

# **Pre-Calculus in Brief with GitHub and L<sup>A</sup>T<sub>E</sub>X PCiB - Version 0.1**

November 7, 2023

MIT License

Available on GitHub at:  
<https://GitHub.com/nicholaskarlson/PCiB>



# Contents

Preface	5
1 Introduction to PCiB	7
2 Open-Source Ethos	9
3 Introduction to GitHub	11
4 Encouragement to Fork	13
5 More About GitHub	15
6 Forking Process	19
7 Editing and Customizing	23
8 Engaging with the Community	27
9 Pre-Calculus Basic Topics	31
10 Topics in Geometry	33
11 Topics in Algebra	35
12 Topics in Trigonometry	37
Appendices	38
I Basic GitHub Guide	39
II Basic L <sup>A</sup> T <sub>E</sub> X Guide	43
Bibliography	45



# Preface

In exploring our past and understanding our roots, *Pre-Calculus in Brief - PCiB - Version 0.1* aspires to be more than just another math book. This book strives to foster collaborative math writing. Note that this book has very few references. The reader is encouraged to use resources available on the Web to fact-check. This book’s view on “causation” and facts is heavily influenced by Mosteller and Tukey [MT77].

## Redefining the Role of the Reader

Pre-Calculus in Brief (PCiB) is an endeavor to reshape how math is written, understood, and studied. It’s not just a passive read but an open-source approach to math, aiming to encourage students to become proactive learners.

This project strives to break the traditional mold of math education and invites readers and professional mathematicians alike to participate actively.

## A Dynamic Relationship with Math

*Pre-Calculus in Brief* is not just a book but a movement and methodology, heralding a new era in how we approach, consume, and interact with math. By positioning the reader as an integral part of the math-book process, PCiB fosters a dynamic relationship with math, making mathematics more accessible, proactive, and relevant. In this shifting paradigm, we are all potential mathematicians, creators of interesting and relevant ways to learn and study math.

Please fork the LaTeX source code for PCiB (available on GitHub) and create your own book that chooses the facts and exercises most relevant to you! Also, starring the PCiB project on GitHub would be greatly appreciated! Thanks for reading PCiB!



# Chapter 1

## Introduction to PCiB

### **Welcoming the World of Historical Investigation with GitHub**

World History in Brief, abbreviated PCiB, isn't merely a passive read. It's an endeavor to reshape how math is written, studied, and taught. By presenting an open-source approach to math, the goal is to include everyone.

### **Fostering a Proactive Engagement with Math**

*World History in Brief* isn't merely a book—it's a clarion call for a renewed engagement with mathematics. Pre-Calculus in Brief (PCiB) is an endeavor to reshape how math is written, understood, and studied. It's not just a passive read but an open-source approach to math, aiming to encourage students to become proactive learners.

This project strives to break the traditional mold of math education and invites readers and professional mathematicians alike to participate actively.

Please fork the L<sup>A</sup>T<sub>E</sub>X source code for PCiB (available on GitHub) and create your own book on Pre-Calculus that chooses the content most relevant to you! Also, starring the PCiB project on GitHub would be greatly appreciated! Thanks for reading PCiB!





# Chapter 2

## Open-Source Ethos

### **The Spirit of Shared Knowledge and Collaboration**

Math, like software, is better when it's open. PCiB draws inspiration from the open-source software movement; this section elucidates how a collaborative, transparent, and shared approach can enhance our understanding of math. Here, we look at the philosophy behind open-source and how it beautifully marries with the study of mathematics.

### **Open-Source Math: Preserving Tradition Through Collaborative Exploration**

Mathematics, like software, thrives when it embraces openness and transparency. PCiB takes a leaf from the proven benefits of the open-source software model; this section highlights how a collaborative and transparent method can improve and deepen our grasp of math and its texts. Here, we delve into the principles of open source and how they align with a thorough exploration mathematics and its texts.

### **Understanding the Open-Source Ethos**

The open-source paradigm revolves around shared ownership, collaboration, and the free exchange of knowledge. In the software realm, this approach has led to groundbreaking innovations built and enhanced by a global community of skilled contributors. United by a mutual objective, these individuals pool their diverse talents and insights to improve and share software solutions for broader public benefit.

## Advantages of the Open-Source Framework in Math

### Collective Insight

Mirroring the collaborative essence of open-source software, many individuals can offer their perspectives and knowledge, making math texts more robust and varied.

### Enhancement and Accuracy

Open platforms foster an environment of constructive criticism, ensuring prompt identification and correction of inaccuracies. This meticulous peer review can help provide a credible and current mathematical text.

### Universal Access

Much as open-source software promotes free access and modification, open-source math prioritizes universal accessibility. This ensures mathematics knowledge isn't restricted to a select few but is available to all curious minds.

## Potential Challenges

Despite its advantages, melding open-source with math has potential pitfalls. The volume of contributions can complicate accuracy verification processes.

However, the very community championing this open-source approach to math can serve as its vigilant protector. They can ensure that contributions undergo rigorous evaluation and referencing, akin to the meticulous checks within the open-source software community.

## Conclusion: Reinvigorating Our Experience with Math

Adopting an open-source perspective to approaching math signifies a refreshed approach. It beckons a worldwide community to collaborate and forge a comprehensive and exciting math text. In this refreshed approach, every individual can play a part, both as a contributor and a learner. Math texts, through this lens, evolve and flourish, reflecting the collective input of active participants.

# Chapter 3

## Introduction to GitHub

### The Hub for Modern Collaboration

#### **Harnessing GitHub: A New Frontier in Collaborative Math Writing**

At the heart of our collaborative math endeavor lies GitHub, a platform traditionally associated with code but now repurposed for our endeavor. This section provides a primer on GitHub, laying the foundation for those unfamiliar and offering insights into its transformative potential for collective math writing, learning, and teaching.

#### **A Brief Introduction to GitHub**

Originally conceptualized as a platform for developers, GitHub is a repository hosting service that facilitates version control using Git. At its core, it allows multiple users to work on a project simultaneously, tracking changes and ensuring that the latest version of a project is always accessible. Over the years, GitHub has grown beyond its initial software-centric confines, becoming a hub for all kinds of collaborative projects, from writing to data science and now to math.

#### **Repurposing GitHub for Historical Narratives**

##### **Version Control**

Math writing, like software, is dynamic and constantly evolving. As new sources or perspectives emerge, math texts may need revisions. GitHub's

version control ensures that every change made to a document is tracked, enabling mathematicians to see how math texts evolve over time.

### **Collaborative Writing**

Multiple contributors can work on a single math text simultaneously. This multi-user capability ensures diverse viewpoints can be seamlessly integrated, making the math text richer and more comprehensive.

### **Review and Feedback**

Just as developers review and comment on code, mathematicians can provide feedback on written content. This feature encourages rigorous peer review, ensuring accuracy and credibility.

### **Open Access**

Math texts on GitHub can be made public, granting anyone access to read, contribute, or fork the text into their own versions. This workflow democratizes math texts, making the creation process a collective endeavor rather than the domain of a select few.

### **Transparency**

All changes and contributions are logged, providing a clear trail of the evolution of a mathematical text. This transparency bolsters the credibility of the text hosted on the platform.

### **Community Building**

Beyond just writing, GitHub fosters a community of mathematicians, enthusiasts, and readers who can discuss, debate, and engage in meaningful dialogues about the math and available math texts on GitHub.

## **Conclusion: Envisioning a Collaborative Mathematical Landscape**

Embracing GitHub as a tool for collaborative math signifies more than just a shift in approach; it heralds a new era of inclusivity, transparency, and dynamism in writing, learning, and the teaching of math.

# Chapter 4

## Encouragement to Fork

### Invitation to Dive Deep and Make It Your Own

PCiB isn't a static entity. It thrives on evolution, adaptation, and diversification, much like math itself. We encourage readers to "fork" - a term you'll soon become intimately familiar with - and create their own versions of this book. Read this section to understand the essence of "forking" and how it can be the starting point of your unique math journey.

### The Concept of Forking: A Brief Overview

In the realm of software development, particularly in platforms like GitHub, "forking" refers to the act of creating a copy of a project, allowing one to make changes independently of the original. In this context, forking PCiB enables readers to take the base content and adapt, modify, and expand upon it, tailoring the narrative to resonate with their perspectives, insights, and understanding.

### How to Begin Your Forking Journey

**Start Small:** You don't need to rewrite entire chapters. Begin by adding annotations, insights, or even footnotes to existing content. As you grow more confident, you can expand and modify larger sections.

**Engage with the Community:** Share your forked version with other readers. This encourages discourse, debate, and constructive feedback, allowing your text to be refined and enhanced.

**Celebrate input:** Encourage others around you to fork and create their own versions. The more indepth the input, the deeper our collective understanding of math potentially becomes.

**Conclusion: The Power of Collective Math**

The invitation to fork PCiB isn't just about creating different versions of a book. It's a call to embrace collective writing, learning, and teaching. By embracing the essence of forking, math is not just something we read but something we actively shape, share, and pass on.

# Chapter 5

## More About GitHub

### Discovering the Power of Collaborative Tools

Diving deeper into the world of GitHub, this chapter provides a comprehensive overview. Beyond its technicalities, we explore how GitHub emerged as a revolutionary platform for collaboration and how it can be leveraged for historical research and narrative building.

### The Genesis of GitHub

GitHub began as a platform designed for software developers to manage and track changes to their codebase. Launched in 2008, it swiftly gained traction due to its user-friendly interface and efficient version control system powered by Git. Over the years, it evolved from a mere repository hosting service to a dynamic hub of collaboration, housing millions of projects and engaging tens of millions of users worldwide.

### GitHub: More than Just Code

While GitHub's origins are rooted in code collaboration, its adaptable nature has made it a favored platform for various non-code projects. Writers, designers, educators, and researchers have discovered the potential of GitHub as a tool for:

#### Document Collaboration

With its built-in version control, contributors can track changes, revert to previous versions, and seamlessly merge updates.

### **Project Management**

With features like "issues" and "milestones," teams can organize tasks, set goals, and monitor progress.

### **Open Access & Transparency**

Public repositories allow for open contributions, ensuring transparency and fostering a sense of collective ownership.

### **Collaborative Writing**

Multiple contributors can simultaneously work on a single document, with every change being tracked and attributed, facilitating teamwork on extensive projects like books or research papers.

### **Engaging the Public**

With the platform's inherent transparency, researchers can make their work-in-progress accessible to the public, inviting insights, corrections, and contributions.

### **Case Study: PCiB's Use of GitHub**

PCiB's journey on GitHub is a testament to the platform's potential in mathematical endeavors. By hosting the book on GitHub, the following is possible:

#### **Feedback Loop**

Readers can raise "issues," pointing out inaccuracies, suggesting enhancements, or even recommending new sections or topics.

#### **Forking**

As previously discussed, readers can "fork" the repository, creating their unique versions of the book while staying connected to the original.

#### **Regular Updates**

With math being dynamic, the book can be regularly updated, with new versions being released as and when significant changes are incorporated.



## **Challenges and Considerations**

While GitHub offers many advantages, it's essential to understand its limitations:

### **Learning Curve**

For those unfamiliar with Git or version control, there can be an initial learning curve.

### **Data Overwhelm**

With vast amounts of data and contributions, ensuring quality and accuracy can be challenging.

### **Diverse Audience Management**

Catering to both tech-savvy and non-tech audiences might require creating additional resources or tutorials to ensure inclusivity.

## **Conclusion: GitHub – A Paradigm Shift in Collaboration**

The rise of GitHub marks a significant shift in how we perceive and participate in collaborative projects. Its adaptability, transparency, and user-centric design make it a powerful tool, not just for coders but for anyone passionate about collective endeavors. In the realm of mathematics, GitHub promises a future where texts are continually refined, expanded, and enriched by a global community.



# Chapter 6

## Forking Process

### The Heart of Collaboration on GitHub

The beauty of open-source lies in its democratization of content creation. In this section, we demystify the process of "forking" on GitHub, guiding you step-by-step on how to take PCiB and create a version uniquely yours.

### Understanding Forking

Before diving into the specifics, it's crucial to understand what "forking" means in the context of GitHub. In the simplest terms, to "fork" a project means to create a personal copy of someone else's project. Forking allows you to freely experiment with changes without affecting the original project. Forking is akin to taking a book you admire and making a copy to write your notes, edits, or additional chapters without altering the original book.

### Why Fork?

#### Experimentation

It provides a safe space where you can test out ideas, make changes, or introduce new content.

#### Personalization

For projects like PCiB, it allows readers to customize the content, tailor it to their perspectives, or even localize it for specific audiences.

## **Collaboration**

If you believe your changes have broad appeal, you can propose that they be incorporated back into the original project, enriching it with your unique contributions.

## **Step-by-Step Forking Guide**

### **Set Up Your GitHub Account**

If you don't have an account on GitHub, you'll need to create one. Visit GitHub's official site and sign up.

### **Navigate to the PCiB Repository**

Once logged in, search for the PCiB project or navigate to its URL directly.

### **Click the 'Fork' Button**

The fork button is located at the top right corner of the repository page; this button will create a copy of PCiB in your account.

### **Clone Your Forked Repository**

Forking allows you to have a local copy on your computer, making editing and experimentation easier. Use the command: `git clone [URL of your forked repo]`.

### **Make Your Changes**

Using your preferred tools, introduce the edits, additions, or modifications you desire.

### **Commit and Push Changes**

Once satisfied, save these changes (known as a "commit") and then "push" them to your forked repository on GitHub.

### **Optional – Create a Pull Request**

If you believe your changes should be incorporated into the original PCiB repository, you can create a "pull request." A pull request notifies the original authors of your suggestions.

## **Things to Keep in Mind**

### **Stay Updated**

The original PCiB project may undergo updates. It's a good practice to regularly "pull" from the original repo to keep your fork up-to-date.

### **Engage with the Community**

Open-source thrives on community interactions. Engage in discussions, seek feedback, and please remain open to constructive criticism.

## **Conclusion: Embracing the Forking Culture**

Forking is more than just a technical process; it symbolizes the ethos of open-source — a world where knowledge is not hoarded but shared, refined, and built upon collectively. By forking PCiB or any other project, you're not just creating a personal copy; you're becoming a part of a global movement that values collaboration, innovation, and the shared pursuit of knowledge. So, embark on this journey, make your unique mark, and contribute to the ever-evolving corpus of collective wisdom.



# Chapter 7

## Editing and Customizing

### Tailoring Repositories to Suit Your Needs

Now, let's build upon the forking process; this segment delves into the next steps. How can you edit and customize your version of PCiB? What tools and techniques are available at your disposal? Embark on this informative journey as we guide you through the intricacies of editing on GitHub.

### Understanding the GitHub Workspace

Before diving into the specifics of editing, it's essential to familiarize yourself with the GitHub workspace. Think of it as a digital toolshed where each tool serves a unique function:

- **Repository (Repo):** This is the project's main folder where all your project's files are stored and where you track all changes.
- **Branches:** These are parallel versions of a repository, allowing you to work on features or edits without altering the main project.
- **Commits:** This is a saved change in the repository, akin to saving a file after making edits.
- **Pull Requests:** This is how you notify the main project of desired changes, proposing that your edits be merged with the original.

### Editing Files Directly on GitHub

For minor changes, you might opt to edit directly on GitHub:

1. **Navigate to the File:** Within your forked PCiB repository, find the file you want to edit.
2. **Click the Pencil Icon:** This button allows you to edit the file.
3. **Make Your Edits:** Modify the content as needed.
4. **Save and Commit:** Below the editing pane, you'll see a "commit changes" section. Add a brief note summarizing your changes and click 'Commit.'

## Editing Files Locally

For extensive customization:

1. **Clone Your Repository:** Use a tool like Git to clone (download) your forked repo to your local computer.
2. **Edit Using Your Preferred Tools:** This could range from text editors to specialized software, depending on the file type.
3. **Commit and Push:** After making your changes, save them (commit) and then upload (push) them to your GitHub repository.

## Utilizing Branches for Extensive Customization

Branches are especially useful for significant overhauls or when working on different versions:

1. **Create a New Branch:** From your main project page, use the branch dropdown to type in a new branch name and create it.
2. **Switch to Your Branch:** Ensure you're working in this new parallel environment.
3. **Make and Commit Changes:** As you would in the main project.
4. **Merging:** Once satisfied with your edits in the branch, you can merge these changes back into the main project or keep them separate as a different version.



## Exploring Additional Tools and Extensions

GitHub's ecosystem is rich with tools and extensions to enhance your editing experience:

- **GitHub Desktop:** An application that simplifies the process of managing your repositories without using command-line tools.
- **Markdown Editors:** Since many GitHub files (like READMEs) are written in Markdown, tools like StackEdit or Dillinger can be invaluable.
- **Extensions for Browsers:** Tools like Octotree can help in navigating repositories more effortlessly.

## Conclusion: The Art of Tailored Content

Editing and customizing on GitHub might seem daunting initially, but with practice, it transforms into a manageable workflow. Many people find that the ability to take a project like PCiB and mold it into something uniquely theirs is empowering. It's a testament to the open-source community's ethos, where shared knowledge becomes the canvas and our collective edits, the brushstrokes, crafting an ever-evolving masterpiece. As you embark on your customization journey, remember that every edit, no matter how small, contributes to the project potentially in significant ways.



# Chapter 8

## Engaging with the Community

### Joining the Global Conversation

#### The Significance of the GitHub Community

The digital age has bestowed upon us the gift of connectivity. On platforms like GitHub, this connectivity transcends borders, disciplines, and ideologies, culminating in a melting pot of diverse ideas and knowledge. For mathematicians and math enthusiasts, GitHub offers a space not only to store and manage content but also to engage with an audience that is passionate, informed, and eager to contribute.

#### 1. Discussions and Debates

One of the most enriching aspects of the GitHub community is the plethora of discussions that unfold:

- **Issues:** A core feature of GitHub, "issues" allow users to raise questions, report problems, or propose enhancements.
- **GitHub Discussions:** A newer feature, Discussions, acts like a community forum. It's an excellent place for extended conversations, brainstorming, and sharing ideas or resources.

#### 2. Collaborative Content Creation

Beyond solitary endeavors, GitHub shines in its collaborative capabilities:

- **Pull Requests:** If you've made an alteration to a math text or added a new perspective, pull requests are the way to propose these changes to

the original repository owner. Pull requests foster a collaborative spirit, where content isn't static but continually evolving with community input.

- **Fork and Merge:** As you've learned, forking allows you to create your version of a repository. Engaging with the Community means you can merge changes from others into your fork, blending a mixture of diverse insights.

### 3. Building and Nurturing Networks

Connections made on GitHub often spill over into lasting professional relationships:

- **Following and Followers:** Like on social media platforms, you can follow contributors whose work resonates with you. Following contributors creates a curated feed of updates and also allows you to be part of a more extensive network.
- **GitHub Stars:** If a particular project or repository impresses you, give it a star! Starring not only bookmarks the project for you but also shows appreciation to the creator.

### 4. Learning and Growing Through Feedback

The Community's feedback is an invaluable asset:

- **Code Reviews:** Although traditionally for software, text writers can use this feature to receive feedback on their methodologies or approaches, refining their work.
- **Community Insights:** The "insights" tab on a repository provides analytics. For text writers, this can give a sense of which topics garner more attention and interest.

### 5. Participating in Community Events

GitHub often hosts and sponsors events:

- **Hackathons:** While traditionally for coders, these events can be repurposed for text writer content creation, where participants collaboratively tackle projects or themes.
- **Webinars and Workshops:** These events can range from mastering GitHub's technical side to thematic discussions on math topics.

## **A Project of Collective Wisdom**

Math, in many ways, is a collective endeavor. GitHub can provide a dynamic Community. By engaging with this Community you can become an active participant in the creation of mathematical texts.



# Chapter 9

## Pre-Calculus Basic Topics

### Introduction

In this chapter, we start talking about actual pre-calculus topics. Here we will introduce geometry, algebra, and trigonometry with details provided in later chapters.





# Chapter 10

## Topics in Geometry

### Understanding Angles and Triangles

#### Introduction

Geometry is important in pre-calculus studies.

#### Conclusion

This chapter surveys important topics in geometry relevant for the study of calculus. Please fork the LaTeX source code for PCiB (available on GitHub) and create your own book that chooses the facts and exercises most relevant to you! Also, starring the PCiB project on GitHub would be greatly appreciated! Thanks for reading PCiB!



# Chapter 11

## Topics in Algebra

### Understanding Basic Number System

#### Introduction

Algebra is important in pre-calculus studies.

#### Conclusion

This chapter surveys important topics in algebra relevant for the study of calculus. Please fork the LaTeX source code for PCiB (available on GitHub) and create your own book that chooses the facts and exercises most relevant to you! Also, starring the PCiB project on GitHub would be greatly appreciated! Thanks for reading PCiB!



# Chapter 12

## Topics in Trigonometry

### Understanding the Basic Trig Functions

### Introduction

Trigonometry is important in pre-calculus studies.

### Conclusion

This chapter surveys important topics in trigonometry relevant for the study of calculus. Please fork the LaTeX source code for PCiB (available on GitHub) and create your own book that chooses the facts and exercises most relevant to you! Also, starring the PCiB project on GitHub would be greatly appreciated! Thanks for reading PCiB!



# Appendix I

## Basic GitHub Guide

### A Quick Start to Your GitHub Journey

Welcome to the fascinating world of GitHub, a platform that has revolutionized the way we collaborate on projects, share code, and build software together. Whether you are a programmer, a writer, or a historian, GitHub provides a set of powerful tools to help you collaborate with others, manage your projects, and contribute to the vast world of open-source software. In this guide, we will walk you through the foundational steps to get started with GitHub, helping you to navigate, contribute, and make the most out of this incredible platform.

### Creating Your GitHub Account

The first step to joining the GitHub community is to create an account. Here's how you can do it:

1. Visit the GitHub website.
2. Click on the “Sign up” button.
3. Fill in the required information, including your username, email address, and password.
4. Verify your account and complete the sign-up process.

Once you have created your account, take a moment to explore your new GitHub dashboard. Here, you will find a variety of tools and features that will help you manage your projects, collaborate with others, and discover new and interesting repositories.

## Creating Your First Repository

A repository (or “repo”) is a digital directory where you can store your project files. Here’s how you can create your first repository:

1. From your GitHub dashboard, click on the “New” button to create a new repository.
2. Give your repository a name and provide a brief description.
3. Initialize this repository with a README file. (This is an optional step, but it’s a good practice to include a README file in every repository to explain what your project is about.)
4. Click “Create repository.”

Congratulations! You have just created your first GitHub repository. You can now start adding files, collaborating with others, and managing your project right from GitHub.

## Making Changes and Commits

GitHub uses Git, a version control system, to keep track of changes made to your project. Here’s a quick guide on how to make changes and commits:

1. Navigate to your repository on GitHub.
2. Find the file you want to edit, and click on it.
3. Click the pencil icon to start editing.
4. Make your changes and then scroll down to the “Commit changes” section.
5. Provide a commit message that explains the changes you made.
6. Choose whether you want to commit directly to the main branch or create a new branch for your changes.
7. Click “Commit changes.”

Your changes are now saved, and a new commit is created. Every commit has a unique ID, making it easy to track changes, revert to previous versions, and collaborate with others.



## Collaborating with Others

One of the biggest strengths of GitHub is its collaborative nature. Here are some ways you can collaborate with others:

- **Forking:** You can fork a repository, create your own copy, make changes, and then propose those changes back to the original project.
- **Issues:** Use issues to report bugs, request new features, or start a discussion with the community.
- **Pull Requests:** Propose changes to a project by creating a pull request. This allows others to review your changes, discuss them, and eventually merge them into the project.

## Conclusion: Embarking on Your GitHub Adventure

Now that you have a basic understanding of GitHub and how it works, you are ready to embark on your GitHub adventure. Explore repositories, contribute to open-source projects, collaborate with others, and build amazing things together. Remember, the GitHub community is vast and supportive, and there is a wealth of knowledge and resources available to help you along the way. Happy coding!



# Appendix II

## Basic L<sup>A</sup>T<sub>E</sub>X Guide

### A Quick Start to Your L<sup>A</sup>T<sub>E</sub>X Journey

Welcome to the immersive world of L<sup>A</sup>T<sub>E</sub>X, a typesetting system widely used for creating scientific and professional documents due to its powerful handling of formulas and bibliographies. This guide is designed to offer you the foundational steps to grasp the basics of L<sup>A</sup>T<sub>E</sub>X, enabling you to craft documents of high typographic quality akin to this book.

### Setting Up Your L<sup>A</sup>T<sub>E</sub>X Environment

Before you can start creating documents with L<sup>A</sup>T<sub>E</sub>X, you need to set up a working L<sup>A</sup>T<sub>E</sub>X environment on your computer. Here's how you can do it:

1. Download and install a T<sub>E</sub>X distribution, which includes L<sup>A</sup>T<sub>E</sub>X. For Windows, MiKTeX is a popular choice, while Mac users might prefer MacTeX, and TeX Live is widely used on Linux.
2. Install a L<sup>A</sup>T<sub>E</sub>X editor. Some popular options include TeXShop (for Mac), TeXworks (cross-platform), and Overleaf (an online L<sup>A</sup>T<sub>E</sub>X editor).
3. Ensure that your T<sub>E</sub>X distribution and L<sup>A</sup>T<sub>E</sub>X editor are properly configured and integrated.

### Creating Your First L<sup>A</sup>T<sub>E</sub>X Document

Once your L<sup>A</sup>T<sub>E</sub>X environment is set up, you are ready to create your first L<sup>A</sup>T<sub>E</sub>X document. Follow these steps:

1. Open your L<sup>A</sup>T<sub>E</sub>X editor and create a new document.
2. Insert the following code to set up a basic L<sup>A</sup>T<sub>E</sub>X document:

```
\documentclass{article}
\begin{document}
Hello, \LaTeX\ world!
\end{document}
```

3. Save your document with a .tex file extension.
4. Compile your document using your L<sup>A</sup>T<sub>E</sub>X editor. This process converts your .tex file into a PDF document.
5. View the output PDF and admire your first L<sup>A</sup>T<sub>E</sub>X creation.

## Understanding L<sup>A</sup>T<sub>E</sub>X Commands and Environments

L<sup>A</sup>T<sub>E</sub>X documents are created using a series of commands and environments. Commands typically start with a backslash `\` and are used to format text, insert special characters, or execute functions. Environments are used to define specific sections of your document that require special formatting.

- **Commands:** For example, `\{italics}` will render the word "italics" in italic font.
- **Environments:** To create a bulleted list, you would use the *itemize* environment:

```
\begin{itemize}
  \item First item
  \item Second item
\end{itemize}
```

## Adding Structure to Your Document

L<sup>A</sup>T<sub>E</sub>X makes it easy to structure your documents with sections, subsections, and chapters. Here's how you can add structure:

```
\section{Introduction}
This is the introduction of your document.
\subsection{Background}
This subsection provides background information.
\subsubsection{Details}
This is a subsubsection for more detailed information.
```

## Including Mathematical Formulas

L<sup>A</sup>T<sub>E</sub>X excels at typesetting mathematical formulas. Use the *equation* environment or the \$ sign for inline formulas. For example:

The quadratic formula is  $\left( x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \right)$ .

## Adding Images and Tables

You can also include images and tables in your L<sup>A</sup>T<sub>E</sub>X documents:

- **Images:** Use the *graphicx* package and the *includegraphics* command.
- **Tables:** Use the *tabular* environment to create tables.

## Compiling Your Document

L<sup>A</sup>T<sub>E</sub>X documents need to be compiled to produce a PDF. This can be done through your L<sup>A</sup>T<sub>E</sub>X editor. If your document includes bibliographies or cross-references, you may need to compile multiple times.

## Conclusion: Embracing the Power of L<sup>A</sup>T<sub>E</sub>X

Congratulations! You have taken your first steps into the world of L<sup>A</sup>T<sub>E</sub>X. With practice, you will discover that L<sup>A</sup>T<sub>E</sub>X is a powerful tool for creating professional-quality documents, from simple articles to complex books. Embrace the learning curve, explore the vast array of packages available, and join the community of L<sup>A</sup>T<sub>E</sub>X users who are ready to help you on your journey. Happy typesetting!



# Bibliography

- [MT77] F. Mosteller and J. W. Tukey. *Data Analysis and Regression: A Second Course in Statistics*. Addison-Wesley Pub Co, Reading, MA, 1977.