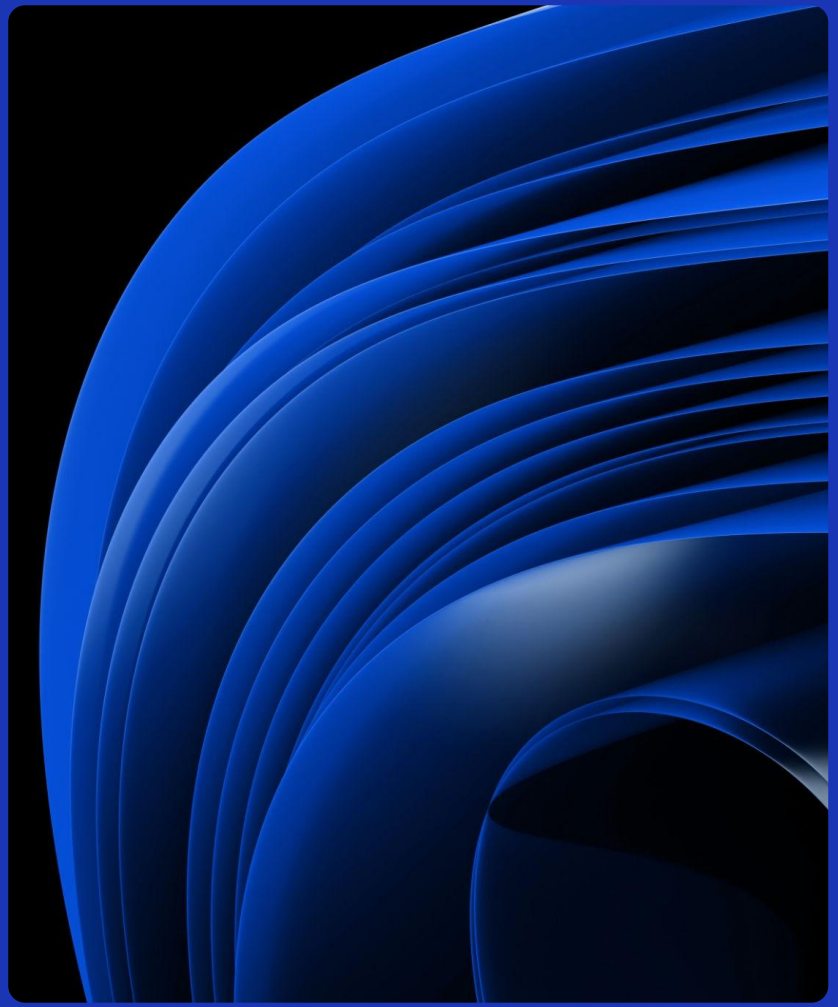
A My Path MN and BDPATCF Collaboration

Intro to Python

Led by William Munnich
Friday 3/22/2025



Functions

A block of code that can be reused over and over again.

```
def greet():
        print("Hello!")


greet()
```

```
greet(name):
        print("Hello", name)


greet(“James”)
```

```python
def greet_user(name, hour):
    if hour < 12:
        time_of_day = "morning"
    elif hour < 18:
        time_of_day = "afternoon"
    else:
        time_of_day = "evening"

    print(f"Good {time_of_day}, {name}!")


greet_user("Ava", 9)      # → Good morning, Ava!
greet_user("Liam", 15)    # → Good afternoon, Liam!
```

Referencing Functions

You can **reuse functions from another Python file** by importing them, like this:

```
from my_file import my_function
```

It’s just like referencing a function **someone else wrote** — which is exactly what happens when you import a **library** like math, csv, or random.

You’re using **pre-written code** to save time and avoid writing everything yourself!

File Types & File Extensions

Scripting	.py, .ps1, .sh
Simple Data Storage	.csv and .json
Audio	.mp3 and .mp4
Word Doc	.doc and .docx
Database	.db
Querying a Database	.sql
Markup	.html, .md

Databases vs Data Storage

Feature	Database	Data Storage File (CSV, JSON, etc.)
Structure	Organized in tables or documents (rows/fields)	Plain text formats (CSV = rows, JSON = objects)
Data Types	Enforces data types (e.g. INT, TEXT, DATE)	Mostly treated as text, type-checking is manual
Speed & Performance	Fast for large data, supports indexing	Slower, reads from start to end
Relationships	Supports links between data (foreign keys)	No built-in relationships between files
Search & Query	Powerful query languages (SQL, NoSQL queries)	Must be manually filtered or processed
Multi-user Access	Designed for many users at once	Not ideal for sharing or live access
Data Integrity & Rules	Can enforce rules (e.g., unique values)	No rules - users must check validity themselves
Real-world Use	Used in apps, games, websites, business systems	Used for exports, backups, configuration, sharing
Examples	MySQL, PostgreSQL, MongoDB, Firebase	.csv, .json, .xml, .txt

What Is a Database?

Key Features of a Database:

- **Stores lots of data** (names, links, numbers, etc.)
- **Keeps it organized** (tables, rows, columns)
- **Makes it easy to search and filter**
- **Can be used by apps, websites, and games**

Real-world Examples:

- Instagram uses databases to store user posts, comments, and likes.
- A video game stores player stats and scores in a database.
- Schools use databases to keep track of students and grades.

DaaS | Database as a Service

Feature	Firebase	AWS (RDS / DynamoDB)	Supabase	Azure (SQL / Cosmos DB)
🔧 Type	NoSQL (Firestore / Realtime DB)	SQL (RDS) + NoSQL (DynamoDB)	SQL (PostgreSQL)	SQL (Azure SQL) + NoSQL (Cosmos DB)
🧱 Structure	Documents & Collections (JSON)	Tables (SQL) or Key-Value	Tables (SQL, PostgreSQL)	Tables (SQL) or Documents/Graphs (NoSQL)
🚀 Best For	Real-time apps, mobile & chat	Scalable enterprise apps	Web apps, startups, full-stack dev	Enterprise apps, global-scale systems
🌍 APIs Automatically Generated?	✅ Firestore SDK / REST	❌ Manual setup	✅ REST + GraphQL auto-generated	❌ Manual setup via SDKs or Logic Apps
💻 Self-hosting Option	❌ No	✅ Yes (some DB engines)	✅ Yes (open-source version available)	❌ No (fully managed by Azure)
💰 Free Tier	✅ Generous	✅ Limited (depends on service)	✅ Generous	✅ Limited (Azure Free Tier available)
🎯 Developer Experience	Beginner-friendly, Google tools	More setup-heavy, powerful	Developer-first, open-source vibe	Strong IDE integration (Visual Studio), enterprise tools