

IBT_EX

A Template for Writing the IB Extended Essay and Internal Assessments.

mholson

Words: 1234

Contents

1	Introduction to mhotext	3
1.1	Purpose	3
1.2	Where to Begin?	3
1.3	Project Structure	4
1.4	The Main File	4
1.5	texmf - The Template Files	4
1.6	Requirements	4
1.7	Getting Started	5
2	Typesetting Mathematics	7
2.1	Inline Mathematics	7
2.2	Display Style Math Mode	9
2.3	Multiline Math Environments	10
2.4	Cases Environment	15
3	Figures & Images	17
3.1	Storing Images	17
3.2	Including Images	17
3.3	Figure Environment	18
3.4	Multiple Horizontally Aligned Images	18
4	Theorem Environments	21
4.1	Definition	21
4.2	Examples	21
4.3	Theorems	21

1 Introduction to mhotext

§1.1 Purpose

This template is designed for International Baccalaureate™ students who are looking to write an extended essay or an internal assessment using L^AT_EX. The template is based on the KOMA Script scrbook class and styled using the custom style file `mhotext.sty`¹, which was inspired by Evan Chan's code used to create Napkin².

While I have done my best to ensure this template meets the submission requirement of the IB, I cannot guarantee your document will meet these standards. It is the responsibility of the student to ensure that any work submitted is formatted correctly by ensuring they are following the official IB documentation and the advice given by their supervising teacher.

While there is extensive online support and documentation on how to write a document using L^AT_EX, it can still be overwhelming for anyone who is just getting started. I have been using T_EX since 2005 and I am still learning! Part of this documentation will be used to provide you with enough knowledge to get you up and writing as quickly as possible.

§1.2 Where to Begin?

Since T_EX is a programming language we are going to need a text editor to write our code and a compiler to transform that code into a pdf document. To keep things simple, we are going to be using an online service called Overleaf³, which will provide us with a modern text editor and a range of compilers that we can use all within the comfort of our favorite web browser.

Now, rather than think of our document as one file that we edit, such as a MS Word or Google document, we are going to think of it as an object called a project which will be made up of files and folders. For us, there is going to be one main file called `main.tex` that will make use of other files within the project to build a pdf of your extended essay or internal assessment.

Example 1. Rather than drag an image into a MS Word document, we will now upload the image to our project and insert it into our pdf document using a line of code.

It doesn't stop there! All the template files necessary to format your document will also exist in your project. Don't worry, these files will be tucked away in a folder called `texmf`.

We are starting to get a little far ahead of ourselves here. All that really matters at this point is that we have an object called a project that is made up of files and folders.

¹<https://github.com/mholson/mhoDotFiles/texmf/>

²<http://web.evanchen.cc/napkin.html>

³<https://overleaf.com>

§1.3 Project Structure

Our project is going to be made up of three folders and one main file called `main.tex`. Let's describe the contents of those three folders.

texmf You probably will not do very much with the files located in this folder as these are the template files that are responsible for formatting the content of your essay. In most cases, you can change the formatting elements of your document right from the `main.tex` file. It is also good practice to not edit these files directly, so that your theme can easily be upgraded if necessary.

assets An asset is a file that you are going to include in your document. In most cases, the only files that you will be including in your document will be images and thus you can upload your images directly to this folder. You can have as many subfolders within your assets folder, but again this folder should be good enough to upload your images into.

chapters This folder is really optional. You could technically write your document all within the `main.tex` file; however, it might be beneficial to break your document up into chapters and write these in separate files. If you are not going to be structuring your paper using chapters, then you should probably just write your document in the `main.tex` file.

Like I said before, I want to get you up and writing as quickly as possible. We are going to get right into editing the `main.tex` file and producing a pdf file with some content. Action!

§1.4 The Main File

The `main.tex` file is where all the action takes place. It is in this file where you will write your content or at least link to the content found in other files in your project. This file can be as simple or as complicated as you would like it to be. I have provided a skeleton template of this file in your template project. So let's start looking at the code.

For us the file is going to start of with multiple lines of text that are going to be ignored by the compiler, which means it will not be included in our pdf output.

§1.5 texmf - The Template Files

Most of you will never need to look in this folder; however, I thought I would provide a short summary of what files are included and how they are being used to format your document.

§1.6 Requirements

All the file dependencies outside the scope of a typical \TeX distribution are available from my texmf repository hosted on Github⁴. The Overleaf™ version will come with batteries included in the texmf directory, which in include.

⁴<https://github.com/mholson/mhoTexmf/tree/main/texmf>

`maabook.cls` the main template file.

`mhotext.cls` customizations made the to main template file.

`mhotheorems.sty` styling custom theorem environments

`mhocolorthemenord` custom color theme based on Nord Theme.

`mhocodestyle.sty` styling code via the listings package.

`mhomath.sty` some of my custom math commands.

§1.7 Getting Started

Definition 1 (Code Comment). Any code that is written in your file that is ignored by the compiler is called **commented text**. All characters on the same line following a `%` symbol will be regarded as commented text and ignored.

Comments are great for including additional information, making temporary changes to your document (without having to erase) and documenting your code for others to follow — including yourself!

```
% This line would be ignored by the compiler.
This line is read by the compiler.

The compiler is reading % and now it is ignoring.

This line is read by the compiler.
The compiler is reading
```

One of the benefits of writing in \LaTeX is the ability to write comments that are not typeset. Any text following a `%` character will be ignored. Not only is this ideal for leaving comments and ideas, but it is als a way to make temporary changes to your document.

```
% This code will be ignored
This text will be visible % while this text will be ignored.
```

This text will be visible

Much of your document will be filled with commands that take the form `.` In fact the first non-comment line of your document will be the command that defines the document template.

```
\documentclass[a4paper]{maatext}
```

more text here

2 Typesetting Mathematics

This chapter will focus on inserting mathematics into your paper. \LaTeX comes with basic mathematics support out of the box, so we will need to incorporate some additional packages to enhance the range of formulas that can be included in our document. You can think of it as adding the functionality of an equation editor into your current word processor. As you become more familiar with the syntax, you will quickly appreciate being able to type your formulas with the same ease you write text.

\LaTeX comes with two modes that are of interest to us:

1. normal and
2. math, which has the two styles
 - a) inline and
 - b) display.

Normal mode is the default mode and where you will write the non-mathematical text of your document. If you want to include expressions and equations, then you are going to need to use either an inline or display math mode, which are environments specific for mathematics. However, we are going to want to include more advanced mathematical formatting such as multiline equations and for this we are going to include some additional packages including those from the $\mathcal{A}\mathcal{M}\mathcal{S}$, American Mathematics Society, and the `mathtools`[2] packages which will bring some enhancements to the $\mathcal{A}\mathcal{M}\mathcal{S}$ math package[1]. Don't worry, these packages are loaded by default when using this style file; however, you should be aware which packages are being used so that you know which documentation to reference for help beyond this quick start up guide.

To get us started, I am going to briefly introduce both the inline and display math environments that are available. Then I will provide some of the more common features available from the $\mathcal{A}\mathcal{M}\mathcal{S}$ math package. Once we have established the environments in which we can write mathematics, we can look at how we can format special characters and symbols such as operators, operands and function names. To make our lives a bit easier, we will include some additional packages along the way and I will share some of the custom macros that are available through the `mhomath.sty` file.

§2.1 Inline Mathematics

In order for the code that you write to be interpreted by the compiler as mathematics, you need to toggle between **normal mode** and **inline math mode** by using the `math` environment.

```
\begin{math}  
  \langle \textit{mathematics content goes here} \rangle  
\end{math}
```

```
Solve all equations of the form
\begin{math}
  ax^2 + bx + c = 0
\end{math}
for
\begin{math}
  x \in \mathbb{R}
\end{math}
by completing the square.
```

Solve all equations of the form $ax^2 + bx + c = 0$ for $x \in \mathbb{R}$ by completing the square.

From the above example, we can see that the code is verbose and difficult to read. Thankfully, this environment has a shortcut notation that should **ALWAYS** be used. The macro `\(` is used to toggle into inline math mode and the macro `\)` is used to toggle back to normal mode.

```
< normal mode text > \( < math environment code > \) < normal mode text >
```

We can now revisit our original example using this simplified notation.

```
Solve all equations of the form \( ax^2 + bx + c = 0 \)
for \( x \in \mathbb{R} \) by completing the square.
```

Solve all equations of the form $ax^2 + bx + c = 0$ for $x \in \mathbb{R}$ by completing the square.

It is helpful to know that spaces in math mode are ignored by the compiler, which means we can be generous with how we use spaces to format our code such that it is easier to parse. As your expressions become more complex, the additional spaces in your code should improve its legibility. Well, at least that has been my experience. Let's consider our example without the extra spaces.

```
Solve all equations of the form \(ax^2+bx+c=0\) for
\(x\in\mathbb{R}\) by completing the square.
```

Solve all equations of the form $ax^2 + bx + c = 0$ for $x \in \mathbb{R}$ by completing the square.

The shortcut inline math environment, `\(<math environment code> \)`, that we have been using is from \LaTeX . However, there is a plain \TeX shortcut for the math environment that remains popular today which is using the dollar sign character as the delimiter.

```
< normal mode text > $ < math environment code > $ < normal mode text >
```

I prefer the `\(<math environment code> \)` notation as it can be helpful in resolving errors. However, if you are new to using \LaTeX , then you might want to use the dollar sign delimiters as they have the advantage of making your code slightly easier to read and faster to type. At this point in the development of \LaTeX as a language, it is really a matter of personal preference. Just be consistent!

```
Solve all equations of the form $ ax^2 + bx + c = 0 $ for
$ x \in \mathbb{R} $ by completing the square.
```

Solve all equations of the form $ax^2 + bx + c = 0$ for $x \in \mathbb{R}$ by completing the square.

A small thought on changing between the two different shortcut notations. It would be possible to do a quick search and replace from the \LaTeX notation to the \TeX notation, but not vice-versa.

§2.2 Display Style Math Mode

There will be times when you will want to emphasize or isolate an expression by centering it on the page and for this we are going to be using the **display style math mode** environment. This environment is restricted to a single line of output like an equation.

```
\begin{displaymath}
  <mathematics content goes here>
\end{displaymath}
```

All the real number values of x that satisfy the quadratic equations of the form $ax^2 + bx + c = 0$ where $a, b, c \in \mathbb{R}$ can be found using the quadratic formula,

```
\begin{displaymath}
  x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.
\end{displaymath}
```

All the real number values of x that satisfy the quadratic equations of the form $ax^2 + bx + c = 0$ where $a, b, c \in \mathbb{R}$ can be found using the quadratic formula,

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Just like the inline math environment, you would **ALWAYS** use the \LaTeX pair of $\left[\langle \textit{math environment code} \rangle \right]$ delimiters as the shortcut. Rather than write the delimiters and mathematical expression on the same line as we did in inline math mode, we give each delimiter its own line and place the math on the line in between the delimiters. We could express the quadratic formula from our previous examples as:

```
\[
  x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.
\]
```

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

It is very likely that you will come across the the double dollar sign shortcut notation for the display math environment, but it should **NEVER** be used as it will most likely bring up issues at some point.

```
$$
  x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.
$$
```

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

If you are looking to include multiple lines in display math mode, then you might be tempted to use consecutive display style math mode environments, but you must resist as a better solution is available. In summary, you should **NEVER** use consecutive display math mode environments, which means you should never write something like:

```
\[
  2x^2 + 14x + 24 = 2( x^2 + 7x + 12 )
\]
\[
  = 2(x + 3)(x + 3)
\].
```

$$2x^2 + 14x + 24 = 2(x^2 + 7x + 12)$$

$$= 2(x + 3)(x + 3)$$

.

§2.3 Multiline Math Environments

Like I said, there are specific environments that you can use to achieve multiline equations. This style file includes the package `mathtools`, which will give us multiple environments to achieve multiline equations and more. The `mathtools` package imports the \mathcal{AMS} math package and enhances its functionality. This means if we are using `mathtools`, then there is no need to import \mathcal{AMS} math.

For complete documentation I suggest you check the official documentation:

- User's Guide for the `amsmath` Package
- The `mathtools` package

Before we look at the using multiline equation environments, I want to point out the single line math environment called `equation`, which is similar to the \LaTeX display math environment

```
\begin{equation}
  \langle \textit{mathematics content goes here} \rangle
\end{equation}
```

The first environment that we are going to consider is very similar to the the \LaTeX display math environment, but it also supports multiline equations. By default the equations will be numbered.

```
\begin{equation}
  x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.
\end{equation}
```

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \quad (2.1)$$

In fact, we get something very close to the \LaTeX display math environment with the star version of this equation environment as the equation numbers are suppressed. All these structured math environments have a star version that suppresses numbering, so I will only include an example for the `equation` environment.

```
\begin{equation*}
  \langle mathematics content goes here \rangle
\end{equation*}
```

```
\begin{equation*}
  x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.
\end{equation*}
```

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

It's time to take a look at some of the multiline equation environments that are available for us to use in our work. To change the form of the expression $2x^2 + 14x + 24$ using the equation environment.

```
\begin{equation*}
  2x^2 + 14x + 24 = 2( x^2 + 7x + 12 ) = 2(x + 3)(x + 3)
\end{equation*}
```

$$2x^2 + 14x + 24 = 2(x^2 + 7x + 12) = 2(x + 3)(x + 3)$$

However, this might be easier to read if it was expressed using a multiline math environment.

It is possible include a split environment within the equation environment to achieve a multiline equation with the option to include expression splitting as well.

```
\begin{split}
  \langle mathematics content goes here \rangle
\end{split}
```

We will use the & character to vertically *align* each equation at the equal sign and two consecutive double backslash characters, `\\` break to the next line.

```
\begin{equation*}
  \begin{split}
    2x^2 + 14x + 24 \\
    &= 2( x^2 + 7x + 12 ) \\
    &= 2(x + 3)(x + 3)
  \end{split}
\end{equation*}
```

$$\begin{aligned} 2x^2 + 14x + 24 &= 2(x^2 + 7x + 12) \\ &= 2(x + 3)(x + 3) \end{aligned}$$

It is even possible to split up long expressions! The `\quad` macro is used to include 4 spaces.

```

\begin{equation*}
\begin{split}
a
&= b - c + d - e + f \\
&\quad - g + h - i + k \\
&= l + m \\
&= n
\end{split}
\end{equation*}

```

$$\begin{aligned}
 a &= b - c + d - e + f \\
 &\quad - g + h - i + k \\
 &= l + m \\
 &= n
 \end{aligned}$$

It might be the case that you have one very long expression that you would like to break up and this can be done with the `multline` environment. Notice that this does not require to be embedded in the `equation` environment.

```

\begin{multiline}
\langle \textit{mathematics content goes here} \rangle
\end{multiline}

```

```

\begin{multline*}
a + b + c + d + e + f + g + h + i + j + k + l + m + n + o + p + q + r + s + t + u + v + w + x + y + z \\
- (a + b + c + d + e + f + g + h + i + j + k + l + m + n + o + p + q + r + s + t + u + v + w + x + y + z)
\end{multline*}

```

$$\begin{aligned}
 &a + b + c + d + e + f + g + h + i + j + k + l + m + n + o + p + q + r + s + t + u + v + w + x + y + z \\
 &- (a + b + c + d + e + f + g + h + i + j + k + l + m + n + o + p + q + r + s + t + u + v + w + x + y + z)
 \end{aligned}$$

My preference is to have my equations to be vertically aligned at the equal sign. However, you might want your equations to be centered on the page. The `gather` environment will take care of this for you.

```

\begin{gather}
\langle \textit{mathematics content goes here} \rangle
\end{gather}

```

```
\begin{gather*}
  a = b + c + d + e \\
  a = f + g \\
  a = h
\end{gather*}
```

$$\begin{aligned}
 a &= b + c + d + e \\
 a &= f + g \\
 a &= h
 \end{aligned}$$

Notice that there is not & character being used for alignment in this environment.

It is now time to introduce the environment that I use for about 95 % of the multiple line expressions needed in my work and it is called align.

```
\begin{align*}
  \langle \textit{mathematics content goes here} \rangle
\end{align*}
```

```
\begin{align*}
  2x^2 + 14x + 24 \\
  &= 2( x^2 + 7x + 12 ) \\
  &= 2(x + 3)(x + 3).
\end{align*}
```

$$\begin{aligned}
 2x^2 + 14x + 24 &= 2(x^2 + 7x + 12) \\
 &= 2(x + 3)(x + 3).
 \end{aligned}$$

This environment allows you to comment you work using tags

```
\begin{align*}
  2x^2 + 14x + 24 \\
  &= 2( x^2 + 7x + 12 ) \tag{factoring two} \\
  &= 2(x + 3)(x + 3). \tag{factoring the quadratic}
\end{align*}
```

$$\begin{aligned}
 2x^2 + 14x + 24 &= 2(x^2 + 7x + 12) && \text{(factoring two)} \\
 &= 2(x + 3)(x + 3). && \text{(factoring the quadratic)}
 \end{aligned}$$

There is no need for multiple align environments in a sentence since there is the `\intertext` macro. If you want less space between the lines, then I can suggest that you use the `\shortintertext` macro instead.

```

Given \(\ x = \cos x \\) and \(\ y = \sin x \), we can show
\begin{align*}
\shortintertext{using the Pythagorean identity}
x^2 + y^2 &= 1 \\
\intertext{substituting \(\ x \) and \(\ y \)}
\left( \cos x \right)^2 + \left( \sin x \right)^2 &= 1 \\
\shortintertext{alternative function squared form}
\cos^2 x + \sin^2 &= 1.
\end{align*}

```

Given $x = \cos x$ and $y = \sin x$, we can show

using the Pythagorean identity

$$x^2 + y^2 = 1$$

substituting x and y

$$(\cos x)^2 + (\sin x)^2 = 1$$

alternative function squared form

$$\cos^2 x + \sin^2 = 1.$$

It is also possible to include two columns of equations

```

\begin{align*}
a_{11} \\
&= b_{11} \\
& \\
a_{12} \\
&= b_{12} \\
a_{21} \\
&= b_{21} \\
& \\
a_{22} \\
&= b_{22} + c_{22}
\end{align*}

```

$$a_{11} = b_{11}$$

$$a_{21} = b_{21}$$

$$a_{12} = b_{12}$$

$$a_{22} = b_{22} + c_{22}$$

If you would like the columns to be fully justified, then there is an alternative environment called `flalign`.

```

\begin{flalign*}
\langle \textit{mathematics content goes here} \rangle
\end{flalign*}

```



```
\begin{flalign*}
  a_{11}
&=b_{11}
&
  a_{12}
&=b_{12}\\
  a_{21}
&=b_{21}
&
  a_{22}
&=b_{22}+c_{22}
\end{flalign*}
```

$$a_{11} = b_{11}$$

$$a_{21} = b_{21}$$

$$a_{12} = b_{12}$$

$$a_{22} = b_{22} + c_{22}$$

§2.4 Cases Environment

There are a couple cases where you might want to treat some part of your formula as a block on the same line and the cases environment get the job done.

```
\begin{cases}
  \langle \textit{mathematics content goes here} \rangle
\end{cases}
```

```
\[
\abs{x}=\begin{cases}
  x & \text{if } (x \geq 0) \\
  -x & \text{if } (x < 0)
\end{cases}
\]
```

$$|x| = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{cases}$$

This can also be used for systems of equations.

```
Solve the following system of equations for ( x ) and ( y ).
\[
\begin{cases}
  x + y = 12 \\
  2x - 3y = 9
\end{cases}
\]
```

Solve the following system of equations for x and y .

$$\begin{cases} x + y = 12 \\ 2x - 3y = 9 \end{cases}$$

3 Figures & Images

§3.1 Storing Images

Images can be easily included in your document by using the `\includegraphics` macro included with the `graphics` package. Before we include our first image, we should probably create a folder to hold all our images to keep our directory lean in terms of the number of files. My personal preference is to keep my images in the folder called `~/assets`, which is relative to the directory of my main file. It is also possible to include subdirectories as well.

Rather than have to provide the path of where your images are located there is a `\graphicspath` macro that you can place in the preamble that allows you to assign one or more locations for \LaTeX to look for image files.

```
\graphicspath{{/assets/}}
```

Or if you have more than one directory

```
\graphicspath{{/assets/}{/images/}{/pdfs/}}
```

This will save you a lot of extra typing and worry about where your images are being stored.

§3.2 Including Images

To include an image we will be using the `\includegraphics` macro, which accepts the path and file name as its argument and can take multiple options. If you made use of the `\graphicspath` macro, made available by default for this template, then the argument is the file name. The file extension is optional, unless you have multiple files with the same name, but with different extensions. Let's scale an image such that it's width is a quarter of the `\textwidth` and center it by placing it in a `center` environment.

```
\begin{center}  
\includegraphics[width = 0.25\textwidth]{image.jpg}  
\end{center}
```



§3.3 Figure Environment

You will probably want to reference your image at some point in your text and will most likely want to provide it with a caption. For this we are going to use the figure environment, which will calculate the best position to place your image within the document. Don't worry, there are ways to manually place the image where you would like it to appear.

```
\begin{figure}[\langle h, t \rangle]
  \langle image content goes here \rangle
\end{figure}
```

```
\begin{center}
\includegraphics[width = 0.25\textwidth]{image.jpg}
\end{center}
\caption{Here is my excellent image.}
\label{fig:img###}
```



Figure 3.1: Here is my excellent image.

This would generate figure 3.1.

§3.4 Multiple Horizontally Aligned Images

Quite often it is the case that we want to include multiple horizontally aligned images. The figure environment is not well equipped for this task, so we are going to make use of the subfigure package which will give us the subfigure environment that we can embed in our figure environment. This will enable us to assign both a caption and a label to each image separately.

```
\begin{subfigure}
  \langle image content goes here \rangle
\end{subfigure}
```

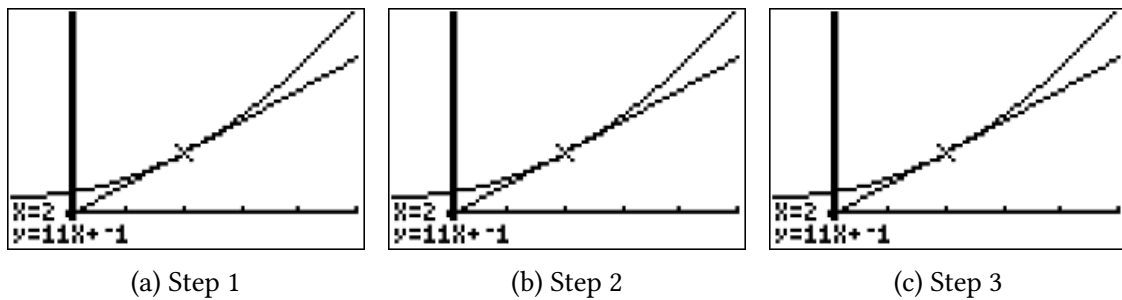


Figure 3.2: The Steps Shown

```

\begin{figure}
  \begin{center}
    \begin{subfigure}[b]{0.3\textwidth}
      \includegraphics[width=\textwidth]{20170509-123642}
      \caption{Step 1}
      \label{fig:step1a}
    \end{subfigure}
    ~ % > > Add desired spacing between images, e. g. ~, \quad, \qquad,
      % \hfill etc. (or a blank line to force the subfigure onto
      % a new line)
    \begin{subfigure}[b]{0.3\textwidth}
      \includegraphics[width=\textwidth]{20170509-123642.png}
      \caption{Step 2}
      \label{fig:step2a}
    \end{subfigure}
    ~ % > > Add desired spacing between images, e. g. ~, \quad, \qquad,
      % \hfill etc. (or a blank line to force the subfigure onto
      % a new line)
    \begin{subfigure}[b]{0.3\textwidth}
      \includegraphics[width=\textwidth]{20170509-123642.png}
      \caption{Step 3}
      \label{fig:step3a}
    \end{subfigure}
    \caption{The Steps Shown}\label{fig:threestepsa}
  \end{center}
\end{figure}

```


4 Theorem Environments

§4.1 Definition

Definition 2 (Definition Of Subtraction). Given a and b are real numbers, then[4] [3]

$$a + (-b) = a - b.$$

§4.2 Examples

Example 2. Solve the equation $x + 4 = 7$ for all $x \in \mathbb{Z}$.

Solution. The solution set is $x = \{3\}$. □

§4.3 Theorems

Axiom 1 (Axiom Of Simple Mathematical Induction). Let $P(n)$ be a proposition, where $n \in \mathbb{Z}^+$. If we can

1. show the basis step is TRUE by showing $P(1)$ is TRUE and
2. show the inductive step is TRUE by showing for each $r \in \mathbb{Z}^+$, whenever $P(r)$ is TRUE, then $P(r + 1)$ is TRUE, where $P(r)$ is called the **induction hypothesis**,

then by the **axiom of mathematical induction** the proposition is proved.

The axiom of simple mathematical induction cannot be proven as it is part of the definition of \mathbb{Z}^+ .

The axiom of simple mathematical induction¹ cannot be proven as it is included as part of the definition of the set of positive integers, \mathbb{Z}^+ .

Proposition 1 (Product of Common Base Powers). Given b^m and b^n , then

$$b^m \cdot b^n = b^{m+n}.$$

Proof. Coming Soon! ■

Theorem 1 (Pythagorean Theorem). Given a right-angled triangle ABC with sides a, b

and c , where c is the hypotenuse, then

$$a^2 + b^2 = c^2.$$

Proof. Someday soon!



References

- [1] *CTAN: User's Guide for the amsmath Package (Version 2.1)*. American Mathematical Society, LATEX Project, \LaTeX project. 2020. URL: <https://ftpmirror1.infania.net/mirror/CTAN/macros/latex/contrib/mathtools/mathtools.pdf> (visited on 07/02/2021).
- [2] Lars Madsen Morten Hogholm. *CTAN: The mathtools package*. \LaTeX 3 project. 2021. URL: <https://ftpmirror1.infania.net/mirror/CTAN/macros/latex/contrib/mathtools/mathtools.pdf> (visited on 07/02/2021).
- [3] Mark Olson. *02AQ - How to Factor a Quadratic Expression*. 2021. URL: <https://youtu.be/3ftBMUrSJtg> (visited on 06/08/2021).
- [4] Eric W. Weisstein. *List*. From MathWorld—A Wolfram Web Resource. Wolfram Research, Inc. Aug. 5, 2003. URL: <https://mathworld.wolfram.com/List.html> (visited on 07/31/2020).