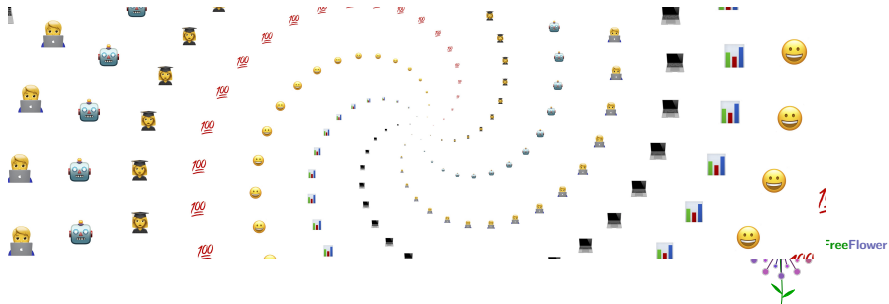


# Textverarbeitung in $\text{\LaTeX}$

$\text{\LaTeX}$

Thomas Graf

Fachschaft Informatik  
Kantonsschule im Lee



# Was ist L<sup>A</sup>T<sub>E</sub>X?

- ▶ Textverarbeitungssoftware

L<sup>A</sup>T<sub>E</sub>X



# Was ist L<sup>A</sup>T<sub>E</sub>X?

L<sup>A</sup>T<sub>E</sub>X

- ▶ Textverarbeitungssoftware
- ▶ sehr gut geeignet zur Verfassung von:



# Was ist L<sup>A</sup>T<sub>E</sub>X?

L<sup>A</sup>T<sub>E</sub>X

- ▶ Textverarbeitungssoftware
- ▶ sehr gut geeignet zur Verfassung von:
  - ▶ Maturitätsarbeiten



# Was ist L<sup>A</sup>T<sub>E</sub>X?

L<sup>A</sup>T<sub>E</sub>X

- ▶ Textverarbeitungssoftware
- ▶ sehr gut geeignet zur Verfassung von:
  - ▶ Maturitätsarbeiten
  - ▶ Bachelorarbeiten / Masterarbeiten



# Was ist L<sup>A</sup>T<sub>E</sub>X?

L<sup>A</sup>T<sub>E</sub>X

- ▶ Textverarbeitungssoftware
- ▶ sehr gut geeignet zur Verfassung von:
  - ▶ Maturitätsarbeiten
  - ▶ Bachelorarbeiten / Masterarbeiten
  - ▶ Beamer Präsentationen (Slides)



# Was ist L<sup>A</sup>T<sub>E</sub>X?

L<sup>A</sup>T<sub>E</sub>X

- ▶ Textverarbeitungssoftware
- ▶ sehr gut geeignet zur Verfassung von:
  - ▶ Maturitätsarbeiten
  - ▶ Bachelorarbeiten / Masterarbeiten
  - ▶ Beamer Präsentationen (Slides)
  - ▶ wissenschaftliche Publikationen



# Was ist L<sup>A</sup>T<sub>E</sub>X?

L<sup>A</sup>T<sub>E</sub>X

- ▶ Textverarbeitungssoftware
- ▶ sehr gut geeignet zur Verfassung von:
  - ▶ Maturitätsarbeiten
  - ▶ Bachelorarbeiten / Masterarbeiten
  - ▶ Beamer Präsentationen (Slides)
  - ▶ wissenschaftliche Publikationen
  - ▶ Büchern



## Korrekte Aussprache von L<sup>A</sup>T<sub>E</sub>X

X x

Das X im Wort L<sup>A</sup>T<sub>E</sub>X ist kein grosses X wie im Wort *Xylophon*, sondern der gross geschriebene griechische Buchstabe Chi.

Die Aussprache von Chi ist ein hartes *ch*, so wie im deutschen Wort *ach*.

Deshalb ist die Aussprache von L<sup>A</sup>T<sub>E</sub>X nicht Latex, sondern **Latech**. (Latex ist der Milchsaft verschiedener Kautschukpflanzen und hat mit Textverarbeitung wenig zu tun.)



# Unterschiede zwischen MS Word und L<sup>A</sup>T<sub>E</sub>X

Man unterscheidet (grob) zwischen zwei Kategorien von Textverarbeitungssoftware:

- ▶ WYSIWYG



# Unterschiede zwischen MS Word und L<sup>A</sup>T<sub>E</sub>X

Man unterscheidet (grob) zwischen zwei Kategorien von Textverarbeitungssoftware:

- ▶ WYSIWYG
- ▶ WYSIWYM



## **WYSIWYG** („**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et“)

- ▶ Bei (echtem) WYSIWYG wird ein Dokument während der Bearbeitung am Bildschirm genauso angezeigt, wie es bei der Ausgabe über ein anderes Gerät, z.B. einen Drucker, aussieht. (→ Echtzeitdarstellung)



## **WYSIWYG** („**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et“)

- ▶ Bei (echtem) WYSIWYG wird ein Dokument während der Bearbeitung am Bildschirm genauso angezeigt, wie es bei der Ausgabe über ein anderes Gerät, z.B. einen Drucker, aussieht. (→ Echtzeitdarstellung)
- ▶ bekannte Beispiele: MS Word, MS PowerPoint, Gmail, WhatsApp



# WYSIWYM

## **WYSIWYM** („**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **M**ean“)

- ▶ Am Bildschirm ist während der Eingabe für den Benutzer nur in etwa erkennbar, was er mit seiner Formatierung bezwecken will.



## **WYSIWYM** („**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **M**ean“)

- ▶ Am Bildschirm ist während der Eingabe für den Benutzer nur in etwa erkennbar, was er mit seiner Formatierung bezwecken will.
- ▶ Die endgültige Formatierung wird erst beim Umwandeln des Dokuments in ein Endformat (z.B. PDF) durchgeführt.



## WYSIWYM („What You See Is What You Mean“)

- ▶ Am Bildschirm ist während der Eingabe für den Benutzer nur in etwa erkennbar, was er mit seiner Formatierung bezwecken will.
- ▶ Die endgültige Formatierung wird erst beim Umwandeln des Dokuments in ein Endformat (z.B. PDF) durchgeführt.
- ▶ bekannte Beispiele:  $\text{\LaTeX}$ , (X)HTML, CSS



kleiner Joke ...



L<sup>A</sup>T<sub>E</sub>X



# Lokale Installation (auf eigenem Rechner)

Benötigt werden typischerweise

- ▶ eine T<sub>E</sub>X-Distribution (TeX Live, MacTex, MiKTeX, ...)



# Lokale Installation (auf eigenem Rechner)

Benötigt werden typischerweise

- ▶ eine T<sub>E</sub>X-Distribution (TeX Live, MacTex, MiKTeX, ...)
- ▶ ein geeigneter T<sub>E</sub>X-Editor (z.B. VS Code, TeXstudio, TEXMAKER)



# Installation einer T<sub>E</sub>X-Distribution

▶ MacOS:

```
brew install --cask mactex
```



# Installation einer T<sub>E</sub>X-Distribution

▶ MacOS:

```
brew install --cask mactex
```

▶ Windows:

```
winget install --id MiKTeX.MiKTeX
```

```
winget install --id StrawberryPerl.StrawberryPerl
```



# Installation einer T<sub>E</sub>X-Distribution

- ▶ MacOS:

```
brew install --cask mactex
```

- ▶ Windows:

```
winget install --id MiKTeX.MiKTeX
```

```
winget install --id StrawberryPerl.StrawberryPerl
```

- ▶ Starte deinen Computer neu.



# Installation und Konfiguration einer $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Extension für VS Code

Installiere die VS Code-Extension Latex Workshop (von James Yu).



# Latex Workshop Rezepte erstellen

Öffne in VS Code die Kommando-Palette durch:

- ▶ MacOS:

`Cmd + Shift + P`

und tippe: `Preferences: Open User Settings (JSON)`. Dies sollte das File `settings.json` öffnen.

**Füge** den Code in `latex_recipies.json` (siehe Moodle) zu deinem `settings.json` hinzu.



# Latex Workshop Rezepte erstellen

Öffne in VS Code die Kommando-Palette durch:

- ▶ MacOS:

`Cmd + Shift + P`

- ▶ Windows:

`Ctrl + Shift + P`

und tippe: Preferences: Open User Settings (JSON). Dies sollte das File `settings.json` öffnen.

**Füge** den Code in `latex_recipies.json` (siehe Moodle) zu deinem `settings.json` hinzu.



# erstes Dokument

Erstellen Sie in VS Code ein neues File `minimal.tex` mit dem folgenden Inhalt:

```
\documentclass{article}
```

```
\begin{document}
```

```
Hello, World!
```

```
\end{document}
```

Programm: `minimal.tex`

und kompiliere dieses mit der Extension Latex Workshop.



# Befehle (commands) in L<sup>A</sup>T<sub>E</sub>X

```
\NameDesBefehls % Befehl ohne Argumente und ohne Optionen  
\NameDesBefehls{} % mit einem Argument  
\NameDesBefehls[]{} % mit Optionen und einem Argument  
\NameDesBefehls[]{}{} % mit Optionen und 2 Argumenten
```



# Beispiele von Befehlen in L<sup>A</sup>T<sub>E</sub>X

- ▶ `\textbf{Jane Doe}` (setzt *Jane Doe* in Fettschrift)



# Beispiele von Befehlen in L<sup>A</sup>T<sub>E</sub>X

- ▶ `\textbf{Jane Doe}` (setzt *Jane Doe* in Fettschrift)
- ▶ `{\huge Hallo}` (macht *Hallo* sehr gross)



# Beispiele von Befehlen in L<sup>A</sup>T<sub>E</sub>X

- ▶ `\textbf{Jane Doe}` (setzt *Jane Doe* in Fettschrift)
- ▶ `{\huge Hallo}` (macht *Hallo* sehr gross)
- ▶ `\today` (setzt das aktuelle Datum)



# Beispiele von Befehlen in L<sup>A</sup>T<sub>E</sub>X

- ▶ `\textbf{Jane Doe}` (setzt *Jane Doe* in Fettschrift)
- ▶ `{\huge Hallo}` (macht *Hallo* sehr gross)
- ▶ `\today` (setzt das aktuelle Datum)
- ▶ `\section{neuer Abschnitt}` (erstellt einen neuen Abschnitt)



# Beispiele von Befehlen in L<sup>A</sup>T<sub>E</sub>X

- ▶ `\textbf{Jane Doe}` (setzt *Jane Doe* in Fettschrift)
- ▶ `{\huge Hallo}` (macht *Hallo* sehr gross)
- ▶ `\today` (setzt das aktuelle Datum)
- ▶ `\section{neuer Abschnitt}` (erstellt einen neuen Abschnitt)
- ▶ `\documentclass[a4paper,12pt]{article}` (setzt die Dokumentklasse auf *article*, das Format auf A4 und die Textgrösse auf 12pt)



# Beispiele von Befehlen in L<sup>A</sup>T<sub>E</sub>X

- ▶ `\textbf{Jane Doe}` (setzt *Jane Doe* in Fettschrift)
- ▶ `{\huge Hallo}` (macht *Hallo* sehr gross)
- ▶ `\today` (setzt das aktuelle Datum)
- ▶ `\section{neuer Abschnitt}` (erstellt einen neuen Abschnitt)
- ▶ `\documentclass[a4paper,12pt]{article}` (setzt die Dokumentklasse auf *article*, das Format auf A4 und die Textgrösse auf 12pt)
- ▶ `\textcolor{Red}{roter Text}` (färbt *roter Text* rot)



## Aufgabe 0.1

Erstelle ein PDF-Dokument mit exakt folgendem Inhalt (recherchiere):

**What can** be asserted without evidence can also be *dismissed* without evidence.



# Pakete

`\usepackage{NameDesPakets}`

`\usepackage[Zusatzoptionen]{NameDesPakets}`

Alle `\usepackage{...}` Befehle müssen **nach**

`\documentclass{...}` und **vor** `\begin{document}` stehen!

wichtige Informationsquelle: <https://ctan.org/>



# Umgebungen (environments) in L<sup>A</sup>T<sub>E</sub>X

- 1 `\begin{NameDerUmgebung}`
- 2 Hier soll etwas Interessantes geschehen.
- 3 `\end{NameDerUmgebung}`

Beispiele:

```
\begin{itemize}
  \item erster Bulletpunkt
  \item zweiter Bulletpunkt
\end{itemize}
```

% Ausnahme: Formel-Environment mit Dollar-Zeichen \$\$  
(oder ähnliche)

```
$$\alpha + 4 = \gamma - 2$
```



## Aufgabe 0.2

Recherchiere im Internet, um folgendes in  $\text{\LaTeX}$  sinngemäss zu reproduzieren:

- ▶ Pythagoras:  $a^2 + b^2 = c^2$
- ▶  $\Rightarrow c = \sqrt{a^2 + b^2}$

Symbole und Formeln müssen zwischen Dollar-Zeichen geschrieben werden: \$ \dots \$.



# Beispiel: Bilder in L<sup>A</sup>T<sub>E</sub>X einfügen

```
1 \begin{figure}  
2   \centering  
3   \includegraphics[width=0.7\textwidth]{baikal.jpg}  
4   \caption{Baikalsee}  
5 \end{figure}  
benötigt \usepackage{graphicx}
```

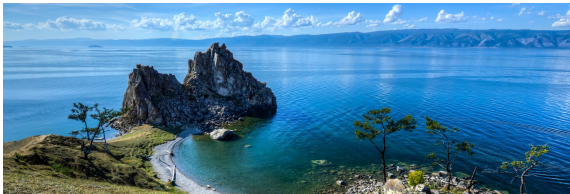


Abbildung: Baikalsee



### Aufgabe 0.3

Speichere ein Bild deiner Wahl in demselben Ordner wie dein  $\text{\LaTeX}$ -Dokument ab. Skaliere das Bild angemessen und gib ihm eine kurze Beschreibung in blauer Farbe.



Abbildung: Salzwasserkrokodil an Strand in Darwin, Australien



## Aufgabe 0.4

Informiere Dich über das `align` Environment und mathematische Formeln in  $\text{\LaTeX}$  im Internet und versuche folgende Ausdrücke zu reproduzieren:

$$e^{i\pi} + 1 = 0 \tag{1}$$

$$\sqrt[4]{8} \tag{2}$$

$$(x + y)^n = \sum_{r=0}^n \binom{n}{r} x^r y^{n-r} \tag{3}$$



## Aufgabe 0.5

### Griechisch

Informationen über das griechische Alphabet:

1. Der erste Buchstabe des griechischen Alphabets ist  $\alpha$ .
2. Das Alphabet umfasst 24 Buchstaben.

```
\documentclass{article}
\begin{document}
\usepackage{xcolor}
\indent
\textcolor{red}{Griechisch}
Informationen über das griechische Alphabet:
\begin{itemize}
  \item Der erste Buchstabe des griechischen Alphabets ist
  \alpha.
  \item Das Alphabet umfasst \underline{24} Buchstaben.
\end{itemize}
```

Finde mindestens vier Fehler im obigen L<sup>A</sup>T<sub>E</sub>X-Dokument.



## Aufgabe 0.6

1. Erstelle im Table-Generator <https://www.tablesgenerator.com/> eine Tabelle mit 3 Zeilen und 4 Spalten und fülle die Einträge mit beliebigen Zahlen. Füge die dabei entstandene Tabelle in ein  $\text{\LaTeX}$ -Dokument ein und betrachte das Resultat.
2. Versuche den AI-Coding Assistenten eine gleichartige Tabelle erstellen zu lassen. Welches Vorgehen scheint dir einfacher?



# Historische Entwicklung

T<sub>E</sub>X ist ein im Jahre 1978 veröffentlichtes Computerprogramm zur Erstellung von digitalen Texten, welches hauptsächlich von dem berühmten US-amerikanischen Informatiker Donald E. Knuth geschrieben wurde. Sein Hauptziel war es, mit T<sub>E</sub>X ein Werkzeug zu schaffen, mit dessen Hilfe hochqualitative Bücher mit wenig Aufwand verfasst werden können. Des weiteren sollte T<sub>E</sub>X auf jedem Computer und zu jeder Zeit identische Resultate (Texte) produzieren <sup>1</sup>.



---

<sup>1</sup>Details zur Geschichte von T<sub>E</sub>X findet man hier: <https://www.tug.org/whatis.html>

# Entwicklung von T<sub>E</sub>X



Abbildung: Donald Ervin Knuth



# Historische Entwicklung

T<sub>E</sub>X ist insbesondere eine Programmiersprache. Diese ermöglicht die Definition diverser Befehle, welche den Computer z.B. beauftragen, die Schriftgrösse anzupassen, Abstände im Text einzubauen oder die Schriftart zu ändern. Eine T<sub>E</sub>X-Distribution enthält viele solcher bereits definierten primitive Befehle. Ein Beispiel eines solchen Befehls ist `\baselineskip = 24pt`, welcher bewirkt, dass zwischen den einzelnen Zeilen im Text ein Abstand von 24 Einheitspunkten gesetzt wird (in T<sub>E</sub>X beginnen Befehle mit einem Backslash `\`).



# Historische Entwicklung

Nehmen wir an, wir möchten nun eine Präsentation wie diese hier schreiben. Konzentrieren wir uns einmal auf die Überschrift „Historische Entwicklung“ dieses Abschnitts (Section). Wie ist diese wohl entstanden? Um eine solche Überschrift in T<sub>E</sub>X zu erstellen, müssten dem Computer diverse Instruktionen (Befehle) übergeben werden:

- ▶ Befehle zur Anpassung der Schriftgrösse und Dicke der Schrift
- ▶ diverse Befehle zum Einfügen bestimmter Abstände vor und nach der Überschrift
- ▶ die blaue Färbung
- ▶ usw.



# Historische Entwicklung

Dies bedeutet, dass wir zahlreiche  $\text{T}_{\text{E}}\text{X}$ -Befehle geschickt kombinieren müssten, um einen neuen Abschnitt deklarieren zu können. Dies erscheint recht unpraktisch zu sein. Was wäre jedoch, wenn jemand sich die Mühe machen würde, so einen komplexen Befehl für das Erstellen eines neuen Textabschnittes aus primitiven  $\text{T}_{\text{E}}\text{X}$ -Befehlen zusammensetzen? Diese tatkräftige Person könnte der Kombination aller dafür benötigten primitiven Befehle einen neuen Namen, wie z.B. `\section{}` geben. Dieser Befehl steht dann als Stellvertreter für die Kombination aller Befehle, welche zur Erstellung eines neuen Abschnitts notwendig sind. Man bezeichnet den dadurch entstandenen neuen Befehl als *Makro* (griechisch  $\mu\alpha\kappa\rho\acute{o}\sigma$  makros = 'gross', 'weit', 'lang').



# Historische Entwicklung

Die Idee hinter Makros ist simpel und effektiv: man fasst alle kleinen (mikro) Befehle, welche benötigt werden, um z.B. einen neuen Abschnitt (Section) zu erstellen, zu einem neuen, grossen (makro) Befehl zusammen und gibt diesem einen Namen, wie beispielsweise `\section{}`. Dass dieses Vorgehen möglich ist, dürfte nicht erstaunen, da  $\text{T}\text{E}\text{X}$  wie bereits erwähnt, eine vollwertige Programmiersprache ist, mit welcher natürlich solche neuen Befehle definiert werden können.



# Historische Entwicklung

Der beflissene US-amerikanische Informatiker Leslie B. Lamport baute am Anfang der 1980er Jahre zahlreiche höchst nützliche Makro-Befehle aus primitiven  $\text{T}_{\text{E}}\text{X}$ -Befehlen zusammen. Das daraus resultierende Softwarepaket, welches die Benutzung von  $\text{T}_{\text{E}}\text{X}$  mit Hilfe von Makros stark vereinfacht, nennt sich  $\text{\LaTeX}$  (kurz für **L**amport **T**e**X**).  $\text{\LaTeX}$  ist also nichts anderes als eine auf  $\text{T}_{\text{E}}\text{X}$  aufbauende (und in  $\text{T}_{\text{E}}\text{X}$  geschriebene) Erweiterung von  $\text{T}_{\text{E}}\text{X}$ . Mittlerweile gibt es tausende von  $\text{T}_{\text{E}}\text{X}$ -Paketen, welche von diversen Leuten verfasst wurden und eine ungeheure Vielfalt an nützlicher Funktionalität zur  $\text{T}_{\text{E}}\text{X}$  /  $\text{\LaTeX}$  Basisdistribution hinzufügen <sup>2</sup>.  $\text{\LaTeX}$  ist eine sogenannte *freie Software* und unterliegt den Regelungen der LaTeX Project Public License (LPPL) <sup>3</sup>.



---

<sup>2</sup>umfangreiche Sammlung von  $\text{T}_{\text{E}}\text{X}$ -Paketen: <https://ctan.org/?lang=en>

<sup>3</sup>Details zur LPPL-Lizenz: <https://www.latex-project.org/lppl/>

# Entwicklung von L<sup>A</sup>T<sub>E</sub>X

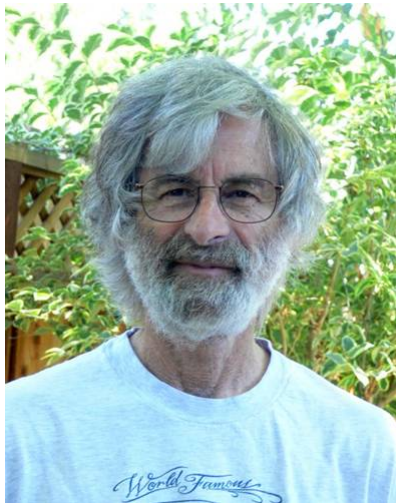


Abbildung: Leslie Lamport



## Macros in L<sup>A</sup>T<sub>E</sub>X (ohne Argumente)

In einem Laborbericht erwähnen wir häufig den vollen Namen von TNT, nämlich **2-Methyl-1,3,5-trinitrobenzen** (in Fettschrift). Dazu müsste man an den entsprechenden Stellen jeweils den Command `\textbf{2-Methyl-1,3,5-trinitrobenzen}` hinschreiben. Wenn man dies oft tun muss, ist es einfach, diesen Command durch ein Makro zu ersetzen, welchem wir den Namen `\TNT` geben möchten:

```
\newcommand{\TNT}{\textbf{2-Methyl-1,3,5-trinitrobenzen}}
```



# Lernziele

- ▶ Begriffe *WYSIWYG* und *WYSIWYM* erklären können
- ▶ den Aufbau und die Struktur eines  $\LaTeX$ -Dokuments kennen
- ▶ Aufbau und Struktur von Befehlen in  $\LaTeX$  verstehen und einige grundlegende Befehle wie `\textit`, `\textbf`, `\sqrt` kennen
- ▶ die Idee hinter Macros und ihren Aufbau kennen
- ▶ typische Fehler in gegebenen Code-Beispielen erkennen und erklären können

