Installing an Editor

Writing and editing code requires a text editor. We will focus on and provide support for two free, multi-platform editors: Atom and VS Code. If you already have a favorite text editor that you are comfortable with, feel free to use it.

Atom—an Extensible Text Editor

Atom is a modern and customizable text editor that supports a wide range of programming languages on multiple platforms. It can be downloaded from atom.io.

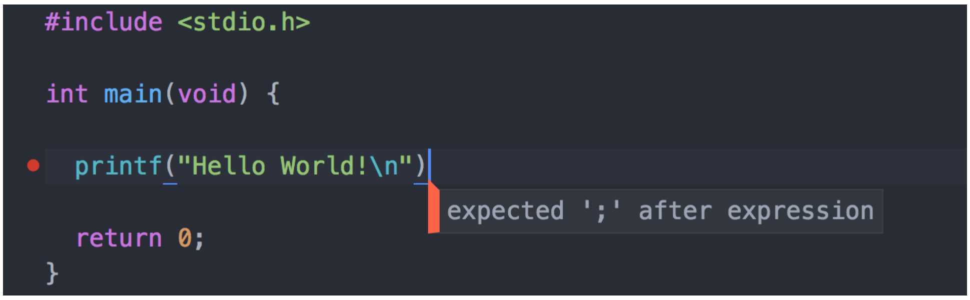
After installing Atom, you can install and manage extensions by opening the Settings view (keyboard shortcut: `ctrl-,` in Windows/Linux and `cmd-,` in macOS). Click on the "Install" tab which brings up a search bar and some featured packages.

Check for errors and suspicious language usage

If you would like Atom to analyze and check your source code for errors or suspicious language usage, you can install a so-called *linter* package. The term comes from lint which was the name of a Unix utility from the late 1970s for detecting a number of bugs and obscurities (derived from the name. The following packages provide a linter for C/C++:

- `linter`
- `linter-gcc` (GCC users) or `linter-clang` (Clang users)

After installing the appropriate linter packages, Atom will mark warnings and errors with orange and red dots, respectively, as illustrated in the screenshot below.



```
#include <stdio.h>

int main(void) {

  printf("Hello World!\n")
                          expected ';' after expression
  return 0;
}
```

Note that you may need to configure the package settings to make sure that it can find your compiler.

Compile your code directly in Atom

The package `gcc-make-run` allows you to compile and run self-contained C programs (i.e., consisting of a single source file) using `f6` as a shortcut, and compiler/linker options can be configured via `ctrl-f6` (Windows/Linux) or `cmd-f6` (macOS). The package works with both GCC and Clang, and it can also handle so-called "makefiles", a concept we will return to shortly.

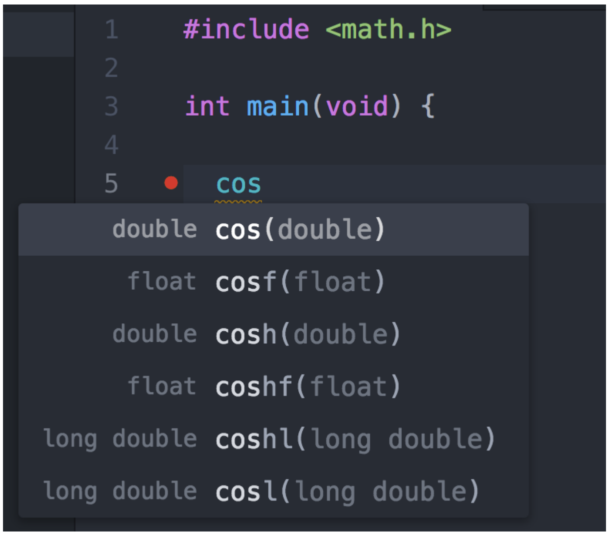
Another option is the `platformio-ide-terminal` package which adds an embedded terminal panel to Atom.

Note that on Windows, the default terminal provided by `platformio-ide-terminal` is *PowerShell*. You can change this in the package settings:

- Write `C:\Windows\System32\cmd.exe` in “Shell Override” if you prefer to use Command Prompt; add `/C bash` to “Shell Arguments” if you prefer a Bash shell.
- Write `ubuntu.exe` in “Shell Override” if you want to use Ubuntu installed via the Windows Subsystem for Linux.

Autocompletion

The package `autocomplete-clang` provides completions by Clang. For example, if you include the `math.h` header file, the Atom package will suggest possible completions if you type `cos`, as shown in the screenshot below.



```
#include <math.h>

int main(void) {

    cos
```

Formatting and automatic indentation

The `atom-beautify` package helps you format and indent your source files to improve readability.

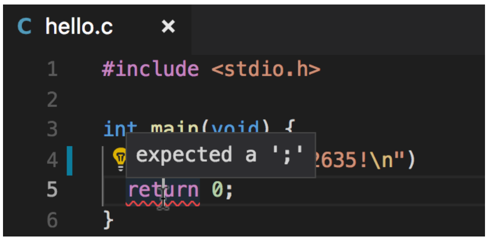
Syntax highlighting in Makefiles

The `language-make` package adds syntax highlighting to Make-files in Atom.

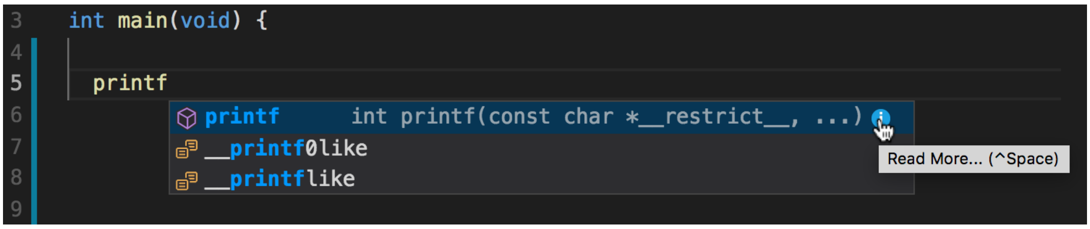
Visual Studio Code

An alternative to Atom is Microsoft’s Visual Studio Code which, like Atom, is a customizable, cross-platform editor that supports many programming languages. You can download it from code.visualstudio.com.

Support for C/C++ is available as an extension that can be installed by clicking on *Extensions* in the *View* menu. Search for “C/C++” which will shows up as the first search result, and click on the install button to install it. This extension provides a number of useful features such as syntax highlighting, error checking, and auto-completion, as shown in the screenshots below.

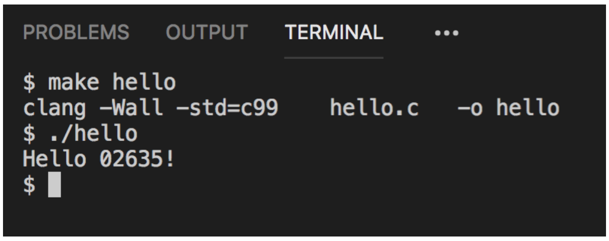


```
#include <stdio.h>

int main(void) {
  expected a ';' 2635!\n")
  return 0;
}
```



```
int main(void) {

  printf
        printf        int printf(const char *__restrict__, ...)
        __printf0like
        __printflike
```

Another useful feature of Visual Studio Code is that it has an integrated terminal. To launch the terminal, click on *Integrated Terminal* in the *View* menu.



```
$ make hello
clang -Wall -std=c99    hello.c   -o hello
$ ./hello
Hello 02635!
$ 
```

Compile your code directly in Visual Studio Code

The “C/C++ Compile Run” extension allows you to compile and run simple C programs directly from the editor. After installing this extension, you can use `f6` (Windows/Linux) or `cmd-R` (macOS) to compile your code and run your program in an embedded terminal panel.