# The Mottaz Standard LaTeX Preamble:
## An Introduction

Tony Mottaz

Last updated: February 10, 2016

## Contents

# 1 Introduction

The beginning of every LaTeX document has a preamble. If you write lots of documents in LaTeX, chances are you have created your own dedicated `preamble.tex` file which you load into every document. As a graduate student in mathematics, I started making my own. As I discovered more and more ways to customize my document and make handy macros, my preamble started to grow. This preamble has become extremely useful for me, and I have put a lot of thought and effort into creating a comprehensive collection of packages and creation of macros which serve the purposes of math students and professors well.

The Mottaz Standard LaTeX Preamble is, in fact, a collection of preambles, each with their own specific use. These preambles are:

- `HOMEWORK` edition
- `QUIZ` edition

The present document is intended to serve as an introduction and usage guide for these preambles I have created. It is organized by first describing all packages and features that are similar to every preamble with sections near the end describing features that are specific to individual preambles. my audience/end-users are students and professors of mathematics who appreciate beautiful and professional looking documents.

Please be sure to take a look at the `README` file that comes along with each preamble. In there you will find helpful information as well as a log of changes between each version.

I have included here my list goals for the future of this project, how to utilize this preamble, and a warning about package related errors that you want to avoid.

If you have any suggestions for changes/additions, or if you have problems/errors of any kind, please send me an email at:

`anthonywmottaz@gmail.com`

Happy TeXing!

*Anthony Mottaz*
*Graduate Assistant*
*Northern Illinois University*

### 1.1 Goals for the future

**I.** Redevelop this preamble into a `class` document.

  **i.** Create a key value option for alternative layout schemes, such as `oneside` vs `twoside`, different options for margins (such as a "tight" scheme), etc. Each scheme should be compatible with any chosen paper size.

  **ii.** Create a key value option for disabling the `microtype` package if using an incompatible font

**II.** Create a library of other preambles for other specific uses, such as

  **i.** Homework (which the current preamble does well)

  **ii.** Creating solution guides for students

  **iii.** Creating tests/quizzes (This project has begun)

**III.** Create a `\randclosedloop` command for creating arbitrary 2-dimensional spaces in tikz pictures. *This project is underway... look for it in the near future!*

**IV.** Add a command similar to `\layout` which will print a page of the unique commands offered by this preamble

## 2 Basic usage

The following gives a guide of how to incorporate this preamble into your own LaTeX documents. I have also included an important warning about potential errors the user might run into regarding the installation/loading of the numerous packages.

### 2.1 Utilizing this preamble

The use of a preamble is straightforward. Save the preambles to your system, and make note of their path. Note that the `HOMEWORK` edition is simply called `preamble.tex`, whereas the other editions have a prefix. For example, you may save the `HOMEWORK` edition preamble to something like

```
C:Users/YourName/Documents/preamble.tex   (Windows)
/home/yourname/Documents/preamble.tex   (Linux)
```

A similar path should give you the `QUIZ` edition:

```
C:Users/YourName/Documents/quiz_preamble.tex   (Windows)
/home/yourname/Documents/quiz_preamble.tex   (Linux)
```

Then in the preamble of your document, you will use the `\input{}` command, filling in the path to your saved preamble. When your `.tex` file compiles, the preamble will be read as if it was actually typed right there in your document. Next, be sure to set the header variables[†]. Here is a sample:

```
1  \documentclass[letterpaper]{article}
2
3  \input{/home/yourname/Documents/preamble.tex}
4  \coursetitle{Math Class}
5  \hwtitle{Homework \#1}
6  \myname{Joe Schmö}
7
8  \begin{document}
9      ...
```

## 2.2 Avoid package related errors

The preamble loads many packages. It is important that you follow these guidelines to avoid errors in your document. First, there may be some packages which you do not already have installed on your system. Refer to the `README` for an up-to-date list of these. If you are using the MiKTEX package manager (recommended for Windows systems), then just run a search for these packages and install them[‡]. If you are using the TEX Live package manager (recommended for Linux systems), then go into a terminal and type

```
sudo tlmgr install packagename
```

to install the package `packagename`. Now that you have all of the necessary packages installed, it is important that you *do not change the order in which they are loaded.* Some packages depend on other packages, and some packages load other packages on their own. If there are specific options that are requested from a certain package, then the package needs to be loaded with those options before another package

[†]To see how this is done, go to the section corresponding to the preamble you are using.

[‡]A few packages, such as my `signchart` package, may need to be downloaded from the CTAN archive.

loads it. If these packages are not loaded in the correct order, you will see an "`option clash`" error.

# 3 The loaded packages

The following is a list of packages that are loaded into the preamble along with a description of what the package does and perhaps an example of its usage. This list is rather long, and it is growing. You may be a minimalist and think that it is silly to load so many things every time, when maybe there are several packages you rarely use. I keep them all here for two reasons: (1) Since we live in the modern age of incredible processor speeds, I do not see any lag in the time it takes to compile a document, so from that standpoint I do not have any motivation to minimize this list. (2) Even though some features are used rarely, it is nice to have them there when you need them.

*Click on the package name to open the documentation in your browser.*

**layout** This package gives the command `\layout` which prints a diagram showing the current layout of the document. The output for this document is shows in appendix A.

**lmodern** This package provides the Latin Modern font — and enhanced version of the Computer Modern fonts with extended glyph coverage and enhanced metrics.

**fontenc** This package is a standard for changing the font encodings in the output PDF document. It is loaded as usual with the `T1` option, specifying the use of T1 encoding.

**inputenc** This package, loaded with the `UTF-8` option, allows LaTeX to read and understand special characters from your source document. For example, if you insert the character 'ö' into your source document, it will give the expected output in your PDF.

**mathtools** The `mathtools` package offers all of the same features as the **amsmath** package, along with additional features and math characters, such as the "colon equals" symbol ≔ which I use frequently. It also allows for "cramped" math. For example, the `\cramped{}` command changes $2^{2^2}$ to $2^{2^2}$. There are many other nice features offered by this package.

**amsthm** This package provides the `theorem` and `proof` environment, as well as tools for building your own theorem-like environments. To see how to define your own theorem-like environment, look at section 5.3.

**amssymb** This package offers additional symbols to use in math mode. A complete list of these symbols can be found here.

**dsfont** This package loads the double stroke letters which are used for representing the natural numbers, the integers, the rationals, the reals, the complex numbers, the Quaternions (Hamiltonians), and a general field. The preamble also defines macros to access these symbols easily. Those macros and their output are as follows:

| Command | Output |
|---|---|
| \N | $\mathbb{N}$ |
| \Z | $\mathbb{Z}$ |
| \Q | $\mathbb{Q}$ |
| \R | $\mathbb{R}$ |
| \C | $\mathbb{C}$ |
| \H | $\mathbb{H}$ |
| \F | $\mathbb{F}$ |

**mathrsfs** This package provides script lettering using the `\mathscr{}` command. For example, `\mathscr{ABC}` produces $\mathscr{ABC}$.

**cancel** This package offers the `\cancel{}` and `\cancelto{}{}` commands (and a few others) which nicely draw a line through math expressions, and possibly draws an arrow to another value. For example, the code

```
\[ \frac{x^2 - 4}{x+2}
= \frac{(x - 2)\cancelto{1}{(x + 2)}}{\cancel{x + 2}}
= x - 2. \]
```

will produce

$$\frac{x^2-4}{x+2} = \frac{(x-2)\cancel{(x+2)}^{1}}{\cancel{x+2}} = x-2.$$

**enumitem** This package allows for total customization control over list environments. This package is loaded with the `shortlabels` option so that the item labels can be customized easily by selecting `A`, `a`, `I`, `i`, or `1` and inserting whatever delimiters you want. For example, the code

```
\begin{enumerate}[\bfseries a.]
        \item First item
        \item Second item
        \item Third item
\end{enumerate}
```

will produce

**a.** First item

**b.** Second item

**c.** Third item

**array** This package offers many helpful customization tools for `tabular` and `array` environments. This preamble uses the `\extrarowheight` parameter to add some height to each row in a `tabular` or `array` environment. Usually, if horizontal lines are used, then capital letters may touch the lines. This preamble defines `\extrarowheight` to be `2pt`, which gives enough buffer for letters to have space between horizontal lines and still look natural. Here is an example:

| I FEEL A LITTLE SQUISHED |
|---|

| THIS IS BETTER |
|---|

**arydshln** This package (say to yourself "array dashed lines") allows the user to use dashed lines in `tabular` and `array` environments. For vertical dashed lines, use the `:` option in place of `|`. For horizontal dashed lines, use `\hdashline` in place of `\hline`. For example, the code

```
\begin{tabular}{::l:c|r||}
Here is & an & example \\ \hdashline
Using & dashed & lines \\ \hline
\end{tabular}
```

will produce

| Here is | an | example |
|---|---|---|
| Using | dashed | lines |

**relsize** This package allows the user to change the size of math or text relative to what the current size is. For example, you have the following expression

$$\frac{x+\frac{3}{2e^y}}{x^{2y-1}}$$

it would be nice if everything was typeset a little larger so that you can

see the fractions and exponents better. Just put the whole expression inside of the `\mathlarger{}` command. For example, the code

```
\[ \mathlarger{
        \frac{x + \frac{3}{2e^y}}{x^{2y-1}}
        } \]
```

will produce the following change:

$$\frac{x + \frac{3}{2e^y}}{x^{2y-1}} \quad \longrightarrow \quad \mathlarger{\frac{x + \frac{3}{2e^y}}{x^{2y-1}}}$$

**xcolor** This package gives the user access to a wider range of predefined colors, as well as several options for defining your own colors. This package is loaded with the `dvipsnames` option to load the 68 standard colors known to `dvips`:

| Name | Col | Name | Col | Name | Col |
|---|---|---|---|---|---|
| Apricot | | Aquamarine | | Bittersweet | |
| Black | | Blue | | BlueGreen | |
| BlueViolet | | BrickRed | | Brown | |
| BurntOrange | | CadetBlue | | CarnationPink | |
| Cerulean | | CornflowerBlue | | Cyan | |
| Dandelion | | DarkOrchid | | Emerald | |
| ForestGreen | | Fuchsia | | Goldenrod | |
| Gray | | Green | | GreenYellow | |
| JungleGreen | | Lavender | | LimeGreen | |
| Magenta | | Mahogany | | Maroon | |
| Melon | | MidnightBlue | | Mulberry | |
| NavyBlue | | OliveGreen | | Orange | |
| OrangeRed | | Orchid | | Peach | |
| Periwinkle | | PineGreen | | Plum | |
| ProcessBlue | | Purple | | RawSienna | |
| Red | | RedOrange | | RedViolet | |
| Rhodamine | | RoyalBlue | | RoyalPurple | |
| RubineRed | | Salmon | | SeaGreen | |
| Sepia | | SkyBlue | | SpringGreen | |
| Tan | | TealBlue | | Thistle | |
| Turquoise | | Violet | | VioletRed | |
| White | | WildStrawberry | | Yellow | |
| YellowGreen | | YellowOrange | | | |

**tikz** This package is essential for any LaTeX users who wish to create any mathematical (or other) images. The link provided is for a site which provides an ever increasing gallery of examples of pictures created with `tikz`. I have also loaded the `tikz` library `shapes`, which is used in the definition of my `\circled{}` command (see section 4.4)

**pgfplots** This package is an extension of `tikz` which provides a simple set of commands for producing plots and graphs. The command
`\pgfplotsset{compat = 1.10}`
means that we are using version 1.10, and the command
`\usepgfplotslibrary{fillbetween}`
loads a collection of commands which easily allow the user to add a shade color between curves on plot. A gallery of plots created with `PGFPlots` can be found here.

**needspace** This package provides the `\needspace{}` command. If the amount of space requested is not available on the current page, then a page break is inserted. This was used to redefine the `proof` environment. I don't care for proofs which have three or fewer lines at the end of a page, so the environment has been redefined using
`\Needspace*{4\baselineskip}`

**todonotes** This package lets the user insert "todo" notes into their document using the `\todo{}` command. This command places a colored note in the margin with a line pointing to the location of the note. The user can also insert a note inline with `\todo[inline]{}`.

Here is an example todo note.

Here is an inline note.

You can also insert a `\listoftodos` anywhere in your document to keep track of all of your todo notes:

## Todo list



**fancyhdr** This package provides an addition `pagestyle` called `fancyhdr`, seen here in this document. There are three header locations (`lhead`, `chead`, `rhead`) and three footer locations (`lfoot`, `cfoot`, `rfoot`) where text may be placed, and a horizontal line spanning the width of the text is placed underneath the header.

**parskip** This package removes the indent from new paragraphs and instead adds a space between them. In my opinion, this format is more aesthetically

pleasing.

**imakeidx** This package allows the user to create an index in their document. Refer to the documentation for instructions on how to use this. **Important:** You will need a new build sequence for compiling your document. After compiling the source once using `latex` or `pdflatex`, you must run `makeindex` on the `\jobname.idx` file that is created, and then run `(pdf)latex` again.

**hyperref** This package inserts hyperlinks into your document. The table of contents, todo list, and other generated lists are automatically created with click-able links. Other links (such as the package name links here) are created with the command `\href{}{}`, where the first argument is a URL, and the second argument is the text for the link. References to labeled objects are also automatically created as hyperlinks. This package is loaded with the `unicode` option so that Greek letters in section headings may be placed in the Adobe bookmarks.

**cleveref** This package offers additional hyperlink capabilities.

**framed** This package provides several environments, including the `framed` environment, which automatically draws a rectangle around everything inside, and the `shaded` environment which creates a shaded region containing the contents within the environment. Before using the `shaded` environment, the user needs to define the shade color. For example, the code

```
\begin{framed}
        This content has a frame around it.
\end{framed}

\definecolor{shadecolor}{RGB}{181,230,216}
\begin{shaded*}
        This content is shaded.
\end{shaded*}
```

will produce

This content has a frame around it.

This content is shaded.

**wasysym** This package provides many additional symbols that can be used in a

document. Refer to the documentation for a list of these symbols. For a well-maintained list of all available symbols in LaTeX, look here.

**lipsum** This package allows the user to insert dummy text, which is helpful during the document creating process. There are 150 paragraphs of the *Lorem ipsum* dummy text, and the user can access any range of these paragraphs using the `\lipsum[]` command. For example, the code

```
\lipsum[2]
```

will produce

> Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

**alphalph** This package gives additional enumeration options for sections, lists, etc. In particular, if a list of symbols is used for enumeration, then the symbols will be duplicated. This was implemented in my redefinition of the footnote symbols (see section 5.2 below).

**pdfpages** This package provides the `\includepdf[]{}` command, which allows the user to easily insert selected pages of any PDF document. For example, the command

```
\includepdf[pages=1-3,5]{SomeOtherThing.pdf}
```

will insert pages 1, 2, 3, and 5 of the PDF called "`SomeOtherThing`" directly into your document.

**float** This package gives some additional control over floats and figures, including the addition of a placement character `H`, which means "put it here, no matter what". Use with caution.

**tabularx** This package provides the `tabularx` environment, which gives the user the ability to define the width of a table, and the width is added to columns labeled with the `X` specification. For example, the code

```
\begin{tabular}{| l | c | r |}
\hline
These columns & are scaled & to the text within. \\ \hline
\end{tabular}

\begin{tabularx}{\textwidth}{| l | X | r |}
\hline
The middle column & will & be expanded. \\ \hline
\end{tabularx}
```

will produce

| These columns | are scaled | to the text within. |
|---|---|---|

| The middle column | will | be expanded. |
|---|---|---|

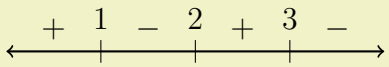
**textgreek**, **upgreek** These packages allow the user to insert upright Greek letters into text mode and math mode, respectively. For example, the normal Greek "beta" looks like $\beta$, but the upright "beta" looks like $\upbeta$. These upright letters are achieved in text mode using commands such as `\textalpha` for lower case, or `\textAlpha` for upper case. They are achieved in math mode using commands such as `\upalpha` for lower case, or `\Upalpha` for upper case.

**fancyvrb** This package allows the user to create fancy `Verbatim` environments, which have been used throughout this document. The user can specify different frames, shading, and text styles, as well as include line numbers, and more.

**signchart** This is a package developed by me to give the user the ability to easily insert nice sign charts. For example, the code

```
\signchart{1,2,3}{+,-,+,-}
```

will produce




**microtype** This package makes subtle changes to the size and placement of text in order to enhance the appearance and readability of the document.

**multicol** This package allows the user to switch in and out of multiple column mode without automatically starting a new page, and using balanced

columns. By using `\begin{multicols}{n}`, the text will enter multiple column mode with $n$ columns.

# 4 Other features

This section outlines the features that are shared by every preamble.

## 4.1 Bold math

To insert bold text, one uses the `\textbf{}` command. If math is inserted into this command, such as

```
\textbf{Let $\varepsilon > 0$ be given.}
```

then the normal output will be

**Let $\varepsilon > 0$ be given.**

I would like math to also become bold. This is achieved using with the following segment of code:

```
\DeclareTextFontCommand{\textbf}{\boldmath\bfseries}
```

Now the same code as above will produce

**Let $\varepsilon > 0$ be given.**

## 4.2 Dark theme

Many plain text editors offer a dark colored theme for decreasing eye strain while typing. Since generally about half of my screen is displaying the output PDF file of the `tex` document I am working on, I thought it would be nice to have a dark colored PDF until I had my finished product. Simply use the command `\darktheme` in your preamble, and your output will look like this:


## 4.3 Math-related commands

There are many math-related commands that are specified for your convenience.

### 4.3.1 Auto-scaled delimiters

Delimiters can be automatically scaled using the `\left` and `\right` commands. This has been automated for the following delimiters:

| Command | Output |
|---|---|
| `\paren{}` | $(\cdot)$ |
| `\ang{}` | $\langle\cdot\rangle$ |
| `\brc{}` | $\{\cdot\}$ |
| `\brkt{}` | $[\cdot]$ |
| `\abs{}` | $\lvert\cdot\rvert$ |
| `\norm{}` | $\lVert\cdot\rVert$ |

### 4.3.2 Special functions

Macros are defined for the ceiling function, via `\ceil{}`, and for the floor function, via `\floor{}`.

### 4.3.3 Operators and other symbols

The `amsmath` package provides the `\DeclareMathOperator{}{}` command which allows us to define operators. The predefined operators here are:

| Description | Command | Output |
|---|---|---|
| Isomorphism | `\iso` | $\cong$ |
| Normal subgroup | `\nsg` | $\trianglelefteq$ |
| Reversed normal subgroup | `\rnsg` | $\trianglerighteq$ |
| Negated normal subgroup | `\nnsg` | $\ntrianglelefteq$ |
| Gradient, i.e. 'del' operator | `\del` | $\nabla$ |
| Complex conjugate | `\bar{}` | $\bar{a}$ |
| The order of a set/element | `\ord` | ord |
| The "sign" function | `\sgn` | sgn |
| The least common multiple | `\lcm` | lcm |
| The algebraic group of automorphisms | `\aut` | Aut |
| The algebraic group of inner automorphisms | `\inn` | Inn |
| The symmetric group | `\sym` | Sym |
| The identity operator | `\id` | id |
| The image of a function | `\img` | Im |
| The stabilizer of an element under a group action | `\stab` | Stab |
| The orbit of an element under a group action | `\orb` | Orb |
| The conjugacy class | `\cl` | $\mathrm{C}\ell$ |
| The core | `\core` | core |
| A Sylow group | `\syl` | Syl |
| The characteristic of a ring | `\cha` | char |
| The trace of a matrix | `\tr` | tr |
| The set of all functions | `\fun` | Fun |
| $\cos + i \sin$ | `\cis` | cis |
| Principal Argument of a complex number | `\Arg` | Arg |
| Limit superior | `\limsup` | $\overline{\lim}$ |
| Limit inferior | `\liminf` | $\underline{\lim}$ |
| Upper integral | `\upint` | $\overline{\int}$ |
| Lower integral | `\lowint` | $\underline{\int}$ |
| Display-sized sum | `\dsum` | $\sum$ |

I have also redefined the `\Re` and `\Im` commands so that instead of printing the fraktur $\Re$ and $\Im$, it prints Re and Im.

### 4.4 Circled

By default, LaTeX does not do a very good job circling numbers in text, so I created a command `\circled{}` which will place a nice circle around numbers. Examples: ④, ($e^x$), (long string)

### 4.5 Margin text

I have created a command `\mpar{}` as a substitute for `\marginpar{}`. The only difference is that the text is set to `\footnotesize`.

Here is a sample set of text using the old command. Notice how often line breaks occur and how annoying it is to read. You are also more likely to get `underfull hbox` errors here.

## 5 Features in the HOMEWORK edition

This section outlines the features that are unique to the `HOMEWORK` edition.

### 5.1 Page layout

The layout of each page is defined as such:

```
\hoffset 0.06\paperwidth
\voffset -0.025\paperheight
\oddsidemargin 0pt
\topmargin 0pt
\headheight 1.3em
\headsep 1.3em
\textheight 0.78\paperheight
\textwidth 0.59\paperwidth
\marginparsep 1.5em
\marginparwidth 0.17\paperwidth
\footskip 2.5\headheight
```

Here is a sample set of text in the margin using the new command. Isn't this much nicer to read than the example above? Now line breaks occur less often, and eye strain is reduced.

The first thing you should notice is that each length is defined entirely on the dimensions of the paper and the size of the text. This is important for compatibility across several page sizes.

The dimensions chosen here generally follow the recommendations set out by typographers. Read the discussion here. The author of the most up-voted answer mentions the `tufte` class, which was clearly created with much thought on typography. This class was intended for writing books, and I think the margin size therein is too large to

be reasonable for this preamble, where the primary use is for writing homework assignments. However, you will find that when the paper size is `letterpaper` (standard in the U.S.) and the text size is `11pt` or `12pt`, then the text width falls within the recommended 60-75 characters per line, and if the text size is `10pt`, then you will see roughly 75-85 characters per line. I feel that this definition is a nice compromise between the typographical guidelines and the (relatively new) desire for roughly 1 inch margins.

## 5.2 Footnote symbols

Since mathematics homework is the intended primary use of this preamble, it is important that the symbols inserted for footnotes will not be confused with mathematical expressions. Enumerating the footnotes with Arabic numerals (1, 2, 3, etc) can look like exponents, so we need to use symbols. However, the default footnote symbols start with an asterisk. This is a problem, because oftentimes an asterisk is used to denote a "special" mathematical object, so a new list of symbols need to be defined.

I decided to use the following symbols: † ‡ ✠ ⋆. These symbols should avoid nearly all confusion associated with mathematical expressions, and, using the `alphalph` package, these symbols will be multiplied if more footnotes are created than available symbols.

I did not include this feature in the `QUIZ` edition, since I do not imagine ever needing footnotes on a quiz.

## 5.3 Theorem styles

As mentioned above, the `amsthm` package allows us to create our own theorem-like environments. The predefined environments in this preamble are:

- Theorem, using `\begin{thm}`
- Proposition, using `\begin{prop}`
- Lemma, using `\begin{lem}`
- Corollary, using `\begin{cor}`
- Numbered versions of everything above, using `\begin{nthm}`, etc.
- Definition, using `\begin{defn}`

- Example, using `\begin{exmp}`
- Solution, using `\begin{sol}`
- Case, using `\begin{case}`
- Note, using `\begin{note}`
- Claim, using `\begin{claim}`

### 5.4 Framed definitions/theorems

By default, definitions and theorem/proposition/lemma/corollary environments will typeset without any special ornamentation. If you would like definitions or theorems/propositions/lemmas/corollaries to be typeset with a frame around them, simply type `\frameddefinitions`, `\framedtheorems`, or `\framedntheorems` in your preamble.

### 5.5 The header

You can populate the header by using the following three commands: `\coursetitle{}`, `\hwtitle{}`, and `\myname{}`. I recommend editing this preamble directly by inserting your name as I have done so that you can save yourself from setting your name for every new document.

## 6 Features in the `QUIZ` edition

### 6.1 Page layout

The layout of each page is defined as such:

```
\hoffset -0.03\paperwidth
\voffset -0.04\paperheight
\oddsidemargin 0pt
\topmargin 0pt
\headheight 1.3em
\headsep 1.3em
\textheight 0.78\paperheight
\textwidth 0.82\paperwidth
\marginparsep 1.5em
```

```
\marginparwidth 0.17\paperwidth
\footskip 2.5\headheight
```

The first thing you should notice is that each length is defined entirely on the dimensions of the paper and the size of the text. This is important for compatibility across several page sizes.

The dimensions chosen here do not follow the same guidelines as for the `HOMEWORK` edition, and this is strictly for aesthetic reasons. Here, we achieve a symmetric look with approximately 1 inch margins.

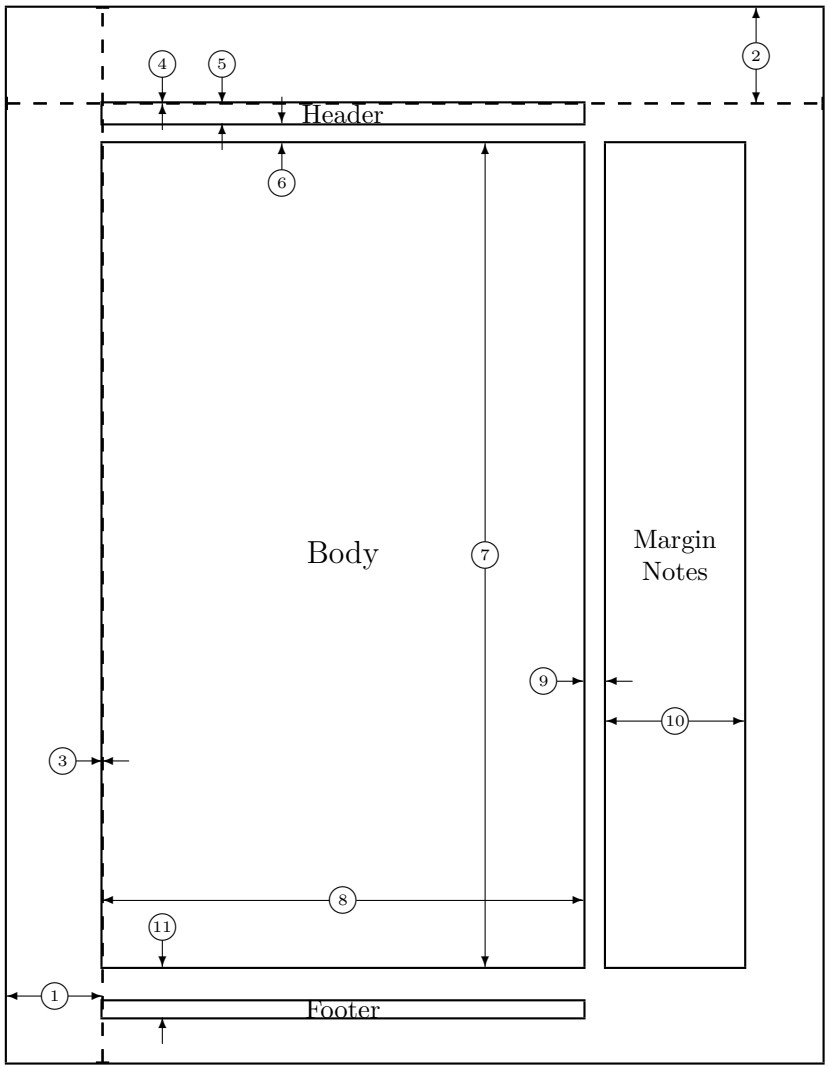
## 6.2 The header

You can populate the header by using the following two commands: `\coursetitle{}` and `\quiztitle{}`.

## 6.3 Points

You can insert '`(n points)`' by using the command `\points{n}`.

# A Output from \layout




```
 1  one inch + \hoffset      2  one inch + \voffset
 3  \oddsidemargin = 0pt     4  \topmargin = 0pt
 5  \headheight = 15pt       6  \headsep = 15pt
 7  \textheight = 620pt      8  \textwidth = 362pt
 9  \marginparsep = 17pt    10  \marginparwidth = 104pt
11  \footskip = 38pt            \marginparpush = 7pt (not shown)
    \hoffset = 0pt              \voffset = 0pt
    \paperwidth = 614pt         \paperheight = 794pt
```

# B Some LaTeX references

Here is a list of references that I have found useful:

- The comprehensive symbol list: click here
- The LaTeX font catalog: click here
- The Comprehensive TeX Archive Network: click here
- Debugging your code: click here
- The TeX-LaTeX stack exchange: click here
- Handwritten symbol recognizing tool: click here
- `tikz` and `PGFPlots` examples: click here
- `PGFPlots` gallery: click here
- Anything else: click here