# VARIABLES AND DATA TYPES

## PYTHON BASICS

*prepared by:*

**Gyro A. Madrona**
Electronics Engineer

# TOPIC OUTLINE

Python

Variables

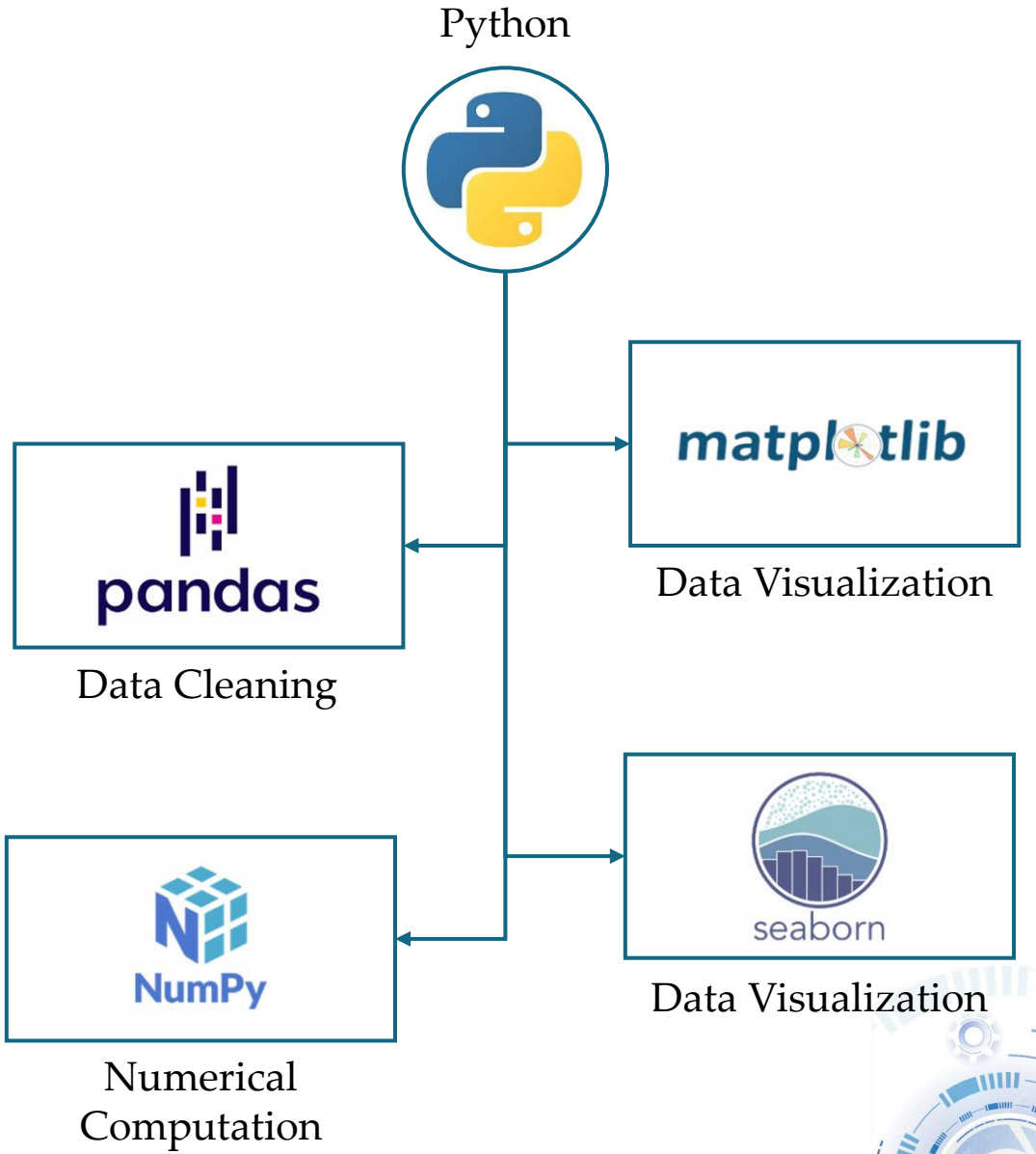
Data Types

Data Structures

# PYTHON

# PYTHON

**Python** is a powerful and versatile programming language widely used in data analytics due to its simplicity, readability, and **extensive library support**. It enables data analysts to perform various tasks, from data cleaning and manipulation to statistical analysis and visualization.

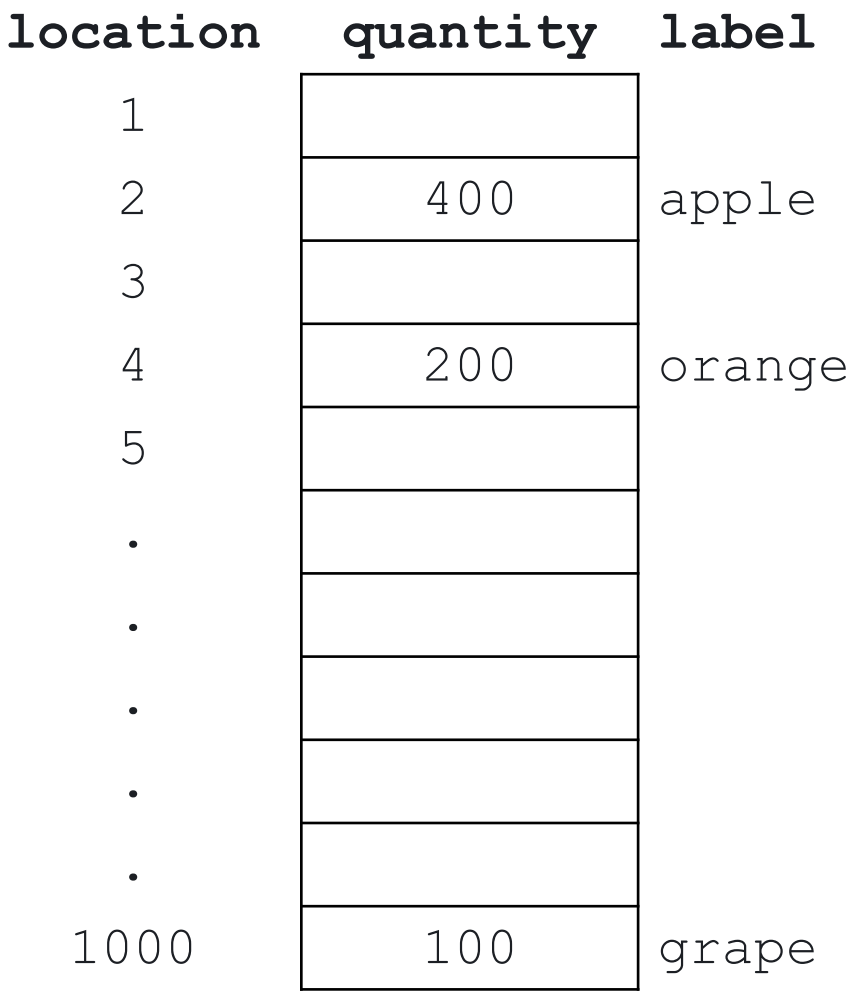



https://www.python.org/

# VARIABLES

# FRUIT CONTAINER ANALOGY

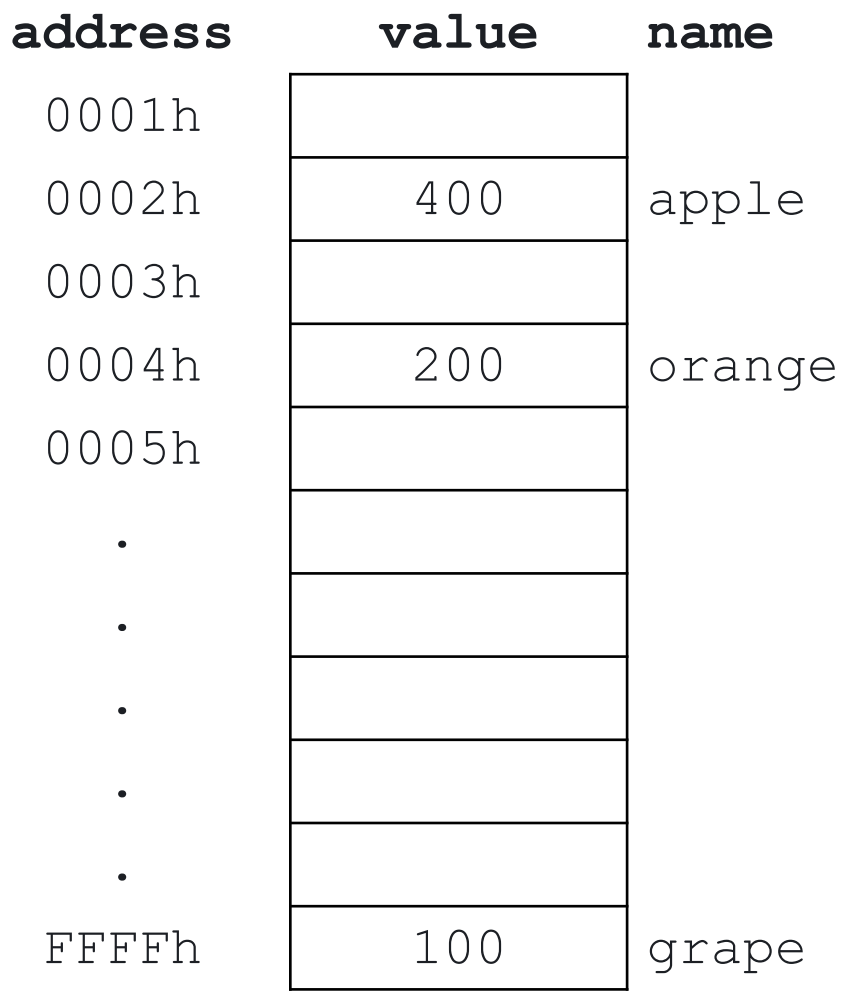
A **variable** is like a **storage container** that holds a specific type of fruit.



# VARIABLE

A **variable** is a named storage location in memory that **holds a value** of a specific data type.

1. **Data type** – Defines what kind of data it can store (e.g., `int`, `double`, `str`).
2. **Name (identifier)** – a unique name assigned to the variable.
3. **Value** – The actual data stored in memory.



# VARIABLE DECLARATION

Variable Declaration Syntax:

```
name = value
```

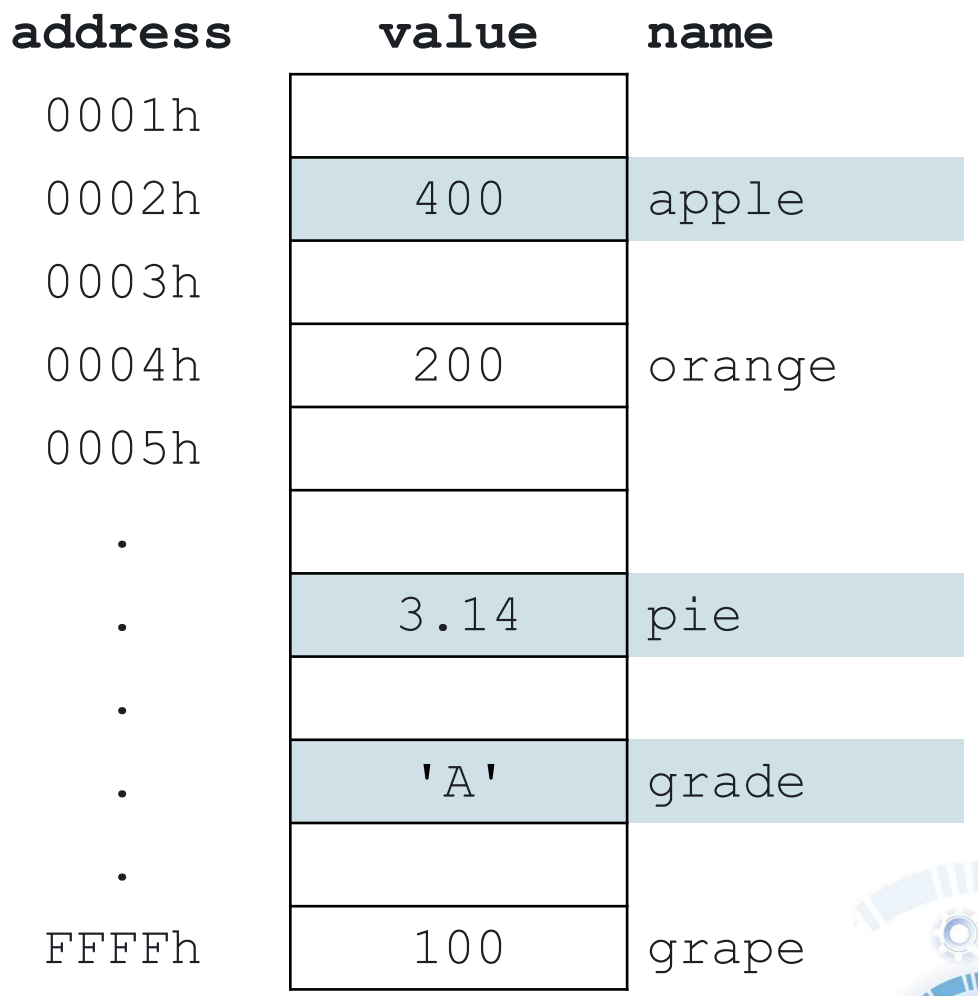
Example:

```
apple = 400

pie = 3.14

grade = 'A'
```

Python dynamically infers variable types based on assigned values without requiring explicit declarations.



# IDENTIFIERS

A variable is identified by a **unique name**, called an **identifier**.

- It can contain letters, digits, and underscores.
- It cannot have "space".
- It cannot start with a digit.
- It cannot be a reserved keyword (`int`, `return`, `class`).
- It is case-sensitive (e.g., `age` and `Age` are different variables)

Valid Identifiers:

// contains only letters
`age`

// starts with an underscore
`_salary`

// contains letters and a digit
`grade1`

// uses an underscore instead of space
`total_price`

# IDENTIFIERS

A variable is identified by a **unique name**, called an **identifier**.

- It can contain letters, digits, and underscores.
- It cannot have "space".
- It cannot start with a digit.
- It cannot be a reserved keyword (**`int`**, **`return`**, **`class`**).
- It is case-sensitive (e.g., **`age`** and **`Age`** are different variables)

Invalid Identifiers:

// starts with a digit
```
1stRank;
```

// uses a reserved keyword
```
class;
```

// contains a space
```
total price;
```

# DATA TYPES

# DATA TYPES

Python **data types** are the classification or categorization of data items. It represents the **kind of value** that tells what operations can be performed on a particular data.

# INTEGER

Example:

```
age = 25

total_students = 34
```

Integer `(int)` is used to store **integer values** (whole numbers).

# BOOLEAN

Example:

```
is_active = True

is_active = False
```

Boolean **(bool)** is used to store Boolean values (**True** or **False**).

# FLOATING POINT

Example:

```
pi = 3.14

height = 1.7
```

Floating point `(float)` is used to store floating-point numbers (decimal values).

# STRING

Example:

```
message = "Hello, World!"
message = 'Hello, World!'
```

**String** represent sequences of characters enclosed in **double quotes** or **single quote** .

# DATA STRUCTURES

# LIST

Examples:

```
fruits = ["apple", "orange", "grapes"]

my_list = [1,2,3, "Python", 5.5]
```

A `list` is an ordered and heterogeneous collection of elements, defined using square brackets `[ ]` .

# TUPLE

A `tuple` is an ordered, **immutable** , and heterogeneous collection of elements. It is defined using parentheses `( )` .

Examples:

```
fruits = ("apple", "orange", "grapes")
```

```
my_list = (1,2,3, "Python", 5.5)
```

# DICTIONARY

A **dictionary** collection of is an unordered and indexed **key-value** pairs. It is defined using curly braces `{ }` by colons `:` .

Examples:

```
employee = {"name" : "Ada",
            "position" : "analyst"}
```

```
employee = {"name" : "Ada",
            "age" : 25,
            "city" : "New York"}
```

# LABORATORY