M23 Official Guide - PDF Content

Cover Page

Title: M23 – The Human-Friendly Data Science Language

Subtitle: Secure, Flexible, Case-Insensitive, and Typo-Tolerant

Author: MUHRIS A.J

Version: 1.0 **Date:** 2025

Table of Contents

- 1. Introduction
- 2. Installation & Setup
- 3. Language Philosophy
- 4. Variables & Pipelines
- 5. Core Commands
- 6. CSV & Tables
- 7. Charts & Visualization
- 8. Database Integration
- 9. API / Website Integration
- 10. Machine Learning
- 11. Error Handling & Typo Tolerance
- 12. Security & Sandbox Mode
- 13. Examples & Workflows
- 14. Mini Projects & Exercises
- 15. Recommended Learning Syllabus

1. Introduction

M23 is a human-readable, English-like programming language designed for:

- Data Science / Analytics
- Machine Learning workflows
- Database & API integration
- Secure, safe, typo-tolerant execution

It allows **flexible**, **case-insensitive commands**, enabling both **beginners and experts** to write code quickly.

2. Installation & Setup

Requirements:

- Python 3.11+
- Libraries: pandas, numpy, scikit-learn, matplotlib, requests

Install Libraries:

pip install pandas numpy scikit-learn matplotlib requests

Run an M23 Script:

python m23.py examples/csv_example.m23

3. Language Philosophy

- English-like syntax
- Case-insensitive commands
- Typo-tolerant with auto-suggestions
- Pipelines with -> operator
- Immutable by default, optional mutable variables
- Safe execution (sandbox, secure ML & DB access)

4. Variables & Pipelines

let age = 25 # immutable

mut revenue = 1000 # mutable

• Use -> to chain operations:

data -> select columns ["date" "revenue"] -> filter revenue > 1000

5. Core Commands

Command Description

load csv Load CSV file into variable

select columns Select specific columns

filter Filter rows

show Display dataframe (head N optional)

train model Train ML model (LinearRegression)

plot Create bar/line charts

save Save dataframe or model

sql Query database

fetch Get API / website data

run Execute external program (sandboxed)

6. CSV & Table Examples

LOAD CSV "sales.csv" AS sales

sales -> select columns ["date" "region" "revenue"]

sales -> filter revenue > 1000

show sales -> head 5

save sales as "filtered_sales.csv"

```
let table = create table [
    ["Name" "Age" "Salary"],
    ["Alice" 30 5000],
    ["Bob" 25 4000]
]
summary = table -> summarize avg_salary = mean(Salary)
show summary
```

7. Charts & Visualization

plot sales type "bar" x "region" y "revenue" title "Revenue by Region" plot sales type "line" x "date" y "revenue" title "Revenue Over Time"

- Supports bar and line charts.
- Future versions: interactive dashboards.

8. Database Integration

sql connect "postgres://user:password@localhost/salesdb" as db
sales_db = sql "SELECT date, region, SUM(amount) as total FROM transactions" on db
show sales_db -> head 10

- · Secure connection with sandboxed queries.
- SQL injection safe via parameterized queries.

9. API / Website Integration

```
data_api = fetch "https://api.example.com/sales" as json
table_api = data_api -> to table
show table_api -> head 5
save table_api as "api_sales.csv"
```

10. Machine Learning Example

LOAD CSV "house.csv" AS data

X = data -> select columns ["size" "bedrooms"]

y = data -> select columns ["price"]

train model HouseModel = LinearRegression on X y

metrics = HouseModel -> evaluate on X y

show metrics

save model HouseModel as "house_model.m23model"

11. Error Handling & Typo Tolerance

Example with typo:

loadd csv "sales.csv" as data # typo

Auto-suggestion:

ERROR: Unknown keyword 'loadd'

Did you mean 'load'? [auto-fix: yes/no]

Pipeline errors caught safely without crashing.

12. Security & Sandbox Mode

- Immutable variables by default
- Safe file access and database queries
- Optional encryption for saved models and CSVs
- No arbitrary system calls unless allowed

13. Full Example Workflow

LOAD CSV "sales.csv" AS sales

```
sales -> select columns ["date" "region" "revenue"]

sales -> filter revenue > 1000

report = sales -> group by region -> summarize total_revenue = sum(revenue)

plot report type "bar" x "region" y "total_revenue" title "Total Revenue by Region"

save report as "report.csv"
```

14. Mini Projects & Exercises

- 1. Analyze sales CSV and generate top 5 regions by revenue.
- 2. Train a LinearRegression model on housing data and evaluate.
- 3. Fetch API data, convert to table, and save CSV.
- 4. Create a dashboard showing revenue trends.

15. Recommended Learning Syllabus (8 Weeks)

Week Topics

- 1 Introduction, Variables, Pipelines
- 2 CSV & Table operations
- 3 Typo tolerance, case-insensitive commands
- 4 Charts & visualization
- 5 Machine Learning basics
- 6 Database & SQL integration
- 7 API / website data fetching
- 8 End-to-end projects & exercises

Instructions to Generate PDF

You can use Python fpdf or reportlab to create a PDF from this structured content:

from fpdf import FPDF

```
pdf = FPDF()
pdf.add_page()
pdf.set_font("Arial", "B", 16)
pdf.cell(0, 10, "M23 - Official Guide", 0, 1, "C")

pdf.set_font("Arial", "", 12)
pdf.multi_cell(0, 8, open("m23_guide_content.txt").read())

pdf.output("M23_Guide.pdf")
```

- Save the guide content in m23_guide_content.txt.
- Run the script to generate M23_Guide.pdf.

Result:

- Full **PDF-ready M23 Guide** with examples, exercises, and syllabus.
- Covers CSV, tables, charts, ML, database, API, sandbox execution.
- Ready to distribute as the official manual.