

C programming language Vs Python programming language:-

Saving files in C programming language and Python programming language is simple, but the **file extensions matter a lot.**

□ 1. Saving a C File

□ Steps:

1. Open any editor:
 - Notepad
 - VS Code
2. Write your C code:

```
#include <stdio.h>
int main() {
    printf("Hello World");
    return 0;
}
```

3. Click **File** → **Save As**
4. Save with **.c extension**

```
hello.c
```

5. Important:
 - Choose "**All Files**" (not .txt)
 - Encoding: UTF-8 (default is fine)

Compile the C file

Use a compiler like GCC:

```
gcc hello.c -o hello
```

□ This creates an executable file:

- Windows → hello.exe
- Mac/Linux → hello

□ Run the c Program

Windows:

hello
Or
Hello.exe

As shown below example:-

```
C:\Users\Big Data\Desktop>set path=C:\MinGW\bin
C:\Users\Big Data\Desktop>gcc hello.c -o hello
C:\Users\Big Data\Desktop>hello
Hello World
C:\Users\Big Data\Desktop>
```

□ 2. Saving a Python File

□ Steps:

1. Open editor:
 - Notepad
 - VS Code
2. Write Python code:

```
print("Hello Python")
```

3. Click **File** → **Save As**
4. Save with **.py extension**

```
hello.py
```

Run program

Open terminal in file location:

```
python hello.py
```

or sometimes:

```
python3 hello.py
```

"Assume the file hello.py is saved in the Python demo folder."

```
(env) D:\>cd python demo
(env) D:\python demo>python hello.py
Hello Python
(env) D:\python demo>_
```

3. Key Difference

Feature	C File	Python File
Extension	.c	.py
Compilation	Required	Not required
Example	hello.c	hello.py

Quick Comparison Table

Step	C	Python
Install	GCC compiler	Python interpreter
File type	.c	.py
Run process	Compile → Run	Direct run
Command	<code>gcc file.c -o file</code>	<code>python file.py</code>

C programming language vs Python programming language.

I'll keep it simple and directly side-by-side so you can *see the difference clearly*.

□ 1. Declaring Data Types

C (you must declare type explicitly)

```
int a = 10;
float b = 5.5;
char c = 'A';
```

Python (no type declaration needed)

```
a = 10
b = 5.5
c = 'A'
```

□ Key Difference:

- C = *static typing* (type is fixed)
 - Python = *dynamic typing* (type decided at runtime)
-

□ 2. If-Else Statement

C

```
int a = 10;

if (a > 5) {
    printf("Greater");
} else {
    printf("Smaller");
}
```

Python

```
a = 10

if a > 5:
    print("Greater")
else:
    print("Smaller")
```

□ Key Difference:

- C uses () and {}

- Python uses : and indentation
-

□ 3. For Loop

C

```
for(int i = 0; i < 5; i++) {  
    printf("%d\n", i);  
}
```

Python

```
for i in range(5):  
    print(i)
```

□ Key Difference:

- C = full control (init, condition, increment)
 - Python = simpler, uses range ()
-

□ 4. Function Example

C

```
#include <stdio.h>  
  
int add(int a, int b) {  
    return a + b;  
}  
  
int main() {  
    int result = add(3, 4);  
    printf("%d", result);  
    return 0;  
}
```

Python

```
def add(a, b):  
    return a + b  
  
result = add(3, 4)  
print(result)
```

□ Key Difference:

- C requires:

- return type (int)
 - main() function
 - Python:
 - no type declaration
 - no mandatory main function
-

☐ Quick Summary Table

Concept	C	Python
Data Types	Must declare	No declaration
If-Else	(), {}	;, indentation
For Loop	Manual control	range () simplified
Function	Typed + main() required	Simple, flexible