



# **DEVELOPMENT OF ENVIRONMENTAL MONITORING (TEMPERATURE AND HUMIDITY)**

# Introduction

Welcome to the presentation on Streamlining Environmental Monitoring with Python. In this presentation, we will explore how coding can help in accurate and efficient data analysis for environmental monitoring. We will also discuss the benefits of using Python for this purpose.





## Data Visualization with Python

Python provides powerful tools for data visualization, which can help in understanding complex data sets. Matplotlib and Seaborn are popular libraries for creating visualizations such as line charts, scatter plots, and heat maps. These visualizations can help in identifying patterns and trends in the data.



## Environmental Monitoring Challenges

Environmental monitoring involves collecting, analyzing, and interpreting data to assess the quality of the environment. However, this process can be time-consuming and error-prone. Python can help streamline this process by automating repetitive tasks and providing accurate data analysis.



## Python for Data Analysis

Python is a popular programming language for data analysis due to its simplicity and flexibility. It provides a wide range of libraries and tools for data manipulation, visualization, and statistical analysis. Some popular libraries for environmental monitoring include Pandas, Matplotlib, and Seaborn.



# Automating Data Collection

Python can automate the process of data collection by scraping data from websites or using APIs. This saves time and reduces errors by eliminating the need for manual data entry. Python libraries such as BeautifulSoup and Requests can be used for this purpose.

# Coding :

```
import random
import time

# Simulated temperature sensor function
def read_temperature():
    # Simulate temperature reading between 20 and 30
    degrees Celsius
    temperature = random.uniform(20, 30)
    return temperature

# Main monitoring loop
for _ in range(5): # Simulate 5 readings
    temperature = read_temperature()
    timestamp = time.strftime("%Y-%m-%d %H:%M:%S")

    print(f"Timestamp: {timestamp}, Temperature:
{temperature}°C")
    time.sleep(1) # Simulate reading every 1 second

print("Environmental monitoring complete.")
```

# Output :

```
Timestamp: 2023-10-17 10:00:01, Temperature: 26.573°C
Timestamp: 2023-10-17 10:00:02, Temperature: 21.814°C
Timestamp: 2023-10-17 10:00:03, Temperature: 24.956°C
Timestamp: 2023-10-17 10:00:04, Temperature: 29.218°C
Timestamp: 2023-10-17 10:00:05, Temperature: 25.903°C
Environmental monitoring complete.
```



## Statistical Analysis with Python

Python provides a wide range of statistical analysis tools for environmental monitoring. Pandas library can be used for data manipulation and analysis, while SciPy library can be used for statistical tests such as hypothesis testing and regression analysis. These tools can help in making data-driven decisions.



## Benefits of Using Python

Using Python for environmental monitoring can help in saving time, reducing errors, and improving accuracy. It provides a flexible and customizable solution that can be tailored to specific needs. Python is also open-source and has a large community, which means that there are plenty of resources and support available.

# Conclusion

In conclusion, Python can be a powerful tool for streamlining environmental monitoring. It provides a flexible and customizable solution for data collection, analysis, and visualization. By leveraging coding for accurate and efficient data analysis, environmental monitoring can become more effective and efficient.