
10 Converting the end notes to a usable format	36
11 Using footnotes or margin notes	38
12 Typesetting figures	40
12.1 Placing figures in the text	40
12.2 Placing figures on their own page	41
13 Designing the synopsis schemata	42
14 Typesetting the author's notes	44
15 Typesetting inline translations	46
16 Finding out uncommon vocabulary for a glossary	48
17 Typesetting verse and block quotations	50
18 Creating an OTF for Chaucer with FontForge	52
IV Appendices	54
19 Creating an index for author and topic references	55
20 Designing a custom biblatex bibliography style to list artwork sources	57

<i>CONTENTS</i>	4
21 Adding an editor's postface	66
22 Closing with a colophon	68

Part I

Introduction

Chapter 1

Selecting a typeface (Junicode)

[Redacted text block]

Chapter 2

Selecting a paper layout and margins



Figure 2.1:

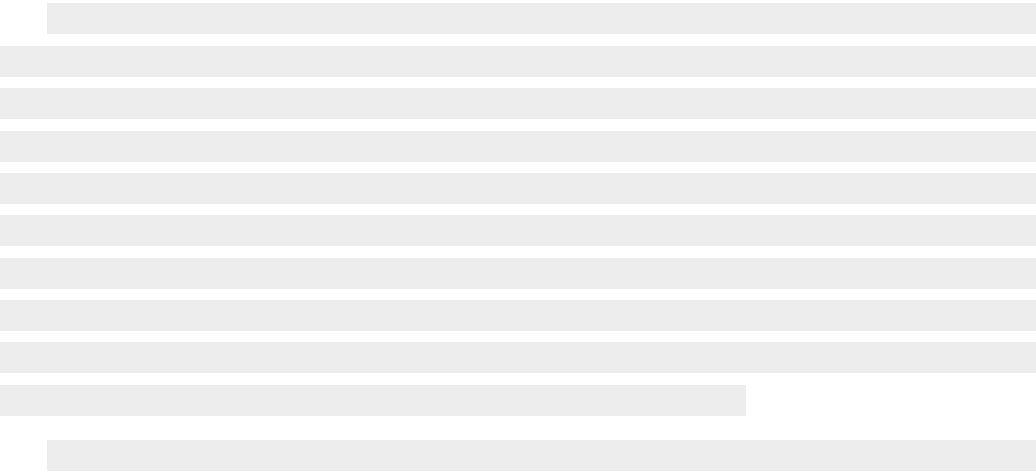
2
3



Figure 2.2:

4

5



6

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]



Chapter 3

Selecting a design style, choices



Chapter 4

Organizing the artwork with biblatex

To simplify the handling of a large amount of artwork in the book, we store each work as a bibliography entry in a bib database and use the bibliography management package [biblatex](#). The bib database is a plain text file with entries of the following format:

```
1 @artwork{TheLoveSong,
2   title      = {The Love Song},
3   annotation = {{Burne-Jones associated this painting with
    ↳ a refrain from a French folk ballad: "Alas, I know a
    ↳ love song, / Sad or happy, each in turn." Cupid, his
    ↳ arrows slung over his shoulder, works the bellows on
    ↳ the portative organ. This picture, which took nine
    ↳ years to complete, unites inspirations that shaped
    ↳ Burne-Jones's art: medievalism, Italian Renaissance
    ↳ painting, romance, beauty, and music. Like a
    ↳ bittersweet melody, the scene suggests a mood of
    ↳ dreamy melancholy. As one critic observed, "There is
    ↳ no story: nothing to guess at, but everything to
    ↳ feel."}},
4   author     = {{Edward Burne-Jones}},
```

```

5   date      = {1868/1877},
6   url       =
   ↪ {https://www.metmuseum.org/art/collection/search/435826},
7   eid       = {47.26},
8   file      =
   ↪ {The-Love-Song-Edward-Burne-Jones-small.jpg},
9   type      = {oil on canvas},
10  }

```

We use the `file` field to store the image filename. Assuming the images are stored in a folder called `figures` in the same directory as the main `TEX` file, we define the default graphics path as

```

1  \graphicspath{./figures/}

```

This means that everytime we include a graphic with the `includegraphics` macro, it will look for it in that prepend the path to whatever we pass to `includegraphics`.

To load the database we insert this line in our main `tex` file:

```

1  \addbibresource{artworks.bib}

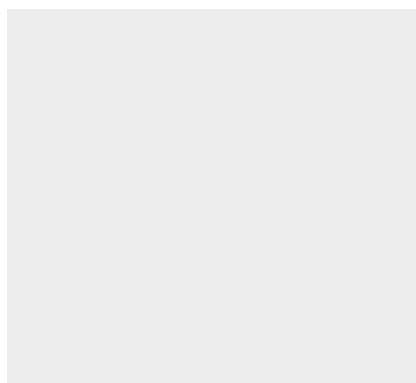
```

Now, using the key for each entry we can use any of the entry fields inside the document. The fields we are interested in are:

- `author`
- `title`
- `year`
- `type`, which contains the art medium

- file

biblatex defines macros to access the fields using the entry key.



8

Figure 4.1:

9



10

[REDACTED]

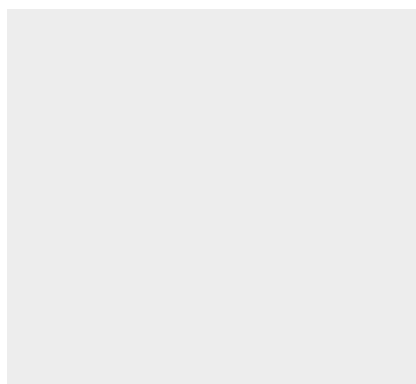


Part II

Preliminary pages

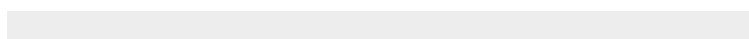
Chapter 5

Designing a cover with Inkscape



11

Figure 5.1:



12



13





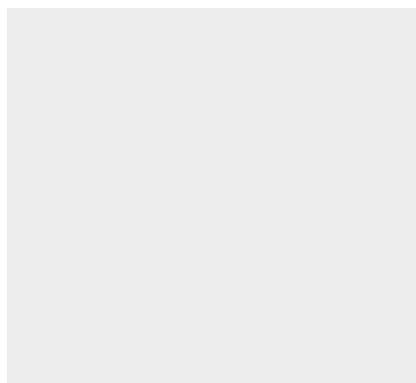
14

Figure 5.2:

15

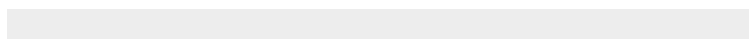
Chapter 6

Designing a diagram for the frontispiece with TikZ



16

Figure 6.1:



17



18

CHAPTER 6. DESIGNING A DIAGRAM FOR THE FRONTISPIECE WITH TIKZ24



CHAPTER 6. DESIGNING A DIAGRAM FOR THE FRONTISPIECE WITH TIKZ25



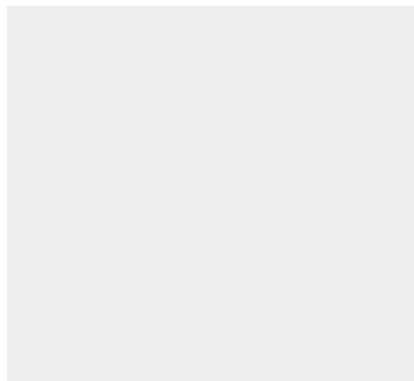
Chapter 7

Including the cover, title pages and frontispiece

[Redacted text block containing multiple paragraphs of placeholder content]

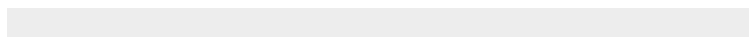
Chapter 8

Adding a list of figures and a list of pre- scriptions

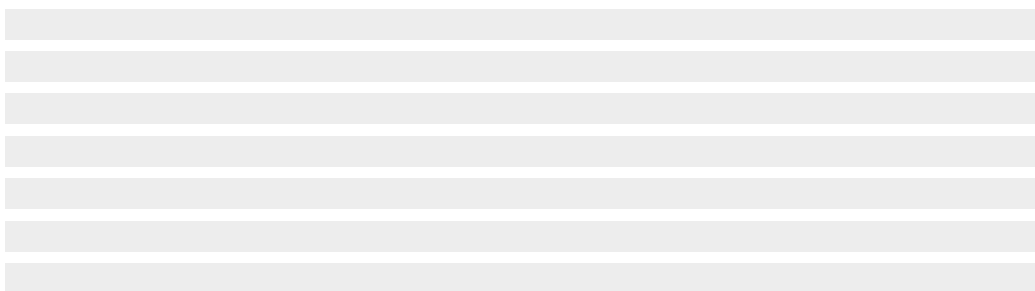


20

Figure 8.1:



21



22

[Redacted text block containing multiple paragraphs of placeholder content]

Part III

Main Part

Chapter 9

Creating formatting and helpful macros

9.1 URLs

Underlined text is not considered particularly pleasing. However, it's an established visual hint for hyperlinks, and we will use it for linking to the book's git repository and the typeface websites.

Using the `contour` package, we can draw the outline of each glyph behind the glyph itself. That way, the descenders of a glyph, which is the part of the glyph going below the base line, do not cross the underline:

hyperlink,

```

1 \usepackage{contour}
2 \usepackage[normalem]{ulem}
3
4 \renewcommand{\ULdepth}{1.8pt}
5 \contourlength{0.8pt}
6
7 \newcommand\colorunderline[1]{\bgroup%
8   \markoverwith{%
9     \textcolor{#1}{%
10      \rule[-0.5ex]{2pt}{0.8pt}}%
11   }%
12 }%
13 \ULon%
14 }
15 \newcommand\myuline[2]{%
16   \colorunderline{#1}{\phantom{#2}}%
17   \llap{\contour{white}{#2}}%
18 }
19 \newcommand\myurl[2]{%
```

```
20 \href{#1}{\myuline{blue}{#2}}%  
21 }
```

Chapter 10

Converting the end notes to a usable format

*CHAPTER 10. CONVERTING THE END NOTES TO A USABLE FORMAT*³⁷

[REDACTED]

[REDACTED]

[REDACTED]

Chapter 11

Using footnotes or margin notes

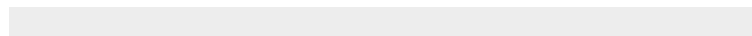
Chapter 12

Typesetting figures



12.1 Placing figures in the text





12.2 Placing figures on their own page



Chapter 13

Designing the synopsis schemata

Chapter 14

Typesetting the author's notes

27

Chapter 15

Typesetting inline translations

Chapter 16

Finding out uncommon vocabulary for a glossary

[REDACTED]

[REDACTED]

[REDACTED]

Chapter 17

Typesetting verse and block quotations

[Redacted text block]

[Redacted text block]

[Redacted text block]

Chapter 18

Creating an OTF for Chaucer with FontForge



Part IV

Appendices

Chapter 19

Creating an index for author and topic references

[REDACTED]

[REDACTED]

[REDACTED]

Chapter 20

Designing a custom biblatex bibliography style to list artwork sources

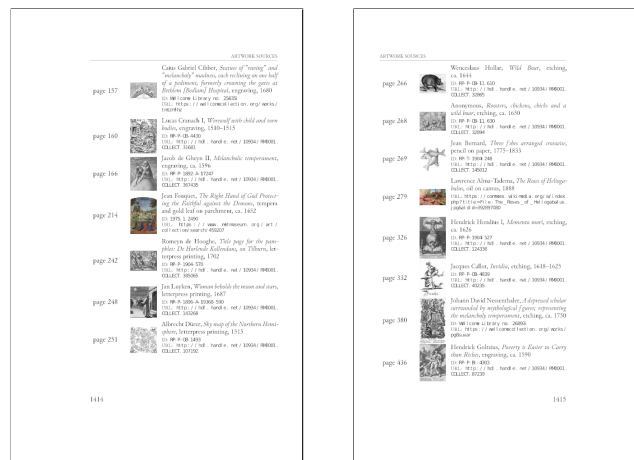


Figure 20.1: Pagespread of art bibliography.

We want to add a list of the artworks included in the book and their sources. We saw how the artwork database was compiled in a bib file back in chapter 4.

biblatex prints the bibliography with a `\printbibliography` macro. The invocation used in the book is:

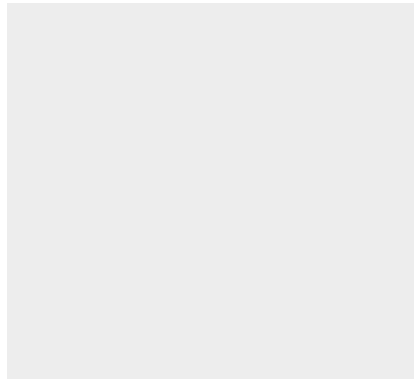
```

1 \defbibnote{artworks-introduction}{All artworks are in the
   ↪ public domain.}
2 \phantomsection
3 \printbibliography[label=app:artworksources,
4                     prenote=artworks-introduction,
5                     type=artwork,
6                     heading=bibnumbered,
7                     title={Artwork Sources}]

```

This will print a chapter with title Artwork Sources and a paragraph after the title, set with the prenote option and defined with a preceding defbibnote command.

Using the numeric bibliography style, the result is this:



33

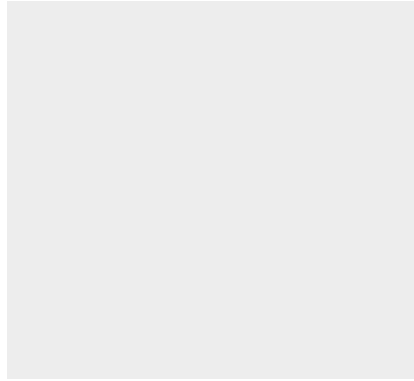
Figure 20.2:

34

Let's copy the numeric source file, called `numeric.bbx` to the project's folder and name it `art-numeric.bbx`.

We want to show a list of every artwork with its name, title, format/medium, year, the source URL and the source's identification number (called *accession number* by librarians). For extra fanciness, let's show a thumbnail of the work and the page where it's located.

First, we want instead of index label numbering to show the page location of each artwork. This is done by defining the bibliography's environment with `\defbibenvironment{bibliograph`



35

Figure 20.3:

36

We want every label to have the same width, and a page number can range from 1 to 3 digits for a book under 10 000 pages. So let's set the label's width to the maximum width possible:

```

1  % This typesets the list entry labels for the artwork
   ↳ bibliography.
2  % The list is of this format:
3  %
4  % page X    [thumbnail] [details]
5  %
6  % page XY   [thumbnail] [details]
7  %
8  % page XYZ  [thumbnail] [details]
9  %
10 % The "page XYZ" part is done in the following
   ↳ bibenvironment macro.
11 % We want the "page .." labels to have the same width
   ↳ regardless of
12 % how many digits the page is. For example we don't want a
   ↳ "Page 5"
```

```

13 % box to be smaller than a "Page 1234" box because this
    ↳ would cause
14 % misalignment of the list items.
15 %
16 % 1. Create new length to store the maximum page label
    ↳ width.
17 \newlength\PageRefLabelWidth
18 % 2. Create new box to store the maximum page label.
19 \newsavebox{\PageRefLabelBox}
20 % 3. Typeset a string that has the same width as the
    ↳ maximum page label
21 % (Book won't be more than 9999 pages, hopefully).
22 \savebox{\PageRefLabelBox}{\mbox{{page 9999}}}
23 % 4. Set length to width of box.
24 \setlength\PageRefLabelWidth{\wd\PageRefLabelBox}

```

Then define the bibliography environment:

```

1 \defbibenvironment{bibliography}
2   {\list
3     % 5. and final step. put the page label in a box of
        ↳ set width.
4
        ↳ {\setlength{\fboxrule}{.5pt}\setlength{\fboxsep}{-.5pt}%
5 \fbox{%
6   \makebox[\PageRefLabelWidth]{%
7     \usebibmacro{pagereflabel}%
8   }}}%
9 \setlength{\labelwidth}{\PageRefLabelWidth}%
10 \setlength{\leftmargin}{\labelwidth}%
11 \setlength{\labelsep}{0pt}%
12 \setlength{\itemsep}{\bibitemsep}%
13 \setlength{\parsep}{\bibparsep}%

```

```

14     \renewcommand*{\makelabel}[1]{\hss##1}}
15     {\endlist}
16     {\item}
17 \endinput

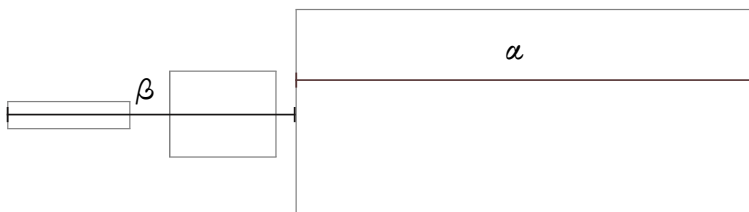
```

We have to decide on the layout of each entry. Let's use the golden ratio ϕ which is a good bet.

The golden ratio which equals $\phi = 1.618\dots$, is two lengths α and β such that $\alpha + \beta$ is to α as α is to β :

$$\frac{\alpha + \beta}{\alpha} = \frac{\alpha}{\beta} = \phi$$

We arbitrarily decide to set each thumbnail's maximum width to the width of the string "thumbnail:". Then, β will contain the page label and the thumbnail, and α will contain the text with the info.



We last encountered length calculations in chapter 12 where we calculated the space required for the annotation text of a figure. That required integer precision, and now we need floating point precision. To do the required arithmetic we use the $\text{\LaTeX}_{3\epsilon}$ floating point interface from the [xfp](#) package.

First, define ϕ as a length. ϕ is equal to $\frac{\sqrt{5}+1}{2}$:

```

1 \edef\GoldenRatio{\fpeval{(sqrt(5)+1)/2}} % phi
2 \edef\GoldenRatioPlusOne{\fpeval{\GoldenRatio+1}} % phi+1

```

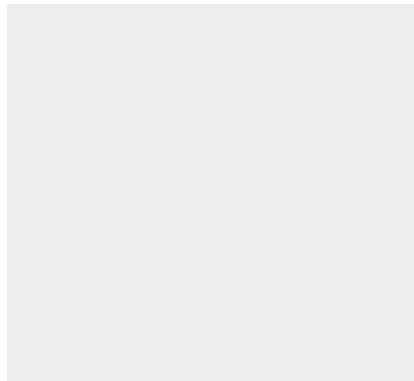
Then $\beta = \phi \times \alpha$ and $\alpha = \frac{\text{textwidth}}{1+\phi}$:

```

1 \newlength\ArtInfoWidth
2 \newlength\ArtThumbnailWidth
3 \newlength\tmpAReg
4 \newlength\tmpBReg
5
6 \newsavebox{\TmpBoxForWidths}
7 \savebox{\TmpBoxForWidths}{\mbox{thumbnail:}}%
8 \setlength\ArtThumbnailWidth{\wd\TmpBoxForWidths}%
9
10 \setlength\tmpAReg{\fpeval{\textwidth/\GoldenRatioPlusOne}pt}
11 \setlength\ArtInfoWidth{\GoldenRatio\tmpAReg}
12 \setlength\tmpBReg{\tmpAReg-\PageRefLabelWidth-\ArtThumbnailWidth}

```

Setting the thumbnail and the text in two minipages we can put them side by side:



37

Figure 20.4:

38

To print the thumbnail let's define a custom biblatex field format. The file's relative path is stored in the file field of each citation entry. We can define a custom format for printing the file field:

```

1 \DeclareFieldFormat{includeartthumb}{\%
2 \graphicspath{./figures/thumbs/}}%
3 \includegraphics[keepaspectratio,width=0.85\textwidth]{#1}}

```

This is a simple `\includegraphics` *only* it changes `\graphicspath` locally, that is inside the group defined by the braces of the macro's definition. Recall that in chapter 4 we set `\graphicspath` to `./figures/`.

We create the thumbnails with a new Makefile target. First, store the paths for each figure in a variable `$INFILES` and the thumb location, which will be inside `figures/thumbs`, in a variable `$OUTFILES`.

```

1 INFILES:=$(shell bash -c "grep file artworks.bib| \
2     cut -d '{' -f2 | cut -d'}' -f1 | sed
3     ↪ 's~/figures\\/'")
4 OUTFILES:=$(shell bash -c "grep file artworks.bib| \
5     cut -d '{' -f2 | cut -d'}' -f1 | sed
6     ↪ 's~/figures/thumbs\\/'")

```

Next, add the target dependency to the pdf:

```

1 anatomy-of-melancholy.pdf: main.tex *.tex $(OUTFILES)

```

This means that before compiling the pdf, the targets `figure/thumbs/foobar_1`, etc. will have to be compiled first. Let's add these targets:

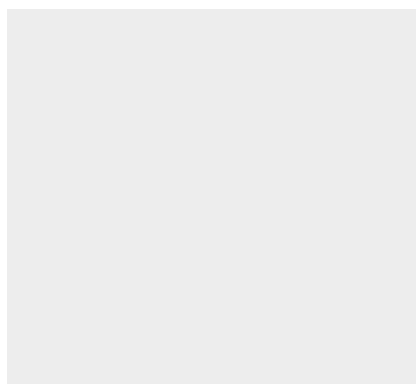
```

1 figures/thumbs/%.jpg: figures/%.jpg
2     @mkdir -p figures/thumbs
3     convert -thumbnail 100 "$<" "$@"

```

The first line creates the `thumbs` folder if it does not exist, and the second is an ImageMagick command that makes a thumbnail of width 100px.

Returning to \LaTeX , print the author, title, year, url, etc:



39

Figure 20.5:

40

Observing the result in Figure 20.6 we see that the actual page dimensions end up as $\alpha + \beta = 10.97\text{cm}$, $\alpha = 6.77\text{cm}$ and $\beta = 4.20\text{cm}$. Indeed the ratio checks out:

$$\frac{\alpha + \beta}{\alpha} = \frac{10.97}{6.77} \approx 1.6204, \text{ and}$$

$$\frac{\alpha}{\beta} = \frac{6.77}{4.2} \approx 1.612 \text{ which is good enough.}$$

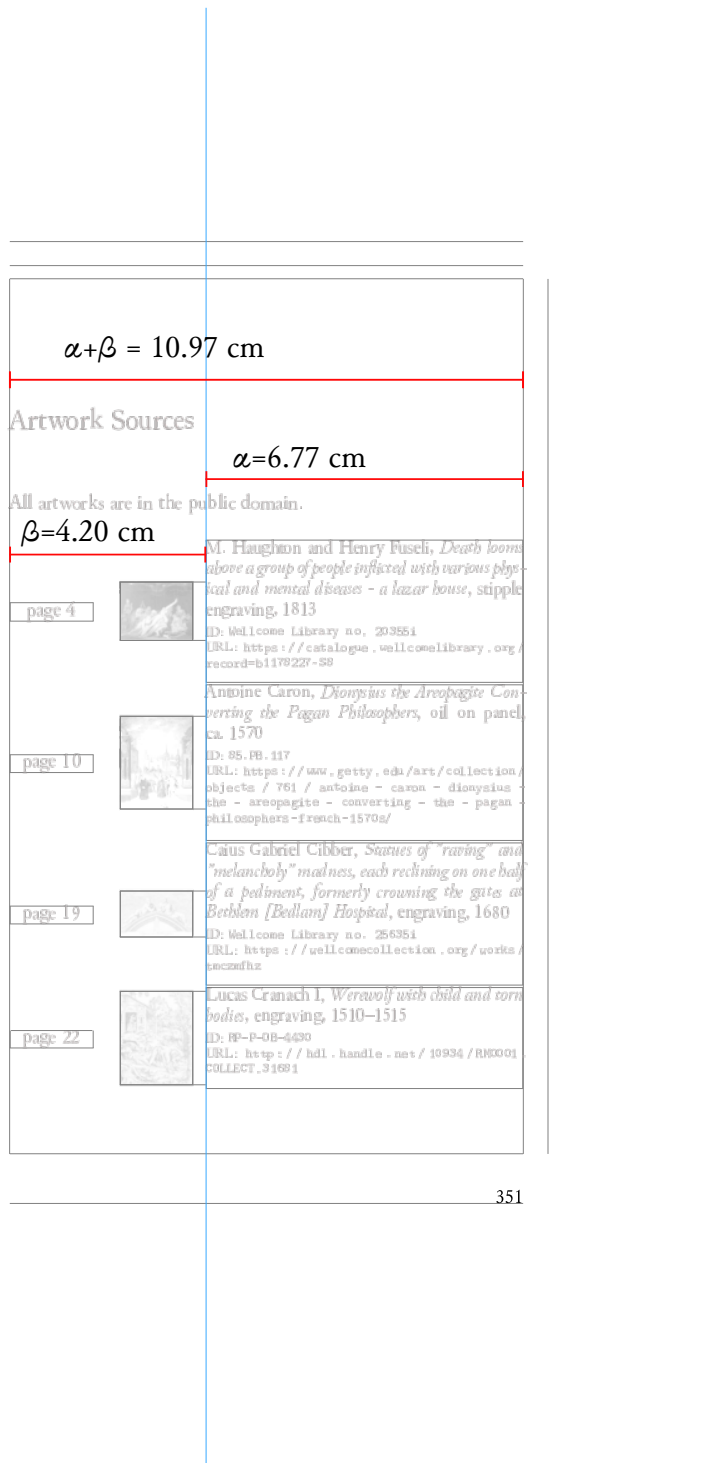


Figure 20.6: Page layout and proportions

Chapter 21

Adding an editor's postface

41

Chapter 22

Closing with a colophon

