

# 1 Markdown to LaTeX Comprehensive Guide

This guide demonstrates all supported markdown syntax features for the md-to-luatex converter.

## 1.1 Headings

```
% % % %  
# Heading 1  
## Heading 2  
### Heading 3  
#### Heading 4  
##### Heading 5  
##### Heading 6
```

markdown

## 2 Heading 1

### 2.1 Heading 2

#### 2.1.1 Heading 3

Heading 4

Heading 5

Heading 6

### 2.2 Text Formatting

#### 2.2.1 Bold

```
**bold text** or __bold text__
```

**bold text** or **bold text**

markdown

#### 2.2.2 Italic

```
_italic text_ or *italic text*
```

*italic text* or *italic text*

markdown

#### 2.2.3 Bold + Italic

```
**_bold and italic_* or ***bold and italic***
```

***bold and italic*** or ***bold and italic***

markdown

### 2.2.4 Strikethrough

markdown

```
~~strikethrough text~~
```

~~strikethrough text~~

### 2.2.5 Highlight

markdown

```
==highlighted text==
```

highlighted text

### 2.2.6 Superscript

markdown

```
^superscript^ (e.g., x^2^)
```

<sup>superscript</sup> (e.g., x<sup>2</sup>)

### 2.2.7 Subscript

markdown

```
~subscript~ (e.g., H~2~O)
```

<sub>subscript</sub> (e.g., H<sub>2</sub>O)

## 2.3 Inline Code

markdown

```
`inline code`
```

inline code

## 2.4 Code Blocks

### 2.4.1 Fenced Code Block

markdown

```
```python
def hello():
    print("Hello, World!")
```
```

python

```
def hello():
    print("Hello, World!")
```

### 2.4.2 Code Block Without Language

```
```\nplain text code block\n```
```

markdown

```
plain text code block
```

### 2.4.3 Terminal Block

```
```terminal\n$ command\noutput result\n```
```

markdown

```
$ command\noutput result
```

text

## 2.5 Links

```
[link text](https://example.com)
```

markdown

[link text](https://example.com)

```
[link text](https://example.com "title")
```

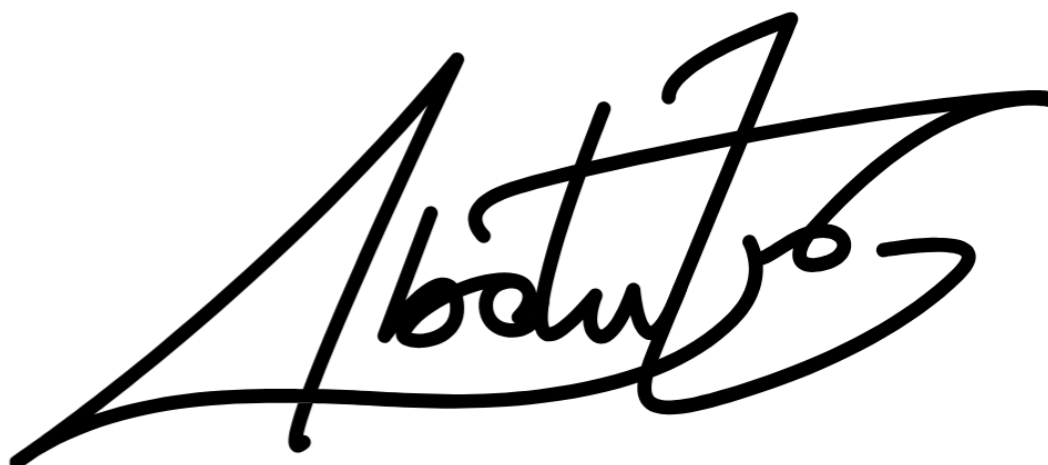
markdown

[link text](https://example.com)

## 2.6 Images

```
![alt text](image.png)
```

markdown



markdown

```
![alt text](image.png "title")
```

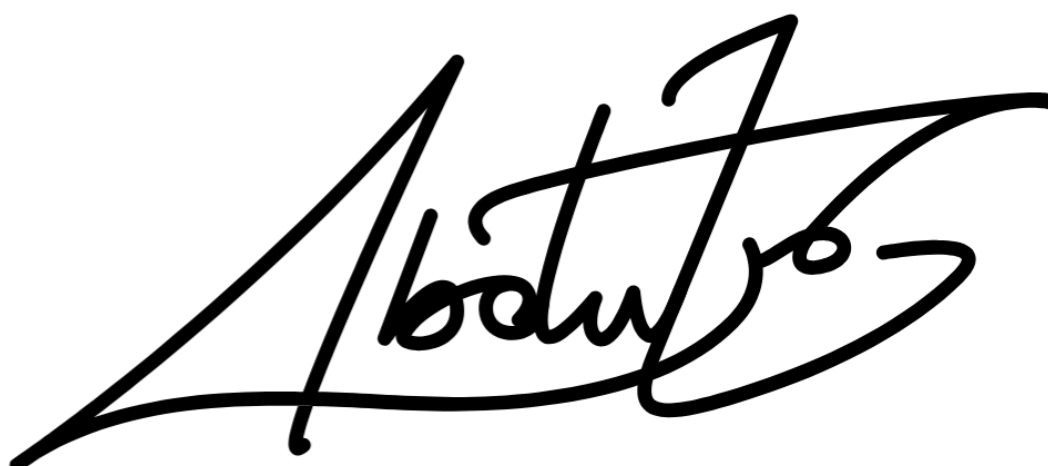
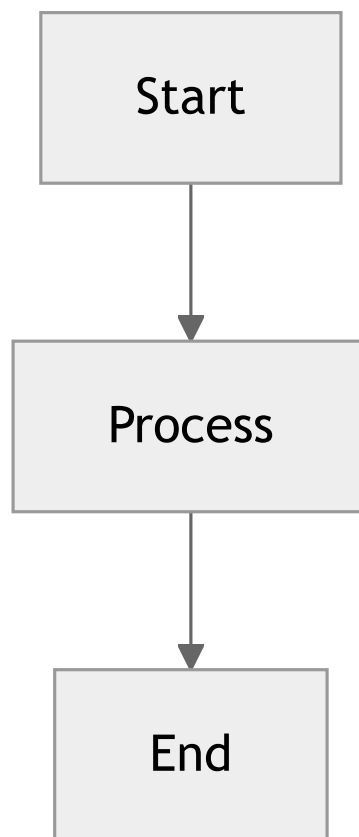


Figure 1: title

## 2.7 Mermaid Diagrams

markdown

```
```mermaid
graph TD
  A[Start] --> B[Process]
  B --> C[End]
```
```



## 2.8 Lists

### 2.8.1 Unordered List

```
- Item 1
- Item 2
  - Nested item 2.1
  - Nested item 2.2
- Item 3
```

markdown

- Item 1
- Item 2
  - Nested item 2.1
  - Nested item 2.2
- Item 3

### 2.8.2 Ordered List

```
1. First item
2. Second item
  1. Nested item 2.1
  2. Nested item 2.2
3. Third item
```

markdown

1. First item
2. Second item
  1. Nested item 2.1
  2. Nested item 2.2
3. Third item

### 2.8.3 Task Lists

markdown

- [x] Completed task
- [ ] Incomplete task
- [/] Partially completed task

- ☒ Completed task
- ☐ Incomplete task
- ☐ Partially completed task

## 2.9 Definition Lists

markdown

```
Term 1
: Definition 1

Term 2
: Definition 2a
: Definition 2b
```

**Term 1**  
Definition 1

**Term 2**  
Definition 2a  
Definition 2b

## 2.10 Tables

### 2.10.1 Pipe Tables

markdown

```
Left	Center	Right	Default
L1	C1	R1	Default
L2	C2	R2	Default
```

| Left | Center | Right | Default |
|------|--------|-------|---------|
| L1   | C1     | R1    | Default |
| L2   | C2     | R2    | Default |

### 2.11 Table with Captions

markdown

```
Header 1	Header 2	Header 3
Cell 1	Cell 2	Cell 3
Cell 4	Cell 5	Cell 6
  
: Sample Table with Caption
```

| Header 1 | Header 2 | Header 3 |
|----------|----------|----------|
| Cell 1   | Cell 2   | Cell 3   |
| Cell 4   | Cell 5   | Cell 6   |

Table 1: Sample Table with Caption

### 2.12 Horizontal Rule

markdown

```
---
```

markdown

```
---
```

markdown

```
***
```

### 2.13 Math Expressions

#### 2.13.1 Inline Math

markdown

```
This is inline math: $E = mc^2$
```

This is inline math:  $E = mc^2$

### 2.13.2 Display Math

```
$$
\int_{0}^{\infty} e^{-x^2} dx = \frac{\sqrt{\pi}}{2}
$$
```

$$\int_0^\infty e^{-x^2} dx = \frac{\sqrt{\pi}}{2}$$

## 2.14 Footnotes

### 2.14.1 Inline Footnote

```
Text with inline footnote^[This is the footnote content].
```

Text with inline footnote<sup>[1]</sup> .

### 2.14.2 Reference Footnote

```
Text with reference footnote[^1].
[^1]: This is the footnote content.
```

Text with reference footnote<sup>[2]</sup> .

## 2.15 Keyboard Shortcuts

### 2.15.1 Single Key

```
[[Ctrl]]
```

Ctrl

### 2.15.2 Key Combination with Plus

```
[[Ctrl]] + [C]
```

Ctrl + C

## 2.16 Line Break

### 2.16.1 Using Two Spaces

Put two spaces at the end of a line 1 to create a line break.

---

[1] This is the footnote content

[2] This is the footnote content.



markdown

```
Line 1<space><space>
Line 2
```

Line 1  
Line 2

### 2.16.2 Using Backslash

markdown

```
Line 1\
Line 2
```

Line 1  
Line 2

### 2.16.3 Using HTML `<br>` Tag

markdown

```
Line 1<br>
Line 2
```

Line 1  
Line 2

markdown

```
Line 1<br/>
Line 2
```

Line 1  
Line 2

### 2.16.4 Using Empty Line (Paragraph Break)

markdown

```
Line 1
```

```
Line 2
```

Line 1  
Line 2

## 2.17 Blockquotes

markdown

```
> This is a blockquote.
> It can span multiple lines.
>
> And multiple paragraphs.
```

This is a blockquote. It can span multiple lines.  
And multiple paragraphs.

### 2.17.1 Nested Blockquotes

markdown

```
> Level 1
>
> > Level 2
> >
> > > Level 3
```

```
Level 1
|
| Level 2
| |
| | Level 3
```

### 2.17.2 Blockquotes with Styled Content

Blockquotes can contain inline code, keyboard shortcuts, tables, and other formatted elements:

markdown

```
> This blockquote contains `inline code` and keyboard shortcuts like [[Ctrl] +
↪ [C]].
>
> It can also have bold, _italic_, and ==highlighted== text.
>
> | Feature | Supported |
> | - - - - - | - - - - - |
> | Tables | Yes |
> | Code | Yes |
```

This blockquote contains `inline code` and keyboard shortcuts like `Ctrl` + `C`.

It can also have **bold**, *italic*, and highlighted text.

| Feature | Supported |
|---------|-----------|
| Tables  | Yes       |
| Code    | Yes       |

### 2.17.3 Blockquotes with Code Blocks

markdown

```
> Here's a code example inside a blockquote:
>
> ```python
> def greet(name):
>     return f"Hello, {name}!"
> ```
>
> The code block maintains its syntax highlighting.
```

Here's a code example inside a blockquote:

```
def greet(name):  
    return f"Hello, {name}!"
```

python

The code block maintains its syntax highlighting.

## 2.18 GitHub Alerts

### 2.18.1 Note Alert

```
\begin{mdalertnote}
```

This is a note alert with blue styling.

```
\end{mdalertnote}
```

markdown

#### NOTE

This is a note alert with blue styling.

### 2.18.2 Tip Alert

```
\begin{mdalerttip}
```

This is a tip alert with green styling.

```
\end{mdalerttip}
```

markdown

#### TIP

This is a tip alert with green styling.

### 2.18.3 Important Alert

```
\begin{mdalertimportant}
```

This is an important alert with purple styling.

```
\end{mdalertimportant}
```

markdown

#### IMPORTANT

This is an important alert with purple styling.

### 2.18.4 Warning Alert

markdown

```
\begin{mdalertwarning}
This is a warning alert with yellow/orange styling.
\end{mdalertwarning}
```

#### WARNING

This is a warning alert with yellow/orange styling.

### 2.18.5 Caution Alert

markdown

```
\begin{mdalertcaution}
This is a caution alert with red styling.
\end{mdalertcaution}
```

#### CAUTION

This is a caution alert with red styling.

### 2.18.6 Alerts with Rich Content

GitHub alerts can also contain formatted text, code, tables, and keyboard shortcuts:

markdown

```
\begin{mdalerttip}
**Pro Tip:** Use `git commit -m "message"` to commit changes.
Common keyboard shortcuts:
- Save: [[Ctrl]] + [S]
- Undo: [[Ctrl]] + [Z]

Command	Description
<code>`git status`</code>	Check repository status
<code>`git log`</code>	View commit history


\end{mdalerttip}
```

#### TIP

**Pro Tip:** Use `git commit -m "message"` to commit changes.  
Common keyboard shortcuts:

- Save: `Ctrl` + `S`
- Undo: `Ctrl` + `Z`

**TIP**

| Command                 | Description             |
|-------------------------|-------------------------|
| <code>git status</code> | Check repository status |
| <code>git log</code>    | View commit history     |

markdown

```
\begin{mdalertwarning}
```

This contains ==highlighted text==, ~~strikethrough~~, and <sup>superscript</sup>!

Code example:

```
``bash
rm -rf /
``
```

**\*\*Never\*\*** run the above command!

```
\end{mdalertwarning}
```

**WARNING**

This contains highlighted text, ~~strikethrough~~, and <sup>superscript</sup>!

Code example:

```
rm -rf /
```

bash

**Never** run the above command!

## 2.19 Executable Python Code Blocks

The converter supports executing Python code blocks directly within your markdown and including their output or generated plots in the final PDF.

### 2.19.1 Prerequisites

For Python code execution to work, you need Python installed on your system. To generate plots with matplotlib, install the required packages:

powershell

```
python -m pip install matplotlib numpy
```

**NOTE**

Mermaid diagram support is already documented in the Mermaid Diagrams section and requires separate installation of `@mermaid-js/mermaid-cli` via npm.

### 2.19.2 Basic Syntax

Mark a Python code block for execution using properties in curly braces:

```
``python {.execute}
print("Hello, World!")
``
```

markdown

### 2.19.3 Available Properties

- **.execute** - Execute the code block (required)
- **.show-code** - Display the source code in the output
- **.show-output** - Display execution output/plot (enabled by default)
- **.hide-code** - Explicitly hide the source code (default)
- **.hide-output** - Hide execution output/plot

### 2.19.4 Simple Print Example

Output only (default):

```
``python {.execute}
print("Hello, World!")
``
```

markdown

Hello, World!

output

Show both code and output:

```
``python {.execute .show-code}
print("Hello, World!")
``
```

markdown

```
print("Hello, World!")
```

python

Hello, World!

output

### 2.19.5 Matplotlib Plots

When your code uses matplotlib, the plot is automatically saved as a PDF and embedded in the document.

Simple plot example (plot only):

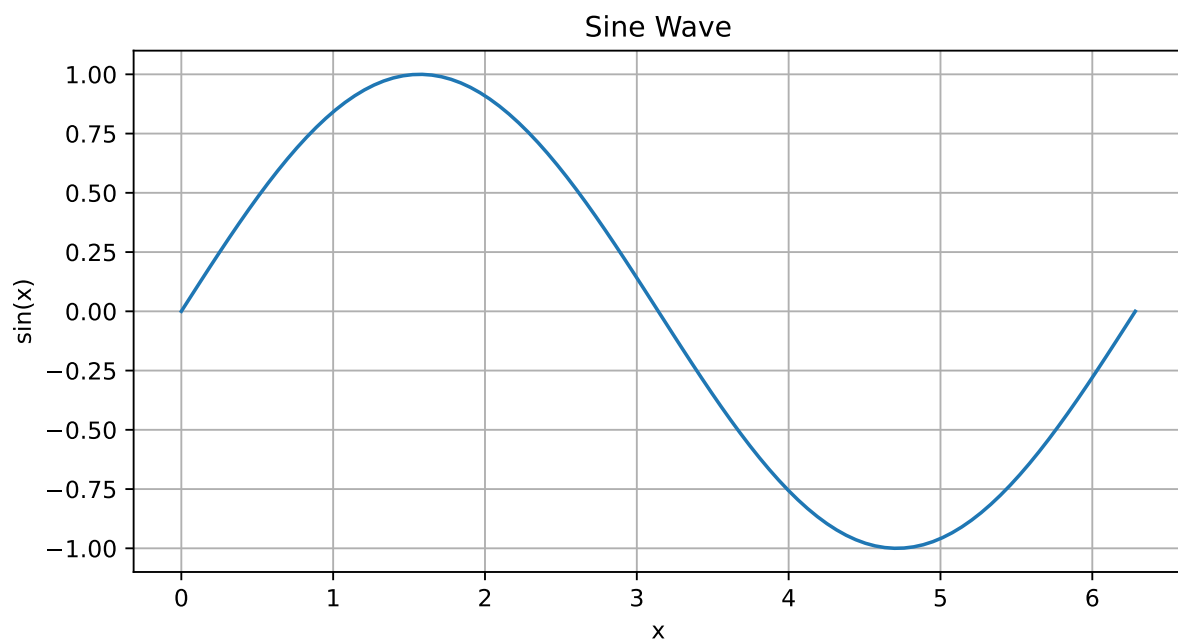
```
``python {.execute}
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 2 * np.pi, 100)
```

markdown

```
y = np.sin(x)

plt.figure(figsize=(8, 4))
plt.plot(x, y)
plt.title('Sine Wave')
plt.xlabel('x')
plt.ylabel('sin(x)')
plt.grid(True)
plt.show()
```



Polar plot with code shown:

```
python { .execute .show-code }
import numpy as np
import matplotlib.pyplot as plt

r = np.arange(0, 2, 0.01)
theta = 2 * np.pi * r
fig, ax = plt.subplots(
    subplot_kw = {'projection': 'polar'}
)
ax.plot(theta, r)
ax.set_rticks([0.5, 1, 1.5, 2])
ax.grid(True)
plt.show()
```

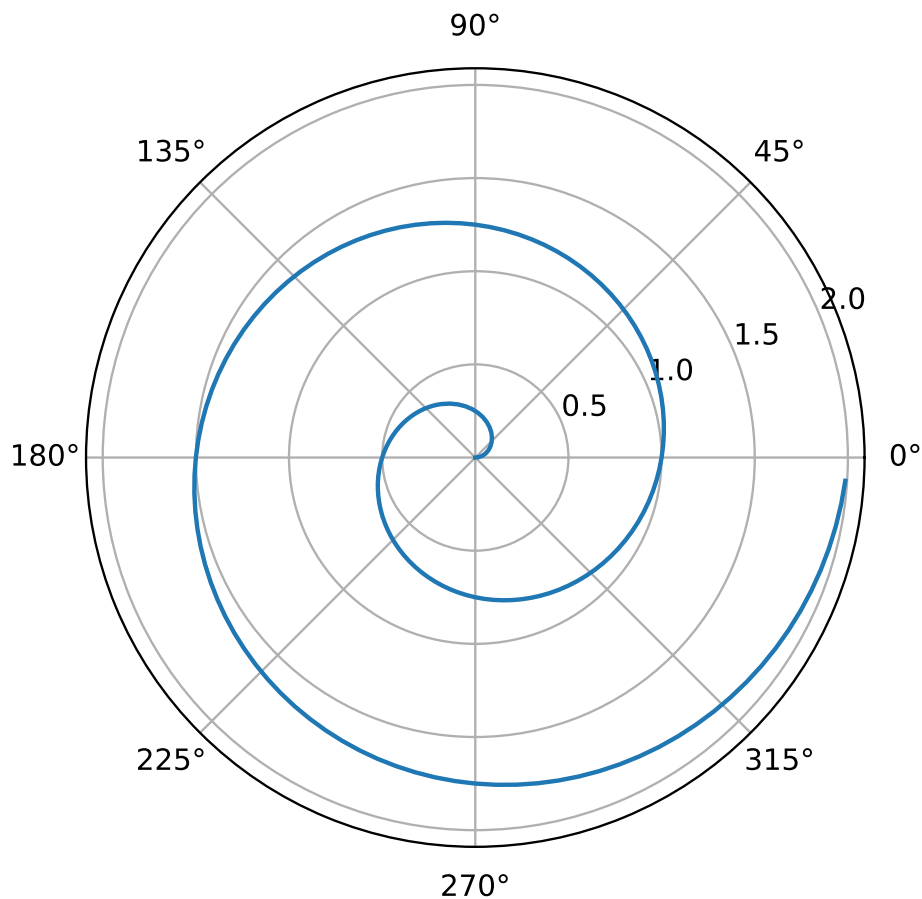
```
import numpy as np
import matplotlib.pyplot as plt

r = np.arange(0, 2, 0.01)
```

```

theta = 2 * np.pi * r
fig, ax = plt.subplots(
    subplot_kw = {'projection': 'polar'}
)
ax.plot(theta, r)
ax.set_rticks([0.5, 1, 1.5, 2])
ax.grid(True)
plt.show()

```



Multiple subplots:

```

python { .execute }
import numpy as np
import matplotlib.pyplot as plt

fig, axes = plt.subplots(2, 2, figsize=(10, 8))
x = np.linspace(0, 2 * np.pi, 100)

axes[0, 0].plot(x, np.sin(x))
axes[0, 0].set_title('Sine')

axes[0, 1].plot(x, np.cos(x))
axes[0, 1].set_title('Cosine')

axes[1, 0].plot(x, np.tan(x))

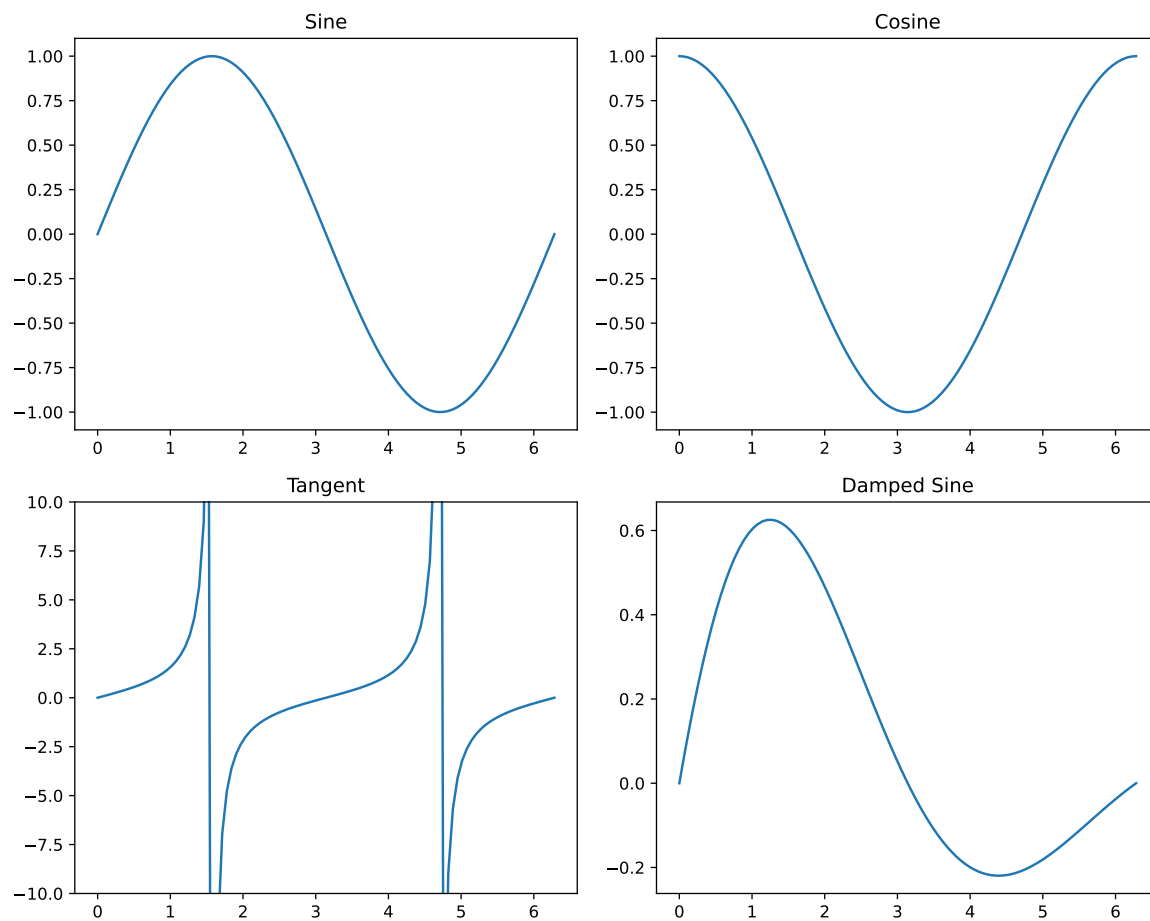

```



```
axes[1, 0].set_title('Tangent')
axes[1, 0].set_ylim(-10, 10)

axes[1, 1].plot(x, np.exp(-x/3) * np.sin(x))
axes[1, 1].set_title('Damped Sine')

plt.tight_layout()
plt.show()
```



## 2.20 Document Metadata (JSON)

Document metadata is configured in a separate JSON file:

```
{
  "title": "Document Title",
  "subtitle": "Course Name",
  "submittedto": "Professor Name",
  "university": "University Name",
  "department": "Department Name",
  "date": "January 1, 2024",
  "submittedby": [
    {
      "name": "Student Name",
      "roll": "Registration Number"
    }
  ]
}
```

```
    }  
  ],  
  "titleTemplate": 1,  
  "enableContentPage": false,  
  "enablePageCredits": false,  
  "moveFootnotesToEnd": false,  
  "enableThatsAllPage": true  
}
```

json

### 2.20.1 Title Template Modes

The `titleTemplate` setting controls how the title page is displayed:

- **0** : No title (disabled)
- **1** : Full university title page with logo (default) - Good for assignments and reports
- **2** : Title header above content - Good for notes
- **3** : Title on separate page - Good for when the contents are enabled

Controls document structure, metadata, and optional pages.