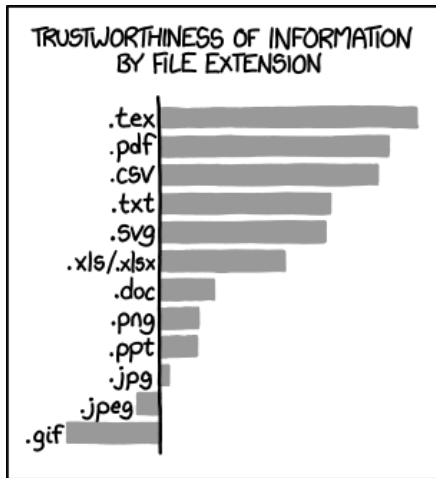


EINFÜHRUNG L^AT_EX

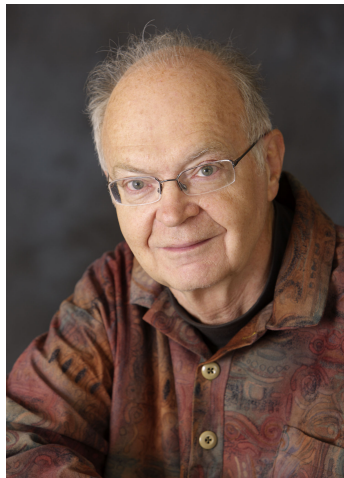


xkcd.com/1301

Was ist L^AT_EX?

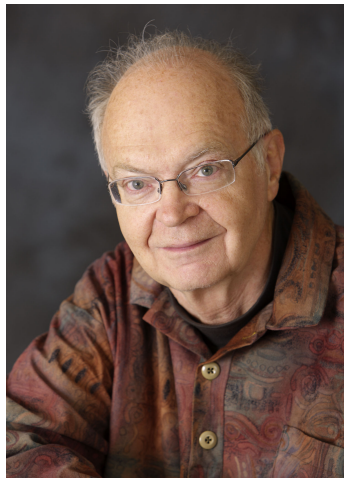
Grundlagen – Ein einfaches Dokument erstellen
Tolle Sachen – Was noch alles möglich ist
Hilfreiche Pakete

Entwicklung
Vergleich
L^AT_EX



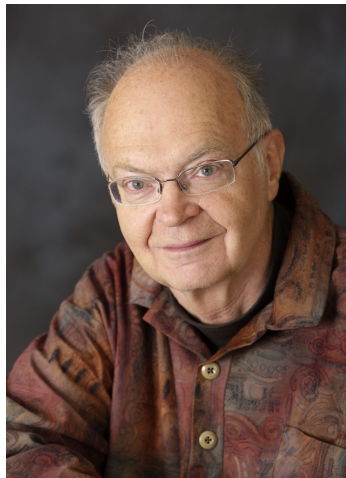
Donald E. Knuth

- ▶ Schreibt seit 1969 an seiner Buchreihe *The Art of Computer Programming*



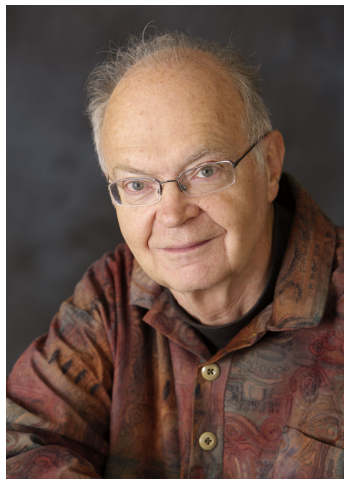
Donald E. Knuth

- ▶ Schreibt seit 1969 an seiner Buchreihe *The Art of Computer Programming*
- ▶ War unzufrieden mit den Möglichkeiten zur Schriftsetzung, und begann 1977 mit der Entwicklung seines eigenen Systems: T_EX

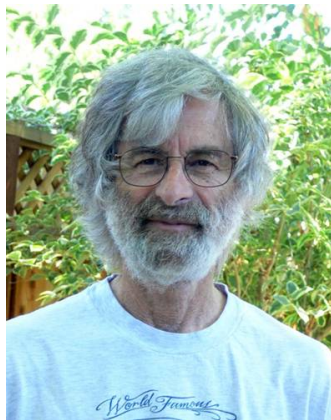


Donald E. Knuth

- ▶ Schreibt seit 1969 an seiner Buchreihe *The Art of Computer Programming*
- ▶ War unzufrieden mit den Möglichkeiten zur Schriftsetzung, und begann 1977 mit der Entwicklung seines eigenen Systems: T_EX
- ▶ T_EX automatisiert den Satz (u. A. Wortabstand und Absätze)

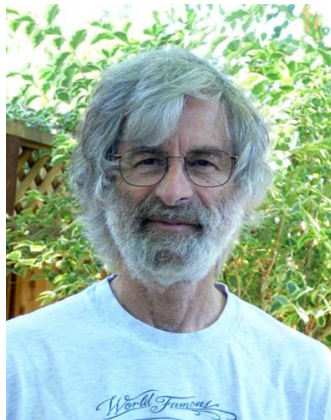


Donald E. Knuth



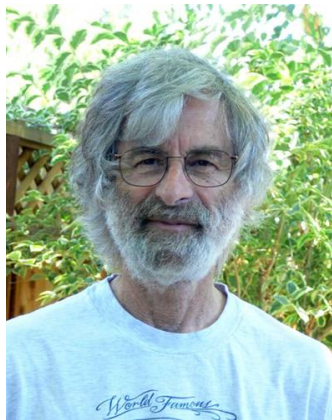
Leslie Lamport

- ▶ L^AT_EX wurde Anfang der 80er von Leslie Lamport entwickelt



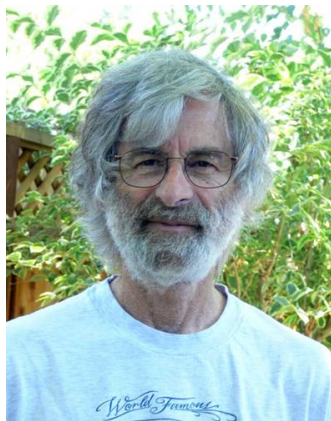
Leslie Lamport

- ▶ L^AT_EX wurde Anfang der 80er von Leslie Lamport entwickelt
- ▶ Vereinfacht die Benutzung von T_EX durch Makros



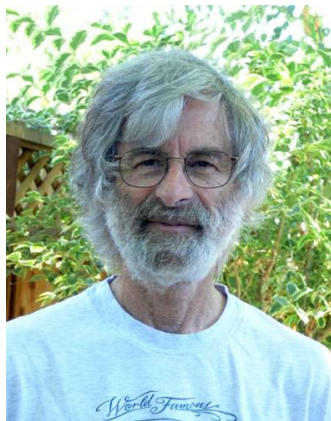
Leslie Lamport

- ▶ L^AT_EX wurde Anfang der 80er von Leslie Lamport entwickelt
- ▶ Vereinfacht die Benutzung von T_EX durch Makros
- ▶ Erweitert T_EX u. A. um Layoutgestaltung



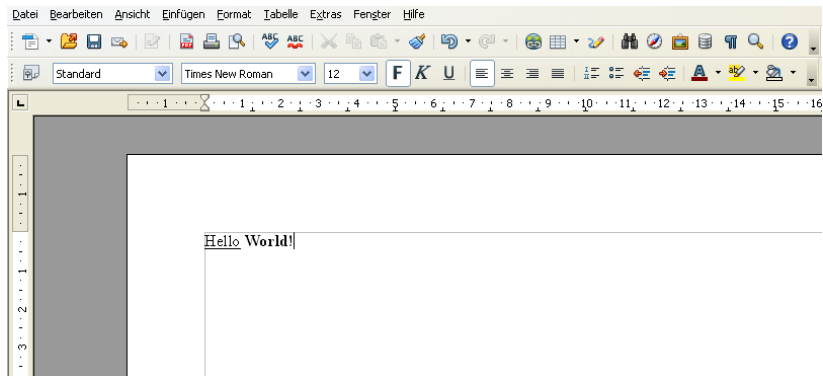
Leslie Lamport

- ▶ L^AT_EX wurde Anfang der 80er von Leslie Lamport entwickelt
- ▶ Vereinfacht die Benutzung von T_EX durch Makros
- ▶ Erweitert T_EX u. A. um Layoutgestaltung
- ▶ Im Gegensatz zu T_EX wird L^AT_EX noch weiterentwickelt



Leslie Lamport

Beispiel: WYSIWYG-Schreibprogramm



Beispiel: HTML

Hello_World.html:

```
<html>

<head>
<title>Hello World</title>
</head>

<body>
<u>Hello</u> <b>World!</b>
</body>

</html>
```

Beispiel: L^AT_EX

Hello_World.tex:

```
\documentclass{article}
\title{Hello World}

\begin{document}
\underline{Hello} \textbf{World!}
\end{document}
```

- ▶ .tex-Datei besteht aus:
 - ▶ Preamble mit dokumentweit gültigen Definitionen, Formatbeschreibungen etc.
 - ▶ Dokumentkörper mit eigentlichem Inhalt

- ▶ .tex-Datei besteht aus:
 - ▶ Preamble mit dokumentweit gültigen Definitionen, Formatbeschreibungen etc.
 - ▶ Dokumentkörper mit eigentlichem Inhalt
- ▶ Dokumentstrukturierung durch Umgebungen und Gliederungsbefehle

- ▶ .tex-Datei besteht aus:
 - ▶ Preamble mit dokumentweit gültigen Definitionen, Formatbeschreibungen etc.
 - ▶ Dokumentkörper mit eigentlichem Inhalt
- ▶ Dokumentstrukturierung durch Umgebungen und Gliederungsbefehle
- ▶ Formatierung, Sonderzeichen, ... (praktisch alles außer einfachem Text) durch Befehle

- ▶ .tex-Datei besteht aus:
 - ▶ Preamble mit dokumentweit gültigen Definitionen, Formatbeschreibungen etc.
 - ▶ Dokumentkörper mit eigentlichem Inhalt
- ▶ Dokumentstrukturierung durch Umgebungen und Gliederungsbefehle
- ▶ Formatierung, Sonderzeichen, ... (praktisch alles außer einfachem Text) durch Befehle

Praktisch alles kann bei Bedarf automatisiert und programmiert werden.

Grundlagen

- ▶ Welche Programme sind für die Arbeit mit L^AT_EX notwendig?
- ▶ Wie sieht ein einfaches Dokument aus?

Was wird benötigt?

- ▶ Notwendig: eine TeX-Distribution (“Compiler”)
(z. B. *MikTeX* für Win, für Linux package *latex*)
- ▶ Empfohlen: ein Editor (z. B. *TeXMaker* (Win/Linux))
 - ▶ Bietet Syntax-highlighting, Autovervollständigung, unkompliziertes kompilieren, ...

Preamble bzw. Header

- ▶ Angabe der Dokumentklasse (zwingend!):

```
\documentclass{Dokumentklasse}
```

Dokumentklassen sind z. B. article, book, report,...

→ beeinflussen einige globale Formatierungseinstellungen

Preamble bzw. Header

- ▶ Angabe der Dokumentklasse (zwingend!):

```
\documentclass{Dokumentklasse}
```

Dokumentklassen sind z. B. article, book, report,...

→ beeinflussen einige globale Formatierungseinstellungen

- ▶ Angabe der benötigten Zusatzpakete:

```
\usepackage[optionale Parameter]{notwendige Parameter}
```

Preamble bzw. Header

- ▶ Angabe der Dokumentklasse (zwingend!):

```
\documentclass{Dokumentklasse}
```

Dokumentklassen sind z. B. article, book, report,...

→ beeinflussen einige globale Formatierungseinstellungen

- ▶ Angabe der benötigten Zusatzpakete:

```
\usepackage[optionale Parameter]{notwendige Parameter}
```

- ▶ Dokumentweit gültige Angaben (author, title, datum...)

```
\author{Horst}
```

```
\title{Wetterbericht}
```

```
\date{1.1.1970}
```

Beispiel: Preamble

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

```
\usepackage[utf8]{inputenc}
```

```
\usepackage[ngerman]{babel}
```

```
\author{Ango}
```

```
\title{Ein Text}
```

Beispiel: Preamble - Erläuterungen

- ▶ `\documentclass[a4paper]{article}` wählt die Dokumentklasse *article* (für kurze Artikel) und gibt als zusätzlichen Parameter die Seitengröße DIN A4 an.

Beispiel: Preamble - Erläuterungen

- ▶ `\documentclass[a4paper]{article}` wählt die Dokumentklasse *article* (für kurze Artikel) und gibt als zusätzlichen Parameter die Seitengröße DIN A4 an.
- ▶ `\usepackage[utf8]{inputenc}` gibt an, dass ich das Zusatzpaket *inputenc* benutzen möchte, das z. B. ermöglicht, auch diverse Umlaute direkt in der tex-Datei zu schreiben

Beispiel: Preamble - Erläuterungen

- ▶ `\documentclass[a4paper]{article}` wählt die Dokumentklasse *article* (für kurze Artikel) und gibt als zusätzlichen Parameter die Seitengröße DIN A4 an.
- ▶ `\usepackage[utf8]{inputenc}` gibt an, dass ich das Zusatzpaket *inputenc* benutzen möchte, das z. B. ermöglicht, auch diverse Umlaute direkt in der tex-Datei zu schreiben
- ▶ `\usepackage[ngerman]{babel}`: das Paket *babel* ändert die Dokumentsprache (durch *ngerman* auf Deutsch), das beeinflusst z. B. Datumsausgaben oder die Überschrift des Inhaltsverzeichnisses.

Beispiel: Preamble - Erläuterungen

- ▶ `\documentclass[a4paper]{article}` wählt die Dokumentklasse *article* (für kurze Artikel) und gibt als zusätzlichen Parameter die Seitengröße DIN A4 an.
- ▶ `\usepackage[utf8]{inputenc}` gibt an, dass ich das Zusatzpaket *inputenc* benutzen möchte, das z. B. ermöglicht, auch diverse Umlaute direkt in der tex-Datei zu schreiben
- ▶ `\usepackage[ngerman]{babel}`: das Paket *babel* ändert die Dokumentsprache (durch *ngerman* auf Deutsch), das beeinflusst z. B. Datumsausgaben oder die Überschrift des Inhaltsverzeichnisses.
- ▶ `\author{Ango}` und `\title{Ein Text}` definieren Autor und Titel des Dokuments.

Body

Der Body wird durch die `document`-Umgebung definiert.

- ▶ Text erscheint automatisch im Blocksatz.

Body

Der Body wird durch die `document`-Umgebung definiert.

- ▶ Text erscheint automatisch im Blocksatz.
- ▶ Jede beliebige Anzahl an Leerzeichen und einzelner Zeilenumbrüche werden als einziges Leerzeichen interpretiert. Der tatsächliche Wortabstand wird für jeden Absatz automatisch optimiert.

Body

Der Body wird durch die `document`-Umgebung definiert.

- ▶ Text erscheint automatisch im Blocksatz.
- ▶ Jede beliebige Anzahl an Leerzeichen und einzelner Zeilenumbrüche werden als einziges Leerzeichen interpretiert. Der tatsächliche Wortabstand wird für jeden Absatz automatisch optimiert.
- ▶ Ein Zeilenumbruch kann jederzeit mit `\\` erzwungen werden.
Ein doppelter Zeilenumbruch beginnt einen neuen Absatz.

Body II

- ▶ Um den Text zu strukturieren, kann man ihn in
 - ▶ Kapitel (*chapter* oder *part*),

Body II

- ▶ Um den Text zu strukturieren, kann man ihn in
 - ▶ Kapitel (*chapter* oder *part*),
 - ▶ Abschnitte(*section*) und Unterabschnitte(*(sub-)subsection*) unterteilen.

Werden automatisch durchnummeriert(1.1, 1.2, 1.2.1, ...)

Body II

- ▶ Um den Text zu strukturieren, kann man ihn in
 - ▶ Kapitel (*chapter* oder *part*),
 - ▶ Abschnitte(*section*) und Unterabschnitte(*(sub-)subsection*) unterteilen.

Werden automatisch durchnummeriert(1.1, 1.2, 1.2.1, ...) und erscheinen im Inhaltsverzeichnis, das man mit

- ▶ `\tableofcontents` erzeugen kann.

Beispiel: Body

```
\begin{document}
\maketitle
\tableofcontents
\section{Hauptteil}
Dies ist etwas Text.
\subsection{Unterabschnitt}
Dieser Unterabschnitt enthält ebenfalls Text.

Und einen zweiten Absatz.
\section{Sonstwas}
Hier steht auch noch Kram.
\end{document}
```

Ergebnis (Preamble und Body)

Ein Text

Ango

5. April 2015

Inhaltsverzeichnis

1	Hauptteil	1
1.1	Unterabschnitt	1
2	Sonstwas	1

1 Hauptteil

Dies ist etwas Text.

1.1 Unterabschnitt

Dieser Unterabschnitt enthält ebenfalls Text.

Und einen zweiten Absatz.

2 Sonstwas

Hier steht auch noch Kram.

Formatierung I

- ▶ Texterscheinung festlegen mit `\textbf{fetter Text}`,
`\textit{kursiver Text}`,...

Formatierung I

- ▶ Texterscheinung festlegen mit `\textbf{fetter Text}`,
`\textit{kursiver Text}`,...
- ▶ Textgröße z. B.
`\tiny`, `\small`, `\normsize`, `\large`, `\huge`

Formatierung I

- ▶ Texterscheinung festlegen mit `\textbf{fetter Text}`,
`\textit{kursiver Text}`,...
- ▶ Textgröße z. B.
`\tiny`, `\small`, `\normsize`, `\large`, `\huge`
- ▶ Textfarbe `\color{Farbe}`
(benötigt das Zusatzpaket *color*).

Formatierung II

Die Befehle für Textgröße und -Farbe (z. B. `\large`, und `\color{red}`) sind keine Umgebungen, sondern wirken auf allem folgenden Text. Soll die Änderung nur für begrenzte Bereiche gelten, muss man sie entweder wieder zurücksetzen, oder den Befehl in eine `{}`-Umgebung einbetten, z. B.:

```
Normaler Text ... {\LARGE größer Text \color{red}
großer roter Text} ... weiterer normaler Text.
```

Formatierung II

Die Befehle für Textgröße und -Farbe (z. B. `\large`, und `\color{red}`) sind keine Umgebungen, sondern wirken auf allem folgenden Text. Soll die Änderung nur für begrenzte Bereiche gelten, muss man sie entweder wieder zurücksetzen, oder den Befehl in eine `{}`-Umgebung einbetten, z. B.:

```
Normaler Text ... {\LARGE größer Text \color{red}
großer roter Text} ... weiterer normaler Text.
```

Normaler Text ... größer Text **großer roter Text** ...
weiterer normaler Text.

Auflistungen I

Für Auflistungen gibt es drei verschiedene Umgebungen:

1. *itemize* erzeugt Listen, in deren Elemente alle mit dem gleichen Symbol markiert sind.

Auflistungen I

Für Auflistungen gibt es drei verschiedene Umgebungen:

1. *itemize* erzeugt Listen, in deren Elemente alle mit dem gleichen Symbol markiert sind.
2. *enumerate* erzeugt Listen, in denen die Elemente automatisch durchgezählt werden.

Auflistungen I

Für Auflistungen gibt es drei verschiedene Umgebungen:

1. *itemize* erzeugt Listen, in deren Elemente alle mit dem gleichen Symbol markiert sind.
 2. *enumerate* erzeugt Listen, in denen die Elemente automatisch durchgezählt werden.
 3. *description* erzeugt Listen, bei denen man für jedes Element eine eigene Markierung festlegen muss.
- ▶ Innerhalb der Aufzählungs-Umgebung beginnen die einzelnen Elemente mit `\item`.

Auflistungen II

Ein Beispiel für eine nummerierte Aufzählung:

```
\begin{enumerate}
  \item Erstes Element
  \item Zweites
  \item Noch eines.
\end{enumerate}
```

Auflistungen II

Ein Beispiel für eine nummerierte Aufzählung:

```
\begin{enumerate}  
  \item Erstes Element  
  \item Zweites  
  \item Noch eines.  
\end{enumerate}
```

1. Erstes Element
2. Zweites
3. Noch eines.

Auflistungen II

Ein Beispiel für eine nummerierte Aufzählung:

```
\begin{enumerate}
  \item Erstes Element
  \item Zweites
  \item Noch eines.
\end{enumerate}
```

1. Erstes Element
2. Zweites
3. Noch eines.

- ▶ Die Art der Nummerierung kann durch ein optionales Argument geändert werden, z. B. `\begin{enumerate}[a]`

Tabellen I

Tabellen werden durch die *tabular* Umgebung erzeugt.

- ▶ Als Parameter wird die Textausrichtung für jede Spalte und Position von vertikalen Trennlinien übergeben.

Tabellen I

Tabellen werden durch die *tabular* Umgebung erzeugt.

- ▶ Als Parameter wird die Textausrichtung für jede Spalte und Position von vertikalen Trennlinien übergeben.
- ▶ In einer Zeile werden die einzelnen Zellen durch & getrennt und das Ende der Zeile durch `\` markiert.

Tabellen II

Ein Beispiel:

```
\begin{tabular}{l|cc|rl}  
1-Spalte & c-Spalte 1 & c-Spalte 2 & & \\ & r-Spalte & l-Spalte 2 & \\\hline  
1: & abc & def & ... & +\\  
2: & ghi & jkl & .. & +  
\end{tabular}
```

Ergebnis:

l-Spalte	c-Spalte 1	c-Spalte 2	r-Spalte	l-Spalte 2
1:	abc	def	...	+
2:	ghi	jkl	..	+

Die math-Umgebung

Eine der Hauptmotivationen bei der Konzeption von T_EXwar, mathematische Formeln einfach und ansehlich darstellbar zu machen.

Die math-Umgebung

Eine der Hauptmotivationen bei der Konzeption von T_EXwar, mathematische Formeln einfach und ansehlich darstellbar zu machen.

- ▶ Die Umgebung `displaymath` ermöglicht die Darstellung von mathematischer Notation und Symbolen.

Die math-Umgebung

Eine der Hauptmotivationen bei der Konzeption von T_EXwar, mathematische Formeln einfach und ansehlich darstellbar zu machen.

- ▶ Die Umgebung `displaymath` ermöglicht die Darstellung von mathematischer Notation und Symbolen.
- ▶ Der Schriftsatz ist verändert (an Formelsetzung angepasst).
- ▶ Die `align`-Umgebung bietet einen leicht ergänzten Mathe-Modus, mit Ausrichtungsmöglichkeiten und Formelnummerierung.

Math-Beispiel I

Eine abgesetzte Formel:

```
\begin{align}  
\sum_{i=1}^{\infty} \frac{1}{2^i}=1  
\end{align}
```

Tadaa!

Math-Beispiel I

Eine abgesetzte Formel:

```
\begin{align}  
\sum_{i=1}^{\infty} \frac{1}{2^i}=1  
\end{align}
```

Tadaa!

Eine abgesetzte Formel:

$$\sum_{i=1}^{\infty} \frac{1}{2^i} = 1 \tag{1}$$

Tadaa!

Math-Beispiel II

Eine Formel in-line: $\sum_{i=1}^{\infty} \frac{1}{2^i} = 1$
Tadaa!

Math-Beispiel II

Eine Formel in-line: $\sum_{i=1}^{\infty} \frac{1}{2^i} = 1$
Tadaa!

Eine Formel in-line: $\sum_{i=1}^{\infty} \frac{1}{2^i} = 1$
Tadaa!

AMS-Pakete

(AMS steht für AMERICAN MATHEMATICS SOCIETY)

- ▶ Die Pakete `amsmath` und `amssymb` verbessern den Mathematikmodus und fügen weitere Symbole und Funktionalitäten hinzu.

AMS-Pakete

(AMS steht für AMERICAN MATHEMATICS SOCIETY)

- ▶ Die Pakete `amsmath` und `amssymb` verbessern den Mathematikmodus und fügen weitere Symbole und Funktionalitäten hinzu.
- ▶ Das Paket `amsthm` fügt fertig formatierte Umgebungen hinzu, um Definitionen, Sätze und Beweise zu formulieren.

- ▶ Kommentare in der tex-datei mit % Kommentar
(Kommentar geht bis zum nächsten Zeilenumbruch)

- ▶ Kommentare in der tex-datei mit % Kommentar
(Kommentar geht bis zum nächsten Zeilenumbruch)
- ▶ Fussnoten können einfach durch
`\footnote{Fussnotentext...}` in den Text eingefügt
werden.

- ▶ Kommentare in der tex-datei mit % Kommentar
(Kommentar geht bis zum nächsten Zeilenumbruch)
- ▶ Fussnoten können einfach durch
`\footnote{Fussnotentext...}` in den Text eingefügt
werden.
- ▶ Mit dem Befehl `\pagestyle{...}` kann der Seitenaufbau
geändert werden (insbesondere Kopf- und Fusszeilen). Als
Argumente können vordefinierte oder selbsterstellte
Formatierungen gewählt werden.

Bilder einfügen

- ▶ Das Paket *graphicx* ermöglicht das unkomplizierte einbinden von Bildern:

Bilder einfügen

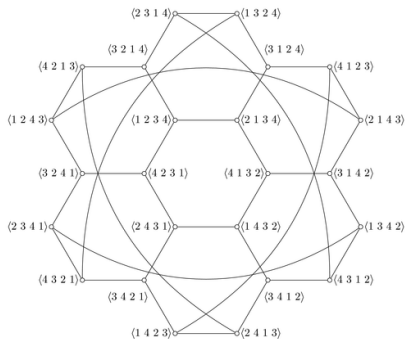
- ▶ Das Paket *graphicx* ermöglicht das unkomplizierte einbinden von Bildern:
- ▶ `\includegraphics{bild.png}` fügt das übergebene Bild an der aktuellen Position im Dokument ein.

Bilder einfügen

- ▶ Das Paket *graphicx* ermöglicht das unkomplizierte einbinden von Bildern:
- ▶ `\includegraphics{bild.png}` fügt das übergebene Bild an der aktuellen Position im Dokument ein.
- ▶ Optionale Parameter ermöglichen u. A. Zuschneiden, Skalieren und Drehen des Bildes.

Zeichnen

- Das Paket *tikz* bietet ein mächtiges Zeichentool, mit dem u. A. Plots, Graphen und Diagramme erstellt werden können.

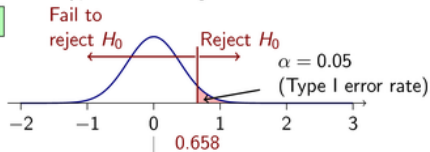


Was ist L^AT_EX?
 Grundlagen – Ein einfaches Dokument erstellen
 Tolle Sachen – Was noch alles möglich ist
 Hilfreiche Pakete



Statistical power in hypothesis testing:

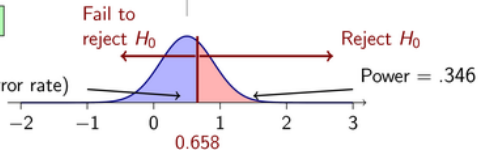
H_0 is true:



Effect size: 0.5

H_a is true:

$\beta = .654$
(Type II error rate)



Programmcode

- ▶ Das Paket *algorithm2e* ermöglicht die Notation von übersichtlichem Pseudocode mit automatischer Zeilennummerierung und Einrückung.

Programmcode

- ▶ Das Paket *algorithm2e* ermöglicht die Notation von übersichtlichem Pseudocode mit automatischer Zeilennummerierung und Einrückung.
- ▶ Außerdem sind viele Programmierstrukturen (Verzweigungen, Schleifen, etc.) Befehle implementiert, die das Schreiben von Pseudocode vereinfachen.

Präsentationen

- ▶ Mit dem Paket *Beamer* können Präsentationen wie diese erstellt werden.
- ▶ Eine Folie wird durch eine *frame*-Umgebung definiert
- ▶ Für das globale Design (Titel- und Fussleisten) gibt es auch eine große Auswahl an Vorlagen.

TRUSTWORTHINESS OF INFORMATION BY FILE EXTENSION

