Claude + Typst

Examples for AI-Assisted Document Generation

Version 1.0 • 2026-01-07

Table of Contents

1 Introduction 4
2 Title Page Variations 4
  2.1 Formal Academic Style 4
  2.2 Technical Report Style 5
3 Tables and Data 6
  3.1 Basic Comparison Table 6
  3.2 Styled Data Table with Alternating Rows 6
4 Code Examples 7
  4.1 Python with Syntax Highlighting 7
  4.2 JSON Configuration Example 7
5 Mathematical Content 8
  5.1 Inline Mathematics 8
  5.2 Display Equations 8
  5.3 Numbered Equations 8
6 Lists and Structured Content 9
  6.1 Nested Bullet Lists 9
  6.2 Numbered Lists with Descriptions 9
7 Two-Column Layout 11
  7.1 Abstract 11
  7.2 Introduction 11
  7.3 Methodology 11
  7.4 Results 11
  7.5 Discussion 12
  7.6 Conclusion 12
8 Figures and Captions 13
9 Headers and Footers 14
10 Callout Boxes and Highlights 14
11 Block Quotes 15
12 Footnotes and Endnotes 15
13 Meeting Minutes Template 16
  13.1 Agenda Items 16
    13.1.1 1. Project Status Updates 16
    13.1.2 2. Q1 Planning 16
    13.1.3 3. Technical Debt Review 16
  13.2 Next Meeting 17
14 Request for Proposal (RFP) Format 18
  14.1 Executive Summary 18
  14.2 Scope of Work 18
  14.3 Technical Requirements 18

14.3.1 Functional Requirements 18
14.3.2 Integration Requirements 19
14.3.3 Performance Requirements 19
14.4 Proposal Requirements 19
14.5 Evaluation Criteria 19
14.6 Submission Instructions 19
15 Conclusion 21
16 Addendum: Source Code Examples 22
16.1 Document Setup 22
16.2 Title Page Construction 22
16.3 Styled Tables with Alternating Rows 23
16.4 Mathematical Equations with Numbering 23
16.5 Code Blocks with Syntax Highlighting 24
16.6 Callout Boxes with Color Coding 24
16.7 Headers and Footers 25
16.8 Two-Column Layout 25
16.9 Figure Placeholders with Cross-References 26
16.10 Block Quotes with Attribution 26

1 Introduction

This document demonstrates various Typst document styles and elements through concrete examples. Each section shows a different document type or formatting technique that Claude Code can generate when properly instructed.

2 Title Page Variations

2.1 Formal Academic Style

Machine Learning Approaches to Natural Language Processing

A Comprehensive Survey

Jane M. Researcher, PhD

Department of Computer Science

University of Technology

January 2026

Submitted in partial fulfillment of the requirements

for the degree of Doctor of Philosophy

2.2 Technical Report Style

System Architecture Specification

Document No: ARCH-2026-001

Revision: 1.2

Classification: Internal Use Only

Author: Engineering Team	**Reviewed By:** Architecture Board
Date: January 7, 2026	**Approved By:** CTO
Status: Draft	**Next Review:** Q2 2026

3 Tables and Data

3.1 Basic Comparison Table

Feature	Solution A	Solution B	Notes
Performance	High	Medium	Based on bench-mark tests
Cost	$5,000	$12,000	Annual license
Scalability	Good	Excellent	Horizontal scaling supported
Support	Community	Enterprise	24/7 for Solution B
Learning Curve	Moderate	Steep	Documentation quality varies
Integration	REST API	REST + GraphQL	Both offer web-hooks

Table 1: Feature comparison between solutions

3.2 Styled Data Table with Alternating Rows

ID	Description	Units	Price	Status
REQ-001	User authentication system	1	$25,000	Complete
REQ-002	Payment processing gateway	1	$40,000	In Progress
REQ-003	Reporting dashboard	3	$15,000	Planned
REQ-004	Mobile application	2	$60,000	In Progress
REQ-005	API rate limiting	1	$8,000	Complete
REQ-006	Data export functionality	1	$12,000	Planned

Table 2: Project requirements and status tracking

4 Code Examples

4.1 Python with Syntax Highlighting

The following example demonstrates API client implementation:

```python
import requests
from typing import Dict, Optional

class APIClient:
    """Client for interacting with the REST API."""

    def __init__(self, base_url: str, api_key: str):
        self.base_url = base_url
        self.headers = {
            "Authorization": f"Bearer {api_key}",
            "Content-Type": "application/json"
        }

    def get_user(self, user_id: int) -> Optional[Dict]:
        """Retrieve user information by ID."""
        response = requests.get(
            f"{self.base_url}/users/{user_id}",
            headers=self.headers
        )

        if response.status_code == 200:
            return response.json()
        return None
```

4.2 JSON Configuration Example

```json
{
  "server": {
    "host": "0.0.0.0",
    "port": 8080,
    "workers": 4
  },
  "database": {
    "type": "postgresql",
    "host": "localhost",
    "port": 5432,
    "name": "production_db",
    "pool_size": 20
  },
  "logging": {
    "level": "INFO",
    "format": "json",
    "output": "/var/log/app.log"
  }
}
```

5 Mathematical Content

5.1 Inline Mathematics

The quadratic formula states that for $ax^2 + bx + c = 0$, the solutions are given by $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$, where $a \neq 0$.

The fundamental theorem of calculus relates differentiation and integration: if f is continuous on $[a, b]$, then $\int_a^b f(x)\,\mathrm{d}x = F(b) - F(a)$ where $F'(x) = f(x)$.

5.2 Display Equations

The normal distribution probability density function:

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Euler's identity, considered one of the most beautiful equations in mathematics:

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

The Taylor series expansion of a function $f(x)$ about point a:

$$f(x) = f(a) + f'(a)(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \frac{f'''(a)}{3!}(x - a)^3 + \ldots$$

5.3 Numbered Equations

$$E = mc^2 \tag{1}$$

$$\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t} \tag{2}$$

$$\nabla \cdot \boldsymbol{E} = \frac{\rho}{\varepsilon_0} \tag{3}$$

Einstein's mass-energy equivalence (Equation 1) and Maxwell's equations (Equation 2, Equation 3) form the foundation of modern physics.

6 Lists and Structured Content

6.1 Nested Bullet Lists

Project deliverables include:

- Phase 1: Foundation
  - Infrastructure setup
    - Cloud environment configuration
    - CI/CD pipeline establishment
    - Monitoring and logging systems
  - Core architecture
    - Database schema design
    - API specification
    - Authentication framework
- Phase 2: Implementation
  - Backend services
    - User management
    - Payment processing
    - Notification system
  - Frontend development
    - Web application
    - Mobile applications (iOS, Android)
    - Admin dashboard
- Phase 3: Testing and Deployment
  - Quality assurance
  - Performance testing
  - Security audit
  - Production deployment

6.2 Numbered Lists with Descriptions

1. **Requirements Gathering**: Conduct stakeholder interviews and document functional and non-functional requirements. Establish acceptance criteria for each feature.
2. **System Design**: Create architectural diagrams, define data models, and specify API contracts. Review with technical team for feasibility.
3. **Implementation**: Develop features according to specification using iterative development methodology. Conduct code reviews and maintain test coverage above 80%.
4. **Testing and Validation**: Execute comprehensive test plans including unit, integration, and end-to-end testing. Validate against original requirements.

5. **Deployment and Monitoring**: Deploy to production environment with zero-downtime strategy. Implement monitoring and alerting for key metrics.

7 Two-Column Layout

7.1 Abstract

This section demonstrates multi-column layout capabilities in Typst. Two-column formatting is commonly used in academic papers, conference proceedings, and technical reports to maximize page space utilization.

The text flows naturally from one column to the next, maintaining proper justification and hyphenation. Headers, figures, and equations can span multiple columns when needed.

7.2 Introduction

Multi-column layouts present unique typesetting challenges. Text must flow smoothly between columns while maintaining readability. Line lengths should be appropriate for the font size to prevent awkward breaks.

Typst handles these requirements automatically, adjusting spacing and breaks to produce professional results. Column balancing ensures even distribution of content across the page.

7.3 Methodology

The approach taken in this document prioritizes clarity and reproducibility. Each example demonstrates a specific capability with sufficient detail for readers to replicate the results.

Examples progress from simple to complex, building understanding incrementally. Code snippets include complete context rather than isolated fragments.

7.4 Results

The combination of Claude Code and Typst enables rapid generation of professionally formatted documents. Users report significant time savings compared to traditional LaTeX workflows.

Quality of output matches or exceeds hand-crafted alternatives for most document types. The iterative refinement process allows for progressive enhancement without starting over.

7.5 Discussion

Several factors contribute to the effectiveness of this approach. Natural language instructions lower the barrier to entry for document creation. The fast compilation cycle enables immediate feedback.

Typst's modern syntax reduces cognitive load compared to LaTeX. Common operations require less boilerplate and are more intuitive to specify.

7.6 Conclusion

AI-assisted document generation represents a meaningful improvement in productivity for technical writing tasks. The workflow demonstrated here combines the best aspects of markup-based typesetting with conversational interaction.

Future work should explore additional document types and styling variations. Integration with existing documentation pipelines would further enhance utility.

8 Figures and Captions

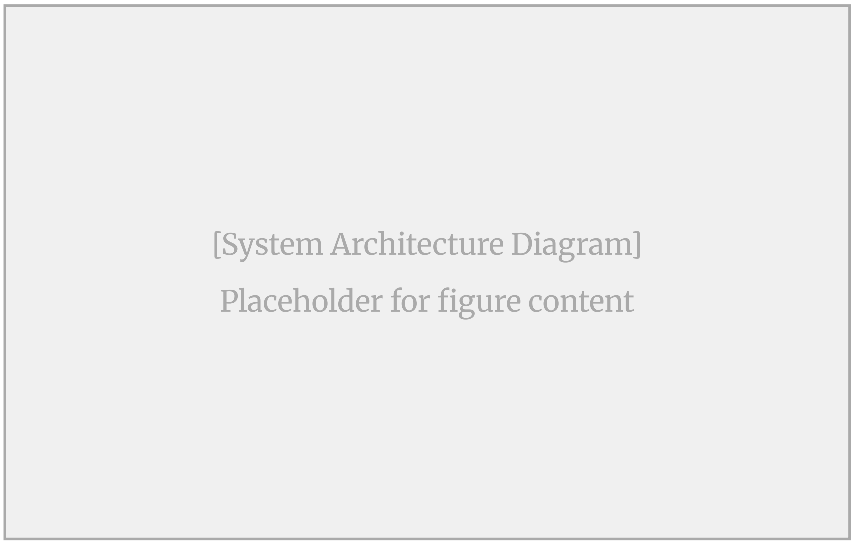

Figure 1: High-level system architecture showing major components and data flows

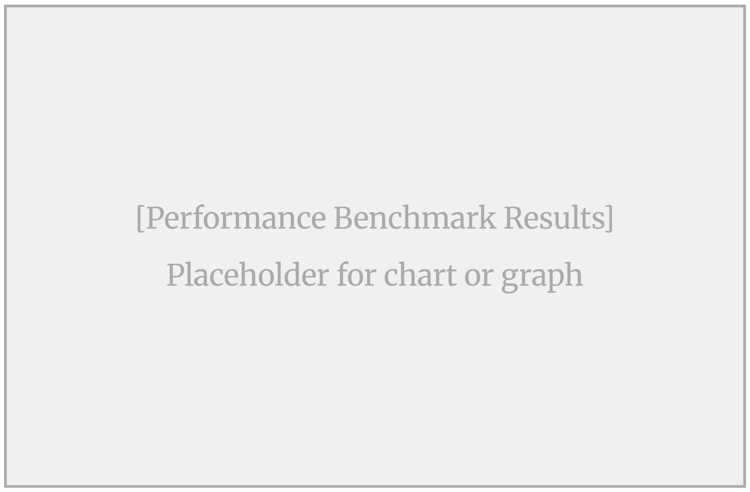
[Performance Benchmark Results]
Placeholder for chart or graph

Figure 2: Response time measurements under varying load conditions

As shown in Figure 1, the system follows a three-tier architecture pattern. Performance characteristics (Figure 2) indicate linear scaling up to 10,000 concurrent users.

9 Headers and Footers

This page and subsequent pages demonstrate custom headers and footers. The header shows the document title and version number, while the footer includes generation date and page numbering.

Headers and footers provide consistent navigation elements throughout longer documents. They can include document metadata, chapter titles, page numbers, or other contextual information.

10 Callout Boxes and Highlights

Note: This is an informational callout box. Use these to highlight important information that deserves special attention from readers.

Warning: This style indicates cautionary information. Readers should pay particular attention to warnings to avoid common pitfalls or errors.

Tip: Helpful suggestions or best practices can be formatted this way to stand out from regular body text while maintaining visual hierarchy.

11 Block Quotes

Extended quotations should be formatted as distinct blocks:

> 1. *A robot may not injure a human being or, through inaction, allow a human being to come to harm.*
> 2. *A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.*
> 3. *A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.*
>
> — Handbook of Robotics, 56th Edition, 2058 A.D

Shorter inline quotations can use standard formatting: "It is only afterward that a new idea seems reasonable. To begin with, it usually seems unreasonable." - also Isaac Asimov.

12 Footnotes and Endnotes

Typst supports footnotes for supplementary information.[1] Footnotes appear at the bottom of the page where they are referenced.

Multiple footnotes[2] can appear in close proximity[3] without causing layout issues. The system handles numbering and positioning automatically.

[1] This is an example footnote providing additional context without disrupting main text flow.

[2] First footnote on this topic.

[3] Second related footnote with additional details.

13 Meeting Minutes Template
Engineering Team Meeting

January 7, 2026 • 2:00 PM – 3:00 PM

Attendees: Sarah Johnson (Chair), Michael Chen, Emma Williams, David Rodriguez, Lisa Thompson

Absent: None

Location: Conference Room B / Zoom

13.1 Agenda Items

13.1.1 1. Project Status Updates

Discussion: Each team member provided updates on their current work streams. The authentication service is on track for next week's release. Frontend team encountered styling issues with the new dashboard.

Action Items:
- Emma to resolve CSS conflicts in dashboard by January 10 (Owner: Emma)
- Michael to complete API documentation review (Owner: Michael)

13.1.2 2. Q1 Planning

Discussion: Reviewed proposed features for Q1 roadmap. Team consensus on prioritizing performance improvements over new features in February. March timeline appears aggressive given current velocity.

Decisions:
- Approved performance optimization sprint for February
- Deferred mobile app enhancements to Q2
- Increased testing coverage requirement to 85%

Action Items:
- Sarah to update roadmap and share with stakeholders by EOW (Owner: Sarah)
- David to prepare performance baseline metrics (Owner: David)

13.1.3 3. Technical Debt Review

Discussion: Identified three critical areas requiring attention: database query optimization, outdated dependencies, and inconsistent error handling across services.

Action Items:

- Schedule technical debt reduction sprint for late January (Owner: Sarah)
- Create tickets for dependency upgrades (Owner: Lisa)
- Document error handling standards (Owner: Michael)

13.2 Next Meeting

Date: January 14, 2026 at 2:00 PM

Location: Conference Room B / Zoom

Agenda Preview: Sprint retrospective, Q1 OKR review, architecture decision on caching layer

14 Request for Proposal (RFP) Format

REQUEST FOR PROPOSAL

Customer Relationship Management System Implementation

RFP Number: 2026-IT-001

Issue Date: January 7, 2026

Response Deadline: February 15, 2026, 5:00 PM PST

14.1 Executive Summary

Acme Corporation is seeking proposals from qualified vendors to implement a comprehensive Customer Relationship Management (CRM) system. The selected solution must support sales, marketing, and customer service operations for approximately 200 users across three office locations.

14.2 Scope of Work

The vendor shall provide:

1. **Software Licensing:** Cloud-based CRM platform with appropriate user licenses
2. **Implementation Services:** System configuration, data migration, and integration
3. **Training:** Comprehensive end-user and administrator training programs
4. **Support:** Ongoing technical support and maintenance services

14.3 Technical Requirements

14.3.1 Functional Requirements

- Contact and account management with custom field support
- Sales pipeline tracking with configurable stages
- Marketing automation including email campaigns and lead scoring
- Customer service ticketing with SLA management
- Reporting and analytics with custom dashboard capabilities

- Mobile access for iOS and Android platforms

14.3.2 Integration Requirements

- Bidirectional sync with Microsoft 365
- Integration with existing ERP system (SAP)
- Email integration (Exchange Server)
- Calendar and scheduling integration
- SSO authentication via Active Directory

14.3.3 Performance Requirements

- System uptime: 99.9% availability
- Page load time: < 2 seconds under normal load
- Support for 200 concurrent users
- Data retention: 7 years minimum
- Backup frequency: Daily incremental, weekly full

14.4 Proposal Requirements

Vendors must submit proposals including:

1. Company background and relevant experience
2. Proposed solution architecture and technical approach
3. Implementation timeline with major milestones
4. Detailed cost breakdown including licensing, implementation, training, and support
5. References from at least three comparable implementations
6. Service level agreements and support model
7. Security and compliance certifications

14.5 Evaluation Criteria

Proposals will be evaluated on:

Criterion	**Weight**
Technical capabilities and features	30%
Implementation approach and timeline	25%
Total cost of ownership	20%
Vendor experience and references	15%
Support and training offerings	10%

14.6 Submission Instructions

Submit proposals electronically to procurement@acme.example.com with subject line "RFP 2026-IT-001 Response." Include all required documentation as PDF attachments.

Questions regarding this RFP should be submitted via email by January 21, 2026. Responses will be published to all vendors by January 24, 2026.

15 Conclusion

This document has presented a range of Typst formatting examples demonstrating capabilities relevant to AI-assisted document generation. From academic papers to business documents, mathematical content to code samples, Typst provides the necessary tools for professional typesetting.

16 Addendum: Source Code Examples

This section provides the Typst source code for key examples demonstrated throughout this document. These snippets can be referenced when instructing Claude to generate similar elements.

16.1 Document Setup

The initial configuration for page layout, typography, and heading numbering:

```
#set page(
  paper: "us-letter",
  margin: (x: 1.5in, y: 1in),
  numbering: "1",
)

#set text(
  font: "EB Garamond",
  size: 13pt,
  lang: "en",
)

#set par(
  justify: true,
  leading: 0.65em,
)

#set heading(numbering: "1.1")
```

16.2 Title Page Construction

Creating a centered title page with multiple text sizes and spacing:

```
#align(center)[
  #v(2in)
  #text(size: 28pt, weight: "bold")[
    Claude + Typst
  ]

  #v(0.5em)
  #text(size: 16pt)[
    Examples for AI-Assisted Document Generation
  ]

  #v(2em)
  #text(size: 13pt)[
    Version 1.0 • #datetime.today().display()
  ]
]

#pagebreak()
```

16.3 Styled Tables with Alternating Rows

Tables with custom colors, alignment, and formatting:

```
// Define function for alternating row colors
#let row-color(idx) = if calc.odd(idx) {
  rgb("#f5f5f5")
} else {
  white
}

#figure(
  table(
    columns: (auto, 1.2fr, 0.8fr, 0.8fr, 1fr),
    align: (left, left, right, right, center),
    stroke: 0.5pt,
    fill: (x, y) => if y == 0 {
      rgb("#333333")
    } else {
      row-color(y)
    },
    table.header(
      [#text(fill: white, weight: "bold")[ID]],
      [#text(fill: white, weight: "bold")[Description]],
      [#text(fill: white, weight: "bold")[Units]],
      [#text(fill: white, weight: "bold")[Price]],
      [#text(fill: white, weight: "bold")[Status]],
    ),
    [REQ-001], [User authentication], [1], [\$25,000], [Complete],
    [REQ-002], [Payment gateway], [1], [\$40,000], [In Progress],
  ),
  caption: [Project requirements tracking]
)
```

16.4 Mathematical Equations with Numbering

Display equations with labels and cross-references:

```
// Enable equation numbering
#set math.equation(numbering: "(1)")

// Create labeled equations
$ E = m c^2 $ <einstein>

$ nabla times bold(E) = - (partial bold(B))/(partial t) $ <faraday>

$ nabla dot bold(E) = rho / epsilon_0 $ <gauss>

// Reference equations in text
Einstein's mass-energy equivalence (@einstein) and
Maxwell's equations (@faraday, @gauss) form the
foundation of modern physics.
```

16.5 Code Blocks with Syntax Highlighting

Including code with automatic syntax highlighting:

~~~~
The following demonstrates API implementation:

```python
import requests
from typing import Dict, Optional

class APIClient:
    def __init__(self, base_url: str, api_key: str):
        self.base_url = base_url
        self.headers = {
            "Authorization": f"Bearer {api_key}"
        }
```
~~~~

16.6 Callout Boxes with Color Coding

Creating highlighted boxes for notes, warnings, and tips:

```
// Information callout
#rect(
  width: 100%,
  inset: 15pt,
  fill: rgb("#e8f4f8"),
  stroke: (left: 4pt + rgb("#2196f3")),
)[
  *Note:* This is an informational callout box.
]

// Warning callout
#rect(
  width: 100%,
  inset: 15pt,
  fill: rgb("#fff3cd"),
  stroke: (left: 4pt + rgb("#ffc107")),
)[
  *Warning:* This indicates cautionary information.
]

// Success/tip callout
#rect(
  width: 100%,
  inset: 15pt,
  fill: rgb("#d4edda"),
  stroke: (left: 4pt + rgb("#28a745")),
)[
  *Tip:* Helpful suggestions can be formatted this way.
]
```

16.7 Headers and Footers

Custom page headers and footers with dynamic content:

```
#set page(
  header: [
    #set text(9pt)
    #grid(
      columns: (1fr, 1fr),
      align(left)[_Claude + Typst Examples_],
      align(right)[Version 1.0]
    )
    #line(length: 100%, stroke: 0.5pt)
  ],
  footer: [
    #line(length: 100%, stroke: 0.5pt)
    #set text(9pt)
    #grid(
      columns: (1fr, 1fr),
      align(left)[Generated: #datetime.today().display()],
      align(right)[
        Page #context counter(page).display("1 of 1", both: true)
      ]
    )
  ]
)
```

16.8 Two-Column Layout

Multi-column text flow with manual column breaks:

```
#columns(2)[
  == Abstract

  This section demonstrates multi-column layout capabilities.
  Text flows naturally from one column to the next.

  #colbreak()

  == Discussion

  Column breaks can be inserted manually using colbreak()
  to control where content splits between columns.
]
```

16.9 Figure Placeholders with Cross-References

Creating figure environments with captions and labels:

```
#figure(
  rect(
    width: 80%,
    height: 200pt,
    stroke: 1pt + gray,
    fill: rgb("#f0f0f0"),
  )[
    #align(center + horizon)[
      #text(size: 11pt, fill: gray)[
        [System Architecture Diagram]

        Placeholder for figure content
      ]
    ]
  ],
  caption: [
    High-level system architecture showing major
    components and data flows
  ]
) <fig-architecture>

// Reference the figure in text
As shown in @fig-architecture, the system follows
a three-tier architecture pattern.
```

16.10 Block Quotes with Attribution

Formatted quotations with author attribution:

```
#block(
  inset: (left: 30pt, right: 30pt, top: 10pt, bottom: 10pt),
  fill: rgb("#fafafa"),
)[
  #set text(style: "italic")

    1. A robot may not injure a human being or, through inaction, allow
a human being to come to harm.
    2. A robot must obey the orders given it by human beings except
where such orders would conflict with the First Law.
    3. A robot must protect its own existence as long as such
protection does not conflict with the First or Second Law.

  #align(right)[
    #text(style: "normal")[— Handbook of Robotics, 56th Edition, 2058
A.D]
  ]
]
```