LaTeX Presentation Template

a simple and clean beamer package

Presenter Name

Complete Institute Name

October 12, 2025

Table of Contents

Table of Contents

Motivation
subtitle

This is the first highlighted keyword to emphasize an important concept.

The second point addresses another key idea in Knuth 1984.

Objectives Scope

Sample Block Title

This block presents a key concept that is crucial for understanding the topic.

Sample Alert Block Title

This block presents a more alarming key concept that is crucial for understanding the topic.

Actors & Features

Actors:

Features:

Contributions

Scientific Contribution

Real-world Contribution

Table of Contents

Advancements

Research gaps

Research gap

⇒ **Concluding statement.**

Table of Contents

Overview

Figure. The caption of the figure.

Sample Process

Algorithm Pseudocode

Goal:

Result:

Step:

Scope:

Table of Contents

Prototyping

GitHub repository:
Demo Website:

Figure. The caption of the figure.

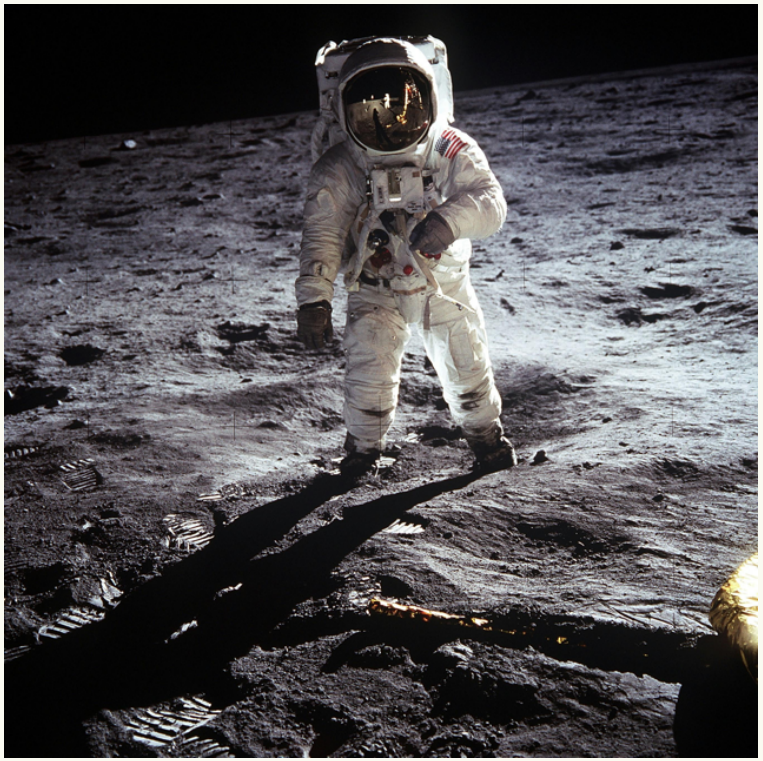

Figure. The caption of the figure.

Table of Contents

Limitations

⇒ **Concluding statement.**

Comparison

Table. Comparison of different methods (✓: YES, ✗: NO).

	Your Method	Method B	Method C	Method D	Method E	Method F
Feature 1	✓	✓	✗	✓	✗	✓
Feature 2	✓	✗	✓	✓	✓	✗
Feature 3	✗	✓	✓	✗	✗	✓
Feature 4	✓	✓	✗	✗	✓	✗
Feature 5	✗	✗	✓	✓	✗	✓
Feature 6	✓	✗	✓	✗	✗	✗

Table of Contents

Demonstration

Process A

Scenario 1

Scenario 2

Process B


thank
you!

Scope

Back to Objectives

Formalizing - `Sample` Algorithm

Back to `Sample` process

Algorithm (Result) ← `Sample`(Input1)

Require: Input1 is a predefined parameter.

```
1:  Set ← ∅
2:  for element ∈ Input1 do
3:      if Condition(element) is true then
4:          Set ← Set ∪ {Process(element)}
5:      else
6:          continue
7:      end if
8:  end for
9:  Intermediate ← Transform(Set)
10: return Result
```

Formalizing - `Sample` Pseudocode

Back to `Sample` process

Algorithm (Result) ← `Sample`(Input1)

Require: Input1 is a predefined parameter.

```
Set ← ∅
for element ∈ Input1 do
    if Condition(element) is true then
        Set ← Set ∪ {Process(element)}
    else
        continue
    end if
end for
Intermediate ← Transform(Set)
return Result
```

References I

[Knu84] Donald E. Knuth. “Literate Programming”. In: *The Computer Journal* 27.2 [1984], pp. 97–111.