Data Types

What Are Data Types?

At the core of any programming language lies the concept of data types a way for the language to understand what kind of value a variable can hold. Data types define not only the nature of the data but also the operations you can perform on it and the amount of memory it will consume. If you think of a variable as a box, then the data type is the label on that box telling the computer what's inside and how to handle it. In some languages (like C++ and Java), you must explicitly declare the data type before using a variable. In others (like Python and JavaScript), the type is determined automatically at runtime based on the assigned value.

Data Types in C++

C++ is a statically typed language, meaning you must specify the type of every variable at compile time. This ensures type safety and better performance, but it also means the compiler will throw errors if you try to assign incompatible types.

- **Primitive Types:**

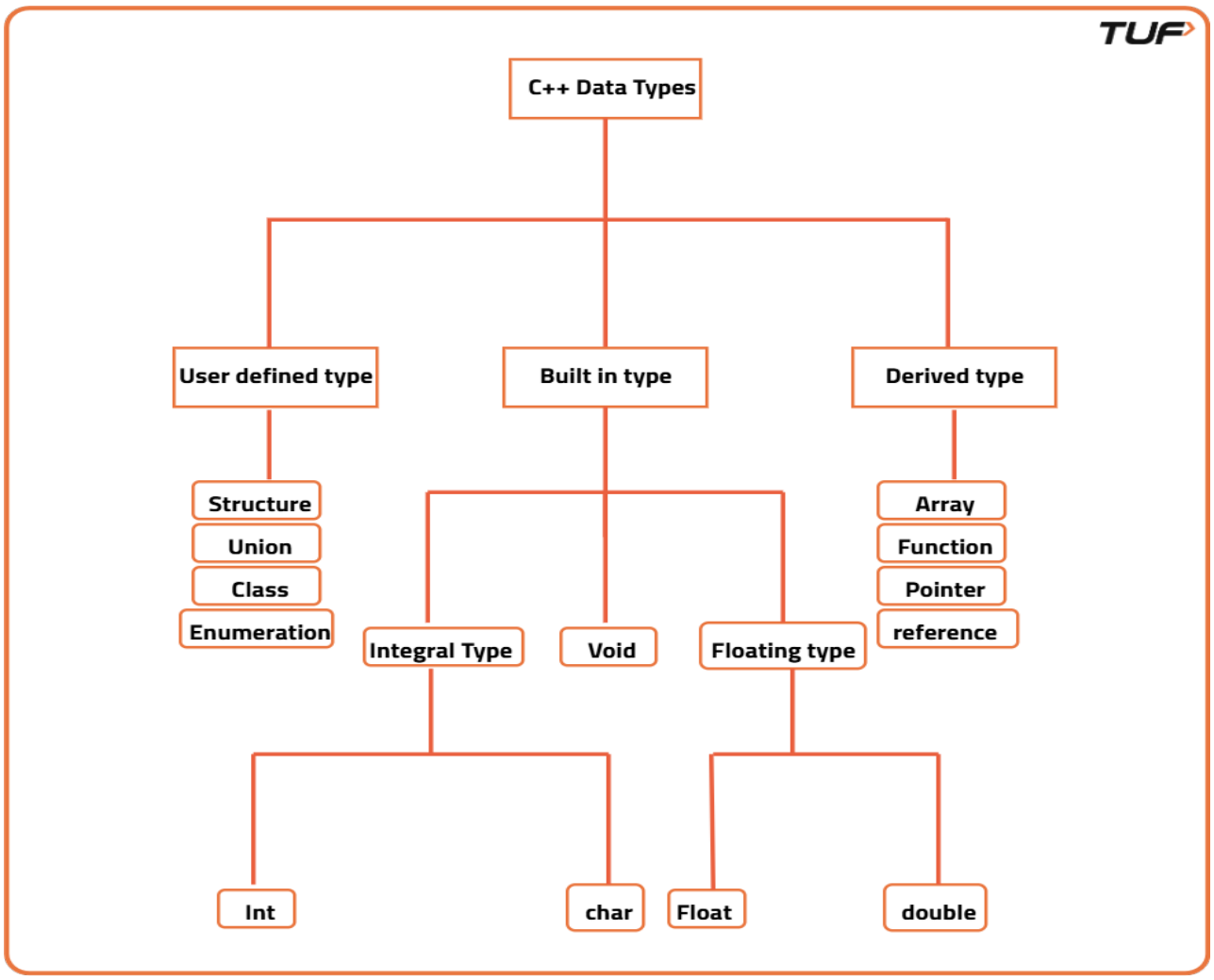
- int for integers (whole numbers).
- float and double for decimal values, with double providing more precision.
- char for single characters.
- bool for Boolean values (true or false).

- **Derived Types:**

Arrays, pointers, references, and function types fall under this.

- **User-defined Types:**

Structures (struct), classes (class), and enumerations (enum).



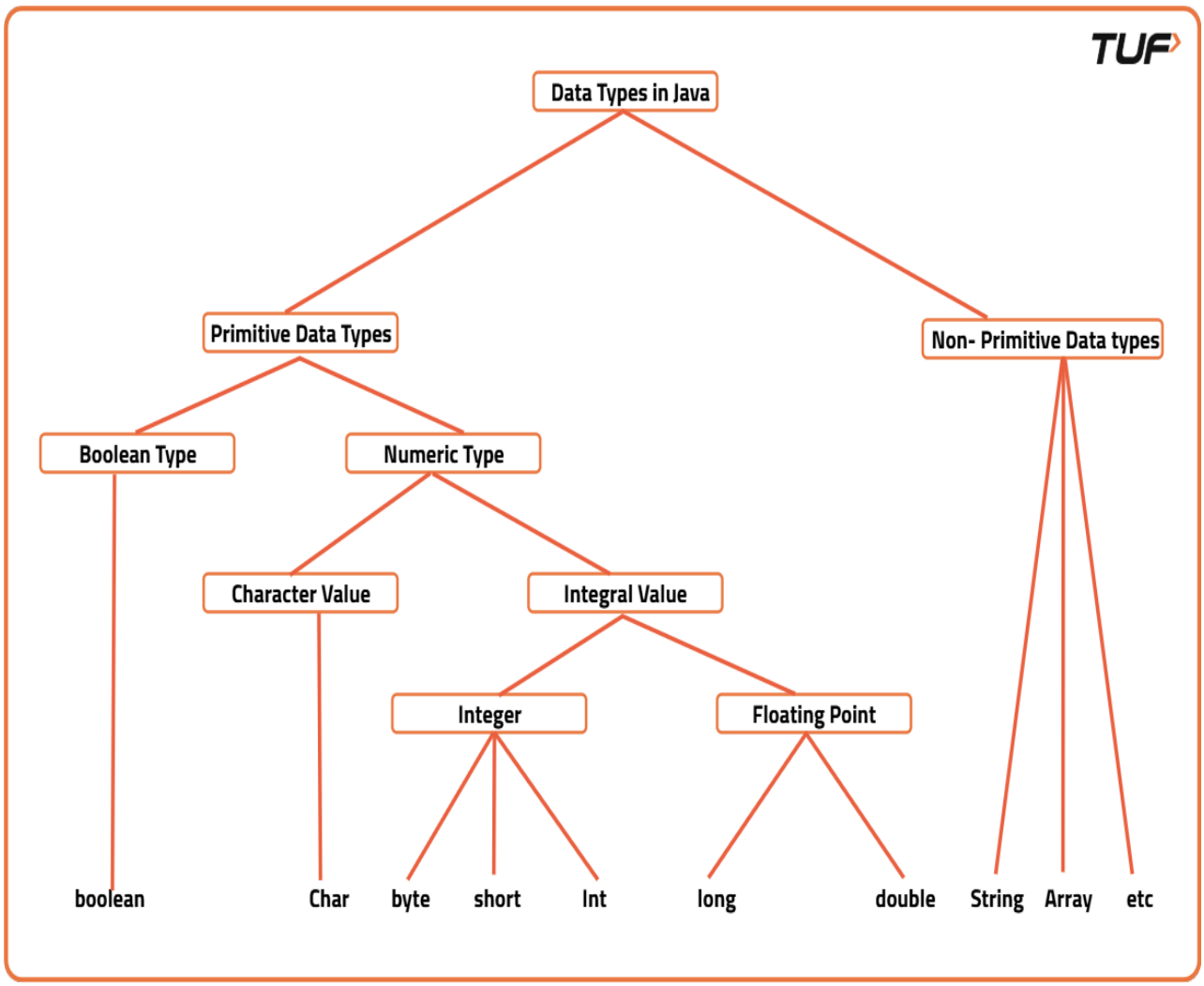
C++ also supports modifiers like unsigned, short, and long to tweak the size and range of numeric types. For example, unsigned int only stores non-negative integers but allows larger maximum values than a signed int.

Data Types in Java

Java is also statically typed, but unlike C++, it runs on the JVM (Java Virtual Machine) and offers automatic memory management via garbage collection. Data types in Java are split into two broad categories:

- **Primitive Types (fixed in size and stored directly in memory):**
  - byte, short, int, long for integers of increasing size.
  - float, double for floating-point numbers.
  - char for Unicode characters.
  - boolean for logical true/false values.
- **Non-Primitive (Reference) Types:**

  These include objects like String, arrays, user-defined classes, and interfaces. A reference type variable holds a memory address pointing to the actual object in the heap.

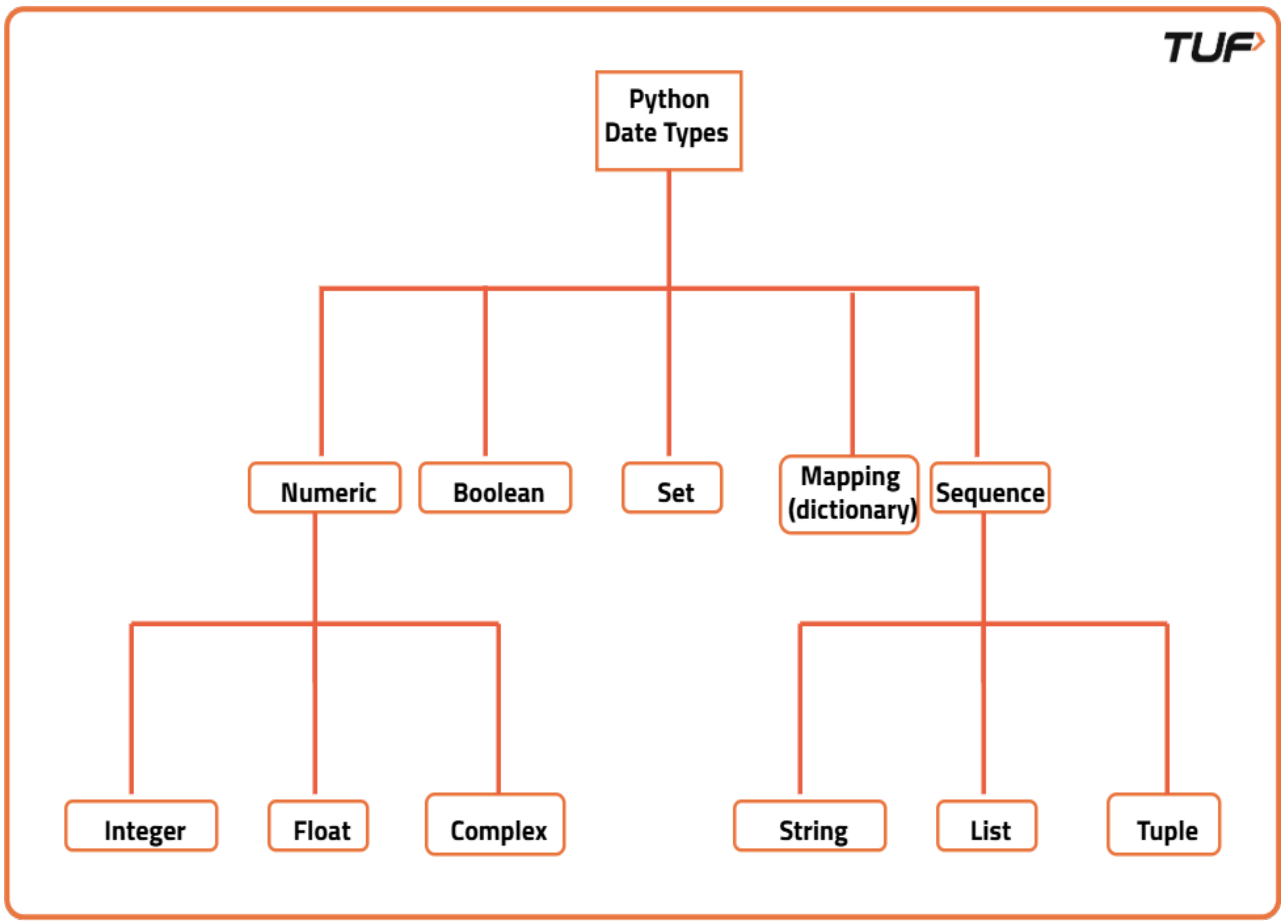


One key difference from C++ is that Java's boolean type can only be true or false you can't treat it as a number.

Data Types in Python

Python is dynamically typed, meaning you don't need to declare the data type the interpreter figures it out at runtime. This makes coding faster but can also lead to subtle bugs if you're not careful.

- Numeric Types:
  - int for integers (unlimited size).
  - float for decimal numbers (double precision).
  - complex for complex numbers (3+5j).
- Sequence Types: list, tuple, and range for ordered collections.
- Text Type: str for strings.
- Mapping Type: dict for key-value pairs.
- Set Types: set and frozenset for unordered collections of unique elements.
- Boolean Type: bool for logical operations.

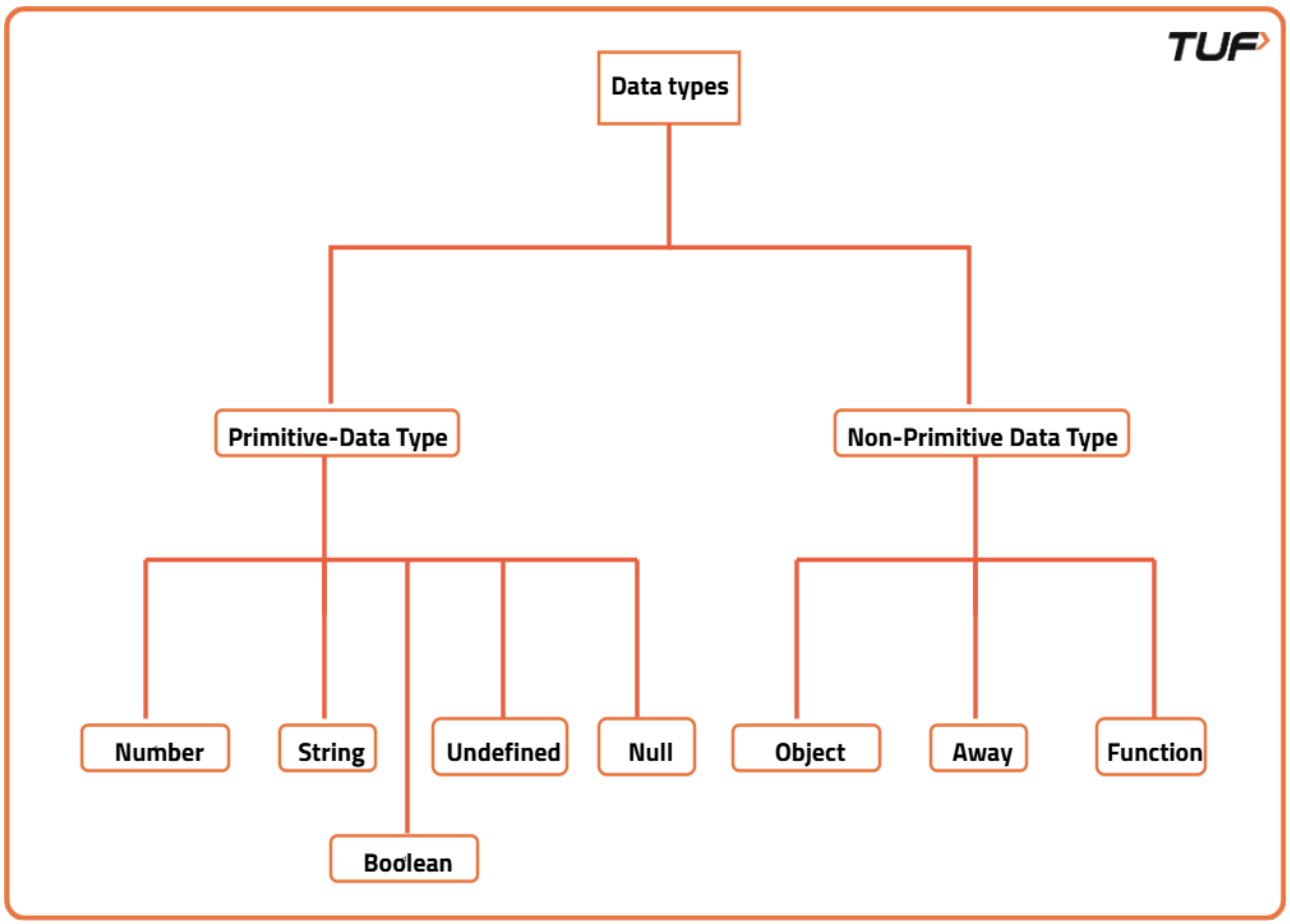


Python is flexible you can assign an integer to a variable, then later assign a string to the same variable without errors. But this flexibility means you have to be more cautious with type-related logic.

Data Types in JavaScript

JavaScript is also dynamically typed but is often considered more “loose” than Python when it comes to type conversions. It has fewer built-in types, but its type coercion rules can cause unexpected behavior if you’re not careful.

- Primitive Types:
    - number for both integers and floating-point numbers.
    - number for both integers and floating-point numbers.
    - string for text.
    - boolean for true/false.
    - undefined for variables declared but not assigned a value.
    - null for intentional absence of a value.
    - symbol for unique identifiers.
- Objects: Arrays, functions, and all other non-primitive structures in JS are objects.



One important quirk: In JavaScript, typeof null returns "object", which is a long-standing bug from the early days of the language.

Why Data Types Matter

Regardless of the language, understanding data types is crucial for writing efficient, bug-free programs. In statically typed languages like C++ and Java, they help catch errors before running the code and allow for optimizations at compile time. In dynamically typed languages like Python and JavaScript, they offer flexibility but require extra care to avoid type mismatch errors. Data types also determine how much memory your variables use, how they’re stored, and what operations can be performed on them.

L	Numeric Types	Text / Char	Boolean	Collections (Core)	Special / Null	User-Defined Types	Notes
C++	short, int, long, long long, float, double, long double (+ unsigned mods)	char, wchar_t, char16_t, char32_t, std::string	bool	std::vector, std::array, std::list, std::deque, std::set, std::map, std::unordered_map	nullptr	struct, class, enum, union, templates	Static typing; sizes depend on platform; RAII & value semantics are common.
Java	byte, short, int, long, float, double	char (UTF-16), String	boolean	ArrayList, LinkedList, HashSet, HashMap, TreeMap, arrays	null	Classes, interfaces, enums, records	Static typing; primitives vs wrappers (int vs Integer); String immuta ble.
Python	int (arbitrary precision), float, complex	str	bool	list, tuple, dict, set, frozenset, range	None	Classes (everything is an object), dataclasses, enums, typing hints	Dynamic typing; immutable vs mutable matters (tuple vs list); integers unbounded.
JavaSc ript	number (IEEE-754), bigint	string	boolean	Array, plain objects ({}), Map, Set, WeakMap, WeakSet, TypedArray	null, undefined, NaN	Objects, classes (syntax over prototypes), symbols	Dynamic typing; type coercion; typeof null === "object" quirk; numbers are double-precision by default.