Beamer-LZU

A Lanzhou University presentation theme for LaTeX Beamer

Sample presenter

November 28, 2026

School of Information Science and Engineering, Lanzhou University

Outline

LaTeX and Beamer 1
- LaTeX 2
- Beamer 3

Animated slides 4
- Incremental list 5
- Reveal an answer 6
- Replace content 7
- Equation derivation 8

Diagrams and layout 9
- Geometric diagram 10
- State diagram 11
- Theorem and proof 12

More features 13
- Two columns 14
- Automatic frame breaks 15

LaTeX and Beamer

LaTeX and Beamer
- LaTeX
- Beamer

Animated slides
- Incremental list
- Reveal an answer
- Replace content
- Equation derivation

Diagrams and layout
- Geometric diagram
- State diagram
- Theorem and proof

More features
- Two columns
- Automatic frame breaks

LaTeX

Text, equations, and citations can live in one source file. Recompile after changing the section order to update the outline and navigation.

An inline equation such as $e^{i\pi} + 1 = 0$ flows with the text. A displayed equation is centered on its own line:

$$\int_0^1 x^2 \,\mathrm{d}x = \frac{1}{3}.$$

Beamer

A logical slide is written in one `frame` environment. It may have several overlay pages that share the same frame number.

Select a section title in the header to jump to its overview page, or select a dot to jump to a content frame. Beamer's overlay syntax is described in its user guide[1].

Animated slides

LaTeX and Beamer
- LaTeX
- Beamer

Animated slides
- Incremental list
- Reveal an answer
- Replace content
- Equation derivation

Diagrams and layout
- Geometric diagram
- State diagram
- Theorem and proof

More features
- Two columns
- Automatic frame breaks

How can we compare the running times of two algorithms?

Test algorithms A and B on the same inputs.

1. **Measure a baseline.** Run A on each input and record its time.

How can we compare the running times of two algorithms?

Test algorithms A and B on the same inputs.

1. **Measure a baseline.** Run A on each input and record its time.
2. **Change the algorithm.** Keep the inputs, machine, and timing method fixed while running B.

How can we compare the running times of two algorithms?

Test algorithms A and B on the same inputs.

1. **Measure a baseline.** Run A on each input and record its time.
2. **Change the algorithm.** Keep the inputs, machine, and timing method fixed while running B.
3. **Compare pairs.** Compute $t_A - t_B$ for each input; a positive value means B is faster.

Which measurement best represents the typical running time?

The same program was run three times. Which reading is the median?

$2.1\,\mathrm{s}$ $9.8\,\mathrm{s}$ $2.0\,\mathrm{s}$

Which measurement best represents the typical running time?

The same program was run three times. Which reading is the median?

$$2.1\,\mathrm{s} \qquad 9.8\,\mathrm{s} \qquad 2.0\,\mathrm{s}$$

Sort the times $2.0 < 2.1 < 9.8$

The middle reading is $2.1\,\mathrm{s}$, from the first run.

How does the mean change as samples arrive?

Three samples arrive in order. Recompute the mean from the samples seen so far.

Arrival	First	Second	Third
Value	2		

Current mean $\bar{x}_1 = 2$

How does the mean change as samples arrive?

Three samples arrive in order. Recompute the mean from the samples seen so far.

Arrival	First	Second	Third
Value	2	4	

Current mean $\bar{x}_2 = \dfrac{2+4}{2} = 3$

How does the mean change as samples arrive?

Three samples arrive in order. Recompute the mean from the samples seen so far.

Arrival	First	Second	Third
Value	2	4	6

Current mean $\bar{x}_3 = \dfrac{2+4+6}{3} = 4$

Deriving the square of a sum

Multiply two identical factors.

$$(x+1)^2 = (x+1)(x+1)$$

Deriving the square of a sum

Multiply two identical factors.

$$\begin{aligned}(x+1)^2 &= (x+1)(x+1) \\ &= x^2 + x + x + 1 \quad \text{expand the product}\end{aligned}$$

Deriving the square of a sum

Multiply two identical factors.

$$
\begin{aligned}
(x+1)^2 &= (x+1)(x+1) \\
&= x^2 + x + x + 1 && \text{expand the product} \\
&= x^2 + 2x + 1 && \text{collect like terms}
\end{aligned}
$$

Diagrams and layout

LaTeX and Beamer
- LaTeX
- Beamer

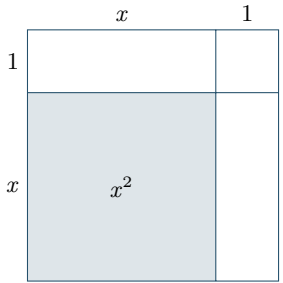
Animated slides
- Incremental list
- Reveal an answer
- Replace content
- Equation derivation

Diagrams and layout
- Geometric diagram
- State diagram
- Theorem and proof

More features
- Two columns
- Automatic frame breaks

Explaining the identity by area

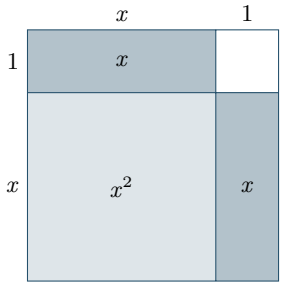
How can a square of side $x + 1$ be divided? Assume $x > 0$.



The lower-left square has area x^2.

Explaining the identity by area

How can a square of side $x + 1$ be divided? Assume $x > 0$.

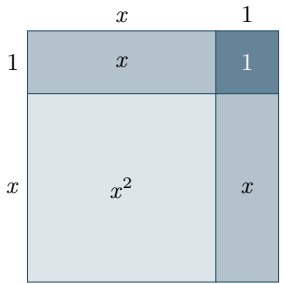


The lower-left square has area x^2.

The two rectangles each have area x, for a total of $2x$.

Explaining the identity by area

How can a square of side $x + 1$ be divided? Assume $x > 0$.



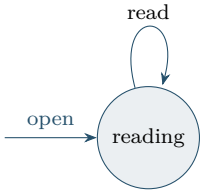
The lower-left square has area x^2.

The two rectangles each have area x, for a total of $2x$.

The small square adds 1.

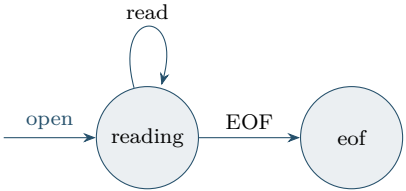
$(x+1)^2 = x^2 + 2x + 1$

How does an open file reach the closed state?



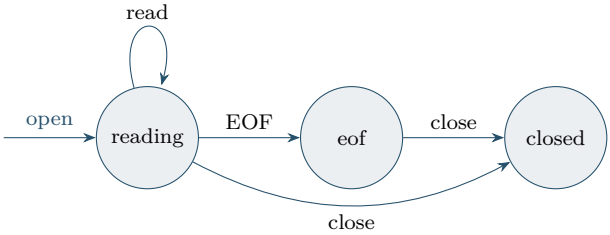
The file remains readable after a successful read.

How does an open file reach the closed state?



An EOF result indicates that the end has been reached.

How does an open file reach the closed state?



Both closing paths are valid: a file may be closed during reading or after EOF.

Theorem and proof

Theorem

The square of every odd integer is odd.

Proof.

Write $n = 2k + 1$ for some $k \in \mathbb{Z}$. Then

$$n^2 = 4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1.$$

Thus n^2 is an even integer plus one. □

More features

LaTeX and Beamer
- LaTeX
- Beamer

Animated slides
- Incremental list
- Reveal an answer
- Replace content
- Equation derivation

Diagrams and layout
- Geometric diagram
- State diagram
- Theorem and proof

More features
- Two columns
- Automatic frame breaks

Two-column layout

Experiment setup

Fix the input size and time the same program three times. Restore the same initial state before each run.

Measurements

Run	1	2	3
Time (s)	2.1	1.9	2.0

The mean time is 2.0 seconds.

A reproducible timing experiment I

Question and measurement. Suppose two implementations solve the same problem. A faster result on one input is useful evidence, but it is not enough to establish a general performance claim. We need a set of inputs that covers the small, typical, and difficult cases we expect in practice. For each input, record the elapsed time of both implementations and retain the input itself so the comparison can be repeated later.

Controlled conditions. Run both programs on the same machine with the same compiler settings and software environment. Background processes, power management, file caches, and the order in which programs run can affect a timing result. Record these conditions, and keep them as stable as possible. If a program reads data from disk, decide whether the experiment measures a cold cache, a warm cache, or both. State the choice clearly.

A reproducible timing experiment II

Repeated runs. Repeat each measurement several times. A single run can be unusually slow because another process used the CPU or because the operating system delayed an operation. Report a summary such as the median together with the spread of the observations. Keep the individual readings so that outliers can be investigated rather than silently removed. If the program has a warm-up phase, describe it and apply it consistently.

Paired comparison. For every input, compare the two timings as a pair. This keeps the effect of input difficulty visible. A table of paired differences can show whether one implementation wins consistently or only on a subset of cases. An overall average can conceal an important slow case, so include enough detail for readers to see the range of outcomes. Graphs should label their units and explain whether the values are raw times, differences, or ratios.

A reproducible timing experiment III

Interpretation. Performance measurements answer a question about the tested workload and environment. They do not automatically predict what will happen on every machine or every future input. Identify the limits of the experiment, including the input selection and any sources of timing noise. When reporting a conclusion, tie it to the evidence: say which implementation was faster on which inputs and by how much. A clear methods description makes the claim easier to check and more useful to others.

Reproduction. Share the source code, input generator, build settings, and measurement script when possible. A reader should be able to recreate the table from the supplied materials. If the numbers change on a different machine, the documented procedure still allows the reader to understand the difference. This long example also demonstrates Beamer's automatic frame breaks: the text continues on the next PDF page without writing a separate frame environment.

Presentation notes

Present the generated PDF directly. For a printed handout, add `handout` to the document class options. Use `handout:` in mutually exclusive overlays to select which alternative remains visible.

References


Till Tantau et al., *The Beamer Class User Guide*. CTAN package page.


Oscar Tuo, *Touying-LZU*, v0.1.1. Release mirror.



蘭州大學
LANZHOU UNIVERSITY

Thank you for listening!