LaTeX Template

Lukas Maximilian Kapferer

lukas@kapferer.or.at
linkedin.com/in/lukas-kapferer
+43 677 61 44 27 44
A-6020 Innsbruck

Version from July 17, 2025

Contents

1 Tutorial

1.1 LaTeX Set-Up

1.2 Commands

```
\section{}
\subsection{}
\subsubsection{}

\section*{}
\subsection*{}
\subsubsection*{}
\subsubsubsection{}

\paragraph{}
\subparagraph{}
```

2 Features

2.1 Tables

Header 1	Header 2	Header 3
Row 1 Col 1	Row 1 Col 2	Row 1 Col 3

Table 1:

2.2 Shortcuts

2.3 Snippets

3 Code IDE

3.1 Java

Dies ist ein Java code.

```java
public class Stock {
    private String symbol;
    private double sharePrice;

    public Stock(String sym, double price) {
        this.symbol = sym;
        this.sharePrice = price;
    }

    public Stock(String sym) {
        this(sym, 0.0); // constructor chaining
    }
}
```

Listing 1: Eine verbesserte Darstellung der Java-Klasse ‘Stock‘.

3.2 Python

```python
    class DataAnalyzer:
    def __init__(self, data_points):
        self.data_points = data_points

    def calculate_average(self):
        """Calculates the average of the data points."""
        if not self.data_points:
            return 0
        return sum(self.data_points) / len(self.data_points)

# Example usage
analyzer = DataAnalyzer([10, 20, 30, 40, 50])
avg = analyzer.calculate_average()
print(f"The average is: {avg}")
```

Listing 2: Ein einfaches Beispiel für eine Python-Klasse.

öajsdföljasöl **Example:**

- TEstjölaskdfjalsjfasöld
- öasldkfjölasdkjföalkdsjföadlskfj
- asödfkjaölsdf

ölaksdjfölkölas
öajsdföljasöl **Example:**

1. TEstjölaskdfjalsjfasöld
2. öasldkfjölasdkjföalkdsjföadlskfj
3. asödfkjaölsdf

ölaksdjfölkölas

3.3 Servus "Obst"

3.3.1 Griasdi

3.3.1.1 Test

4 Test

Header1	Header2	Header3
Row1 Col1	Row1 Col2	Row1 Col3

Header1	Header2	Header3
Row1 Col1	Row1 Col2	Row1 Col3

Theorem 1

Lorem ipsum, ich dreh den Sack um.

Title

Lorem ipsum, ich dreh den Sack um.

Title

aslkdjfaölskdjföajdk

Title

xcvbnm,sdajfkasdf

Title

asjfölkjasdlf

```
import numpy as np
from scipy.stats import norm

def bs_call(S0, K, r, sigma, T):
    d1 = (np.log(S0/K) + (r + 0.5*sigma**2)*T)/(sigma*np.sqrt(T))
    d2 = d1 - sigma*np.sqrt(T)
    return S0*norm.cdf(d1) - K*np.exp(-r*T)*norm.cdf(d2)
```