Reading Text — Stage 03: Python Fundamentals

Python is a flexible, high-level programming language widely used in financial engineering due to its readability and extensive ecosystem. In this stage, you strengthen your foundation by mastering core syntax, data structures, and essential libraries.

Core Python Syntax & Control Flow

Functions encapsulate logic for reuse. Loops allow iteration, but in data-heavy finance, vectorization (via NumPy and pandas) is faster and more concise.

Decorators wrap functions to extend behavior — useful for logging, timing, or validation.

Data Structures

- **Lists**: ordered, mutable, general-purpose.
- **Dictionaries**: key-value mapping.
- **NumPy arrays**: homogeneous, fast math operations.
- **pandas Series/DataFrames**: tabular data handling with labels.

Choosing the right structure impacts performance and code clarity.

Vectorized Computation

Instead of looping over millions of rows, vectorized operations apply transformations across entire arrays/DataFrames at once.

Example:

```
arr * 2  # Vectorized
```

is far faster than:

```
[x * 2 for x in arr]
```

pandas Inspection Methods

Use `.info()` for dtypes and non-null counts, `.describe()` for quick stats, `.isna().sum()` for missing value checks.

Reusability

Move utility functions to `src/` early. This enforces clean, modular code and supports later pipeline stages.

Connection to Financial Engineering

These skills support data acquisition, preprocessing, and exploratory data analysis — foundational to risk models, portfolio optimization, and trading algorithms.
