Subjective Programming Introduction

Table of Contents

License notice

Introduction

This is a subjectively introductory guide to programming in Python and C.

You can view/download a PDF version of this guide.

The chosen programming languages

In this guide, we'll be using Python and C.

This is so you can both learn the easy way of programming and the hard way, and get a balanced learning experience.

It'll be a bit of a learning experience for me, too.

Stuff that you need

- A computer
- An operating system
- A terminal/command prompt
- A text editor
- A C compiler
- A Python 3 installation

Install Fedora and you'll get all of that in one package! :D

Otherwise, you can work with what you've got.

Computer

If you're reading this on a phone, get a fricking computer. Programming is hell on a mobile device, especially C programming.

Any computer will work, as long as it supports Python 3.

Operating system

If you're using Windows, you should use at least Windows 10.

If you're a Mac user, the latest version of macOS is recommended.

If you're on Linux, just make sure your packages are up to date.

Terminal or command prompt

On Windows 11, it's Windows Terminal.

On Windows 10 or earlier, it's PowerShell. (Don't use Command Prompt.)

On macOS, it's Terminal.

On Linux, just search for "terminal" (or use the TTY if you're epic)

Text editor

The most popular choice for editing code across any OS is Visual Studio Code. It's a super solid pick.

You can use Notepad if you're on Windows, but you should get a dedicated text editor with syntax highlighting.

macOS has TextEdit, but that's really more of a rich text editor.

On Linux, you're spoilt for choice when it comes to text editors. VS Code, as usual, is a good pick, but feel free to browse the web and find an editor that speaks to you. For editing at the terminal, I highly recommend Micro.

C compiler

On Linux, you should install `base-devel` or `build-essential` or whatever it's called.

On Windows, install MinGW-w64.

On macOS, try running `gcc` in Terminal. You should be prompted to install the Xcode development tools.

Python 3 installation

On Windows or macOS, install Python. On Linux, you should already have Python.

Run `python3 -V` (`python -V` on Windows) in your terminal to check it's there.

Got all that?

Great! Let's begin.

A simple example

Printing "Hello World"

Python

```
print("Hello World!")
```

The Python example is very simple. Python's a simple language.

C

```
#include <stdio.h>
```

```
int main() {
    printf("Hello World!");
}
```

The C example looks much more complex! This is what programming in *most* languages is like. It's not that C is too hard, it's that Python set your expectations for it to be easy, so comparatively, it's a big code block.

But this isn't a psychology guide, this is programming. Let's keep going.

Python can look complex, too

```
def main():
    print("Hello World!")
if __name__ == "__main__":
    main()
```

Ooh! Look! Now the Python example is 4 lines long.

What's going on here, is that we're defining a `main` function, called the main entry point.

You don't have to define `main` in Python, because it's been done for you.

In C, though, you do have to do it yourself.

Data types

In programming, you have the concept of **data types**.

Data types, are, obviously enough, types of data.

C has `int`, `float`, `double`, `long`, `short` and `char`.

Python has `int`, `float`,, `bool`, `str`, `list`, `tuple` and `dict`.

We'll keep this simple and use the common types, `int`, `float`/`double` and `str`/`char`.

Representing whole numbers

You can represent whole numbers with `int`, short for integer, which means whole number.

Python

```
x = 1 # is int! Python's types are implicit.
```

C

```
#include <stdio.h>
int main() {
    int x = 1; // C's types are explicit, so you have to declare int.
}
```

It's just a little bit different. As mentioned in the comments, Python has **implicit types**, so you don't need to declare the type. In fact, if you do, it errors out.

```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> int x = 1
SyntaxError: invalid syntax
>>>
```

Whereas if you don't declare the type in C, the compiler will complain that the variable doesn't exist.

```
$ ed implicit.c
implicit.c: No such file or directory
i
#include <stdio.h>
int main() {
    x = 1
}
.
w
45
$ gcc implicit.c
implicit.c: In function 'main':
implicit.c:3:5: error: 'x' undeclared (first use in this function)
    3 |     x = 1;
      |     ^
implicit.c:3:5: note: each undeclared identifier is reported only once for each
function it appears in
$
```

Don't mind the use of ed. Use whatever text editor you want.

> **NOTE** This guide is unfinished and unfortunately ends here. Come back later maybe?