

# 1. Introduction: From Python to C

## 1.1 Key Differences Overview

Aspect	Python	C
<b>Compilation</b>	Interpreted	Compiled
<b>Type System</b>	Dynamic typing	Static typing
<b>Memory Management</b>	Automatic	Manual
<b>Syntax Style</b>	Indentation-based	Brace-based
<b>Performance</b>	Slower execution	Faster execution
<b>Development Speed</b>	Faster to write	More verbose

Some of you might ask, What does it mean by `Static Typing` vs `Dynamic Typing` ?

### Dynamic Typing ( Python ):

- Variables can change their data type during program execution
- Type checking happens at runtime
- No need to declare variable types explicitly
- More flexible but can lead to runtime errors

```
# Python - Dynamic Typing
x = 5          # x is an integer
x = "Hello"   # Now x is a string (allowed!)
x = 3.14      # Now x is a float (also allowed!)
```

### Static Typing ( C ):

- Variables must be declared with a specific data type
- Type checking happens at compile time
- Once declared, a variable cannot change its type
- Less flexible but catches type errors before program runs

```
// C - Static Typing
int x = 5;          // x is declared as integer
```

```
x = "Hello";           // ERROR! Cannot assign string to integer variable
float y = 3.14f;      // y must be declared as float to store decimal numbers
```

### Advantages of Static Typing ( C ):

- Errors caught during compilation, not during execution
- Better performance (no runtime type checking needed)
- More predictable memory usage
- Clearer code documentation (types are explicit)

### Advantages of Dynamic Typing ( Python ):

- Faster prototyping and development
- More flexible for rapid changes
- Less verbose code
- Easier for beginners to learn

## 1.2 Basic Program Structure

### Python:

```
# Simple Python program
print("Hello, World!")
```

### C:

```
#include <stdio.h>

int main() {
    printf("Hello, World!\n");
    return 0;
}
```

### Key Points:

- C requires explicit inclusion of header files (such as `#include <stdio.h>`)
- Every C program must have a `main()` function
- C statements end with semicolons (`;`)
- C uses curly braces `{}` to define code blocks
- Functions must explicitly return a value except for void function (we will learn more about this on the next module)