

# VS CODE

# Download และ Install VS Code

<https://code.visualstudio.com/>

The screenshot shows the official Visual Studio Code website. At the top, there's a navigation bar with links to 'Docs', 'Updates', 'Blog', 'API', 'Extensions', 'FAQ', and 'Learn'. A prominent blue button on the right says 'Download'. Below the navigation, a message announces 'Version 1.69 is now available! Read about the new features and fixes from June.' On the left, there's a large promotional image with the text 'Code editing. Redefined.' and 'Free. Built on open source. Runs everywhere.' Below this, a blue button says 'Download for Windows' with 'Stable Build' underneath. To the right, a code editor window is displayed, showing a file named 'serviceWorker.js' with some JavaScript code. The code includes imports for 'index.js' and 'serviceWorker.js', and defines a service worker registration function. The bottom of the code editor has a terminal tab and a status bar indicating 'You can now view create-react-app in the browser.'



Setup - Microsoft Visual Studio Code (User)



## License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

*This license applies to the Visual Studio Code product. Source Code for Visual Studio Code is available at <https://github.com/Microsoft/vscode> under the MIT license agreement at <https://github.com/microsoft/vscode/blob/main/LICENSE.txt>. Additional license information can be found in our FAQ at <https://code.visualstudio.com/docs/supporting/faq>.*

## MICROSOFT SOFTWARE LICENSE TERMS

### MICROSOFT VISUAL STUDIO CODE

These license terms are an agreement between you and Microsoft Corporation (or based

- I accept the agreement
- I do not accept the agreement

Next >

Cancel

[Overview](#)[SETUP](#)[GET STARTED](#)[USER GUIDE](#)[LANGUAGES](#)[NODEJS / JAVASCRIPT](#)[TYPESCRIPT](#)[PYTHON](#)[JAVA](#)[C++](#)[CONTAINERS](#)

# Getting Started

Visual Studio Code

available for

Node.js and

Python, PHP

Visual Studio

## Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.

Additional icons:

Create a desktop icon

Other:

Add "Open with Code" action to Windows Explorer file context menu

Add "Open with Code" action to Windows Explorer directory context menu

Register Code as an editor for supported file types

Add to PATH (requires shell restart)

&lt; Back

Next &gt;

Cancel



## GETTING STARTED

[VS Code in Action](#)[Top Extensions](#)[First Steps](#)[Keyboard Shortcuts](#)[Downloads](#)[Privacy](#)[Tweet this link](#)[Subscribe](#)[Ask questions](#)[Follow @code](#)[Request features](#)[Report issues](#)[Watch videos](#)

# TDM-GCC Complier



# tdm-gcc

GCC compiler, Windows-friendly.

[home](#) / [about](#) / [download](#) / [donate](#) / [archive](#)

The latest release is based on **GCC 10.3.0**.

MinGW-w64 based

[tdm64-gcc-10.3.0-2.exe](#), 76.6 MB

MinGW.org based

[tdm-gcc-10.3.0.exe](#), 60.2 MB

24 MAY 2021 • RELEASE / NEW SERIES / CHangelog



tdm-gcc

GCC compiler, Windows-friendly.

[home](#) / [about](#) / [download](#) / [donate](#) / [archive](#)

The latest release is based on GCC 10.3.0.

MinGW-w64 based  
tdm-gcc-10.3.0-2.exe, 76.6 MB

MinGW.org based  
tdm-gcc-10.3.0.exe, 60.2 MB

TDM-GCC Setup

#### Wizard Action

Choose which action you want the setup wizard to perform.

#### Create

: Create a new TDM-GCC installation

#### Manage

: Manage an existing TDM-GCC installation

#### Remove

: Remove a TDM-GCC installation

Check for updated files on the TDM-GCC server

TDM-GCC Setup 1.2005.1

I'm proud  
a Windows/Windows Server 2008 R2 was soon up and running after an employee change go

Keep reading for download links and change notes.

(Read more...)

TDM-GCC Setup

#### Wizard Action

Choose which action you want the setup wizard to perform.

#### Create

: Create a new TDM-GCC installation

#### Manage

: Manage an existing TDM-GCC installation

#### Remove

: Remove a TDM-GCC installation

TDM-GCC Setup

#### New Installation: Installation Directory

Choose the installation directory to use.

Setup will install TDM-GCC in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

#### Installation Directory

C:\TDM-GCC\64

[Browse...](#)

Space available: 74.2 GB

TDM-GCC Setup 1.2005.1

< Back

[Next >](#)

[Cancel](#)



TDM-GCC Setup

#### Select Edition

Choose which edition of TDM-GCC you want to install.

#### MinGW/TDM (32-bit)

Create a MinGW-based installation

#### MinGW-w64/TDM64 (32-bit and 64-bit)

Create a MinGW-w64-based installation

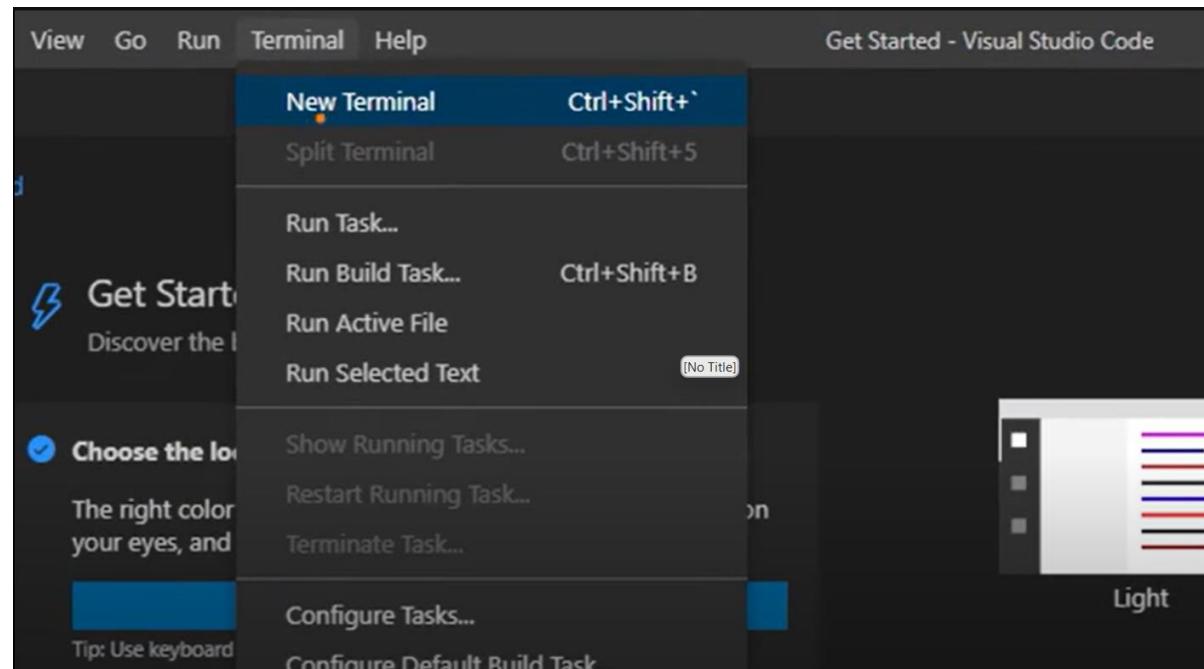
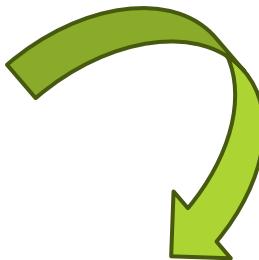
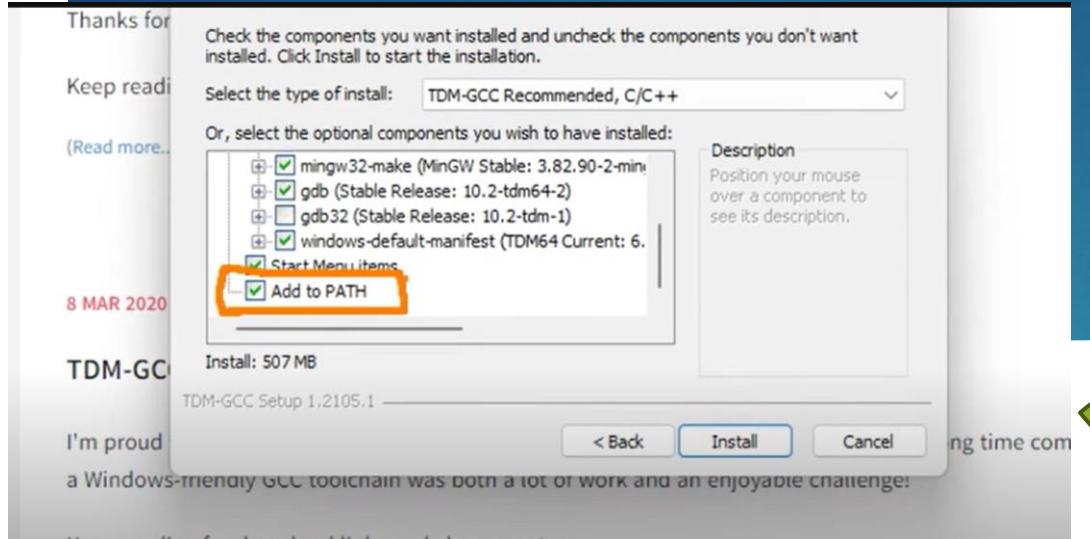
TDM-GCC Setup 1.2005.1

< Back

[Next >](#)

[Cancel](#)





File Edit Selection View Go Run Terminal Help

Get Started - Visual Studio Code

Get Started X

Get Started with VS Code

Discover the best customizations to make VS Code yours.

Choose the look you want

The right color palette helps you focus on your code, is easy on your eyes, and is simply more fun to use.

[Browse Color Themes](#)

Tip: Use keyboard shortcut (Ctrl+K Ctrl+T)

Sync to and from other devices

One shortcut to access everything

Light Dark Dark High Contrast Light High Contrast

See More Themes...

Code collects usage data. Read our [privacy statement](#) and learn how to [opt out](#).

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

gcc + ⊞ ☰ ^

Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\Peter> g++  
g++.exe: fatal error: no input files  
compilation terminated.  
PS C:\Users\Peter> gcc  
gcc.exe: fatal error: no input files  
compilation terminated.

[No Title]

**Get Started with VS Code**

Discover the best customizations to make VS Code yours.

**Choose the look you want**

The right color palette helps you focus on your code, is easy on your eyes, and is simply more fun to use.

**Browse Color Themes**

Tip: Use keyboard shortcut (Ctrl+K Ctrl+T)

**Sync to and from other devices**

**Get Started with VS Code**

Discover the best customizations to make VS Code yours.

**Choose the look you want**

The right color palette helps you focus on your code, is easy on your eyes, and is simply more fun to use.

**Browse Color Themes**

Tip: Use keyboard shortcut (Ctrl+K Ctrl+T)

**Sync to and from other devices**

**One shortcut to access everything**

**Rich support for all your languages**

**Open up your code**

File Edit Selection View Go Run Terminal Help

EXTENSIONS MARKETPLACE

**C/C++**

**C/C++** v1.10.8 Microsoft 33,481,909 ★★★★☆ (406) C/C++ IntelliSense, debugging, and code browsing. [Install](#)

**C/C++ Themes** v0.8.3 Microsoft UI Themes for C/C++ extensions. [Install](#)

**C/C++ External...** v0.7.2 Microsoft Popular extensions for C/C++ development. [Install](#)

**C/C++ Comp...** v1.0.6 danielpnata826 Compile & Run single c/c++ files. [Install](#)

**C/C++ Clang...** v1.0.4 Yousaki MITANI Completion and Diagnostic Tools for C/C++. [Install](#)

**C/C++ Runner** v1.5.0 franneck94 Compile, run and debug C/C++ projects. [Install](#)

**Kite AutoCom...** v0.3.0 Kite AI code completions for all languages. [Install](#)

File Edit Selection View Go Run Terminal Help

EXTENSIONS MARKETPLACE

**Code Runner** v0.11.8 Jun Han 13,699,841 ★★★★★ .run Run C, C++, Java, JS, PHP, Python, Perl, Ruby, Node.js, Go, C#, VB, Lua, R, RUST, AS3, OCaml, F#, Elixir, and many more. [Install](#)

**EditorConfig** v0.13.0 EditorConfig Support for Visual Studio Code. [Install](#)

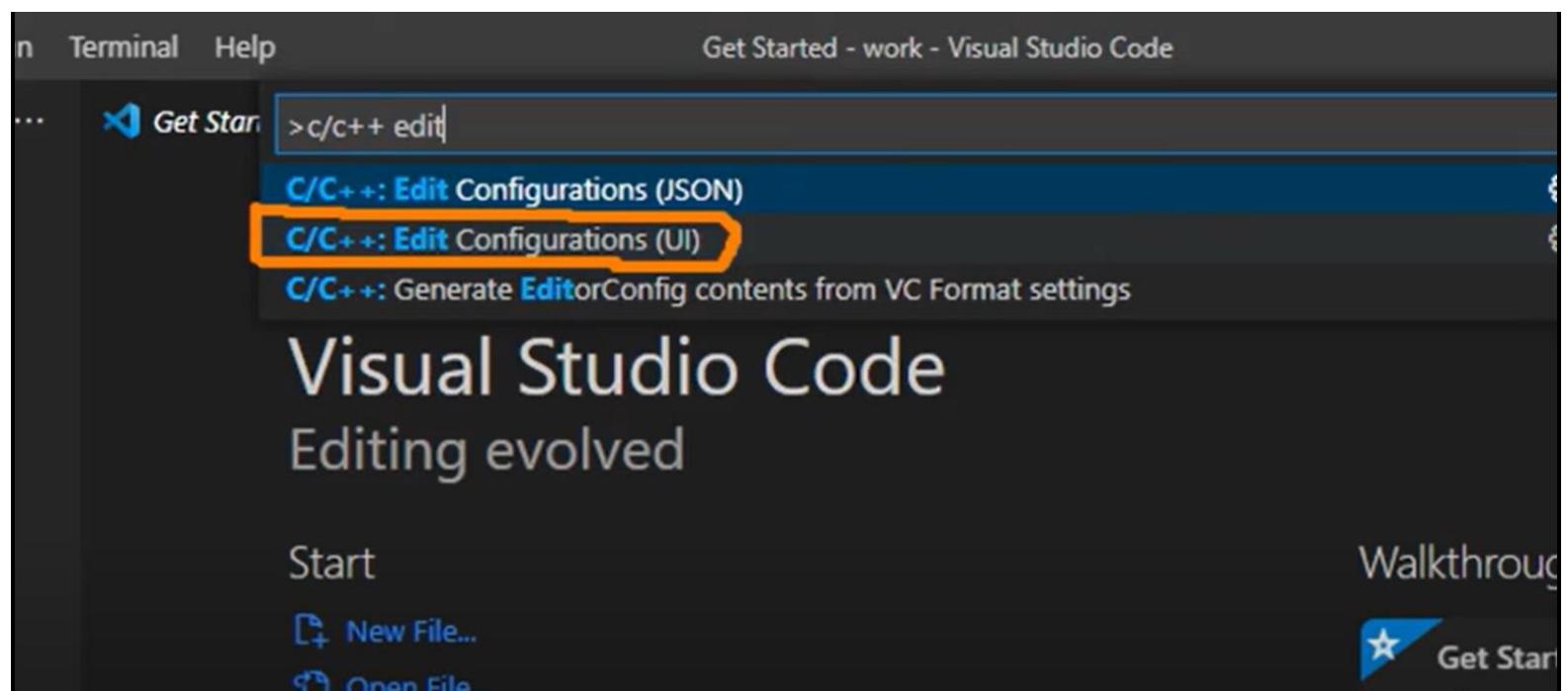
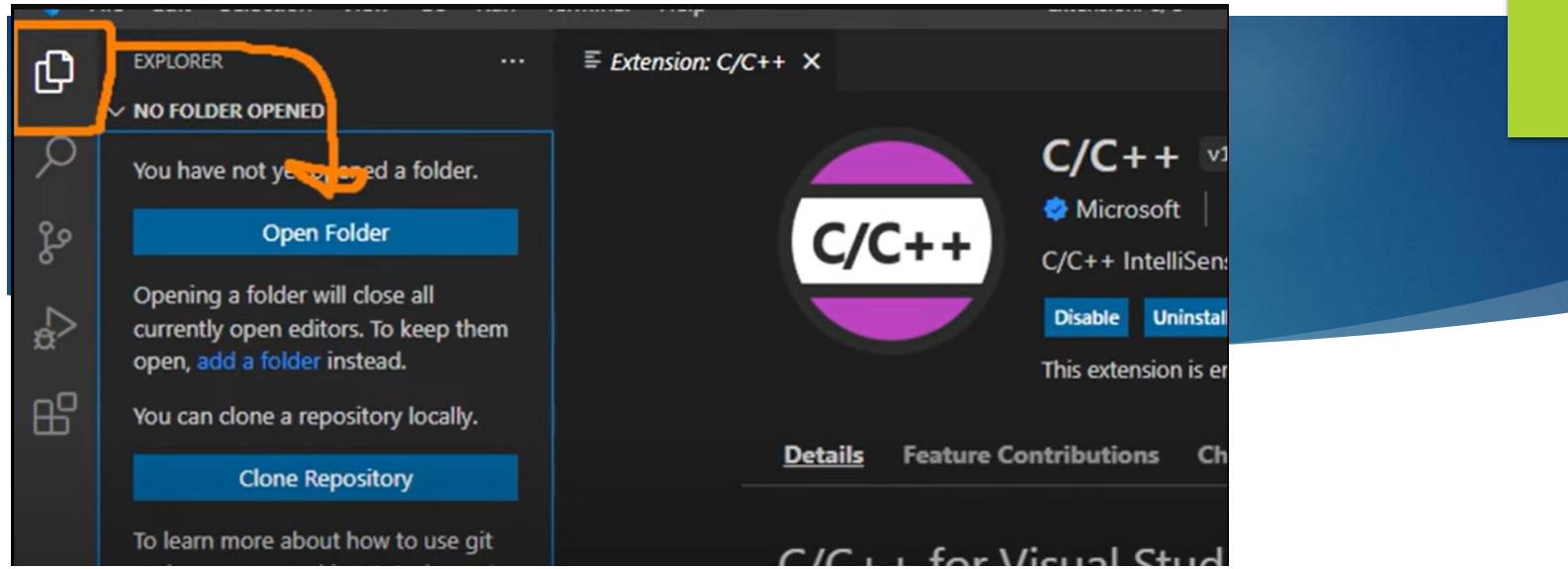
**C/C++ Compiler** v0.1.9 danielpnata826 Compile & Run single c/c++ files. [Install](#)

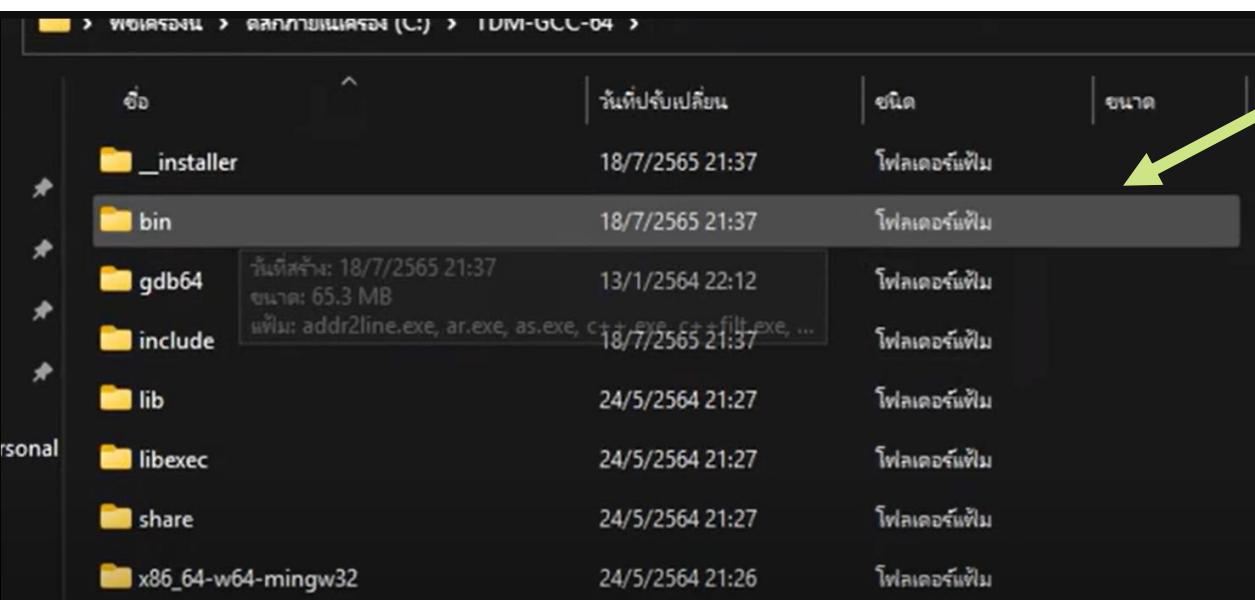
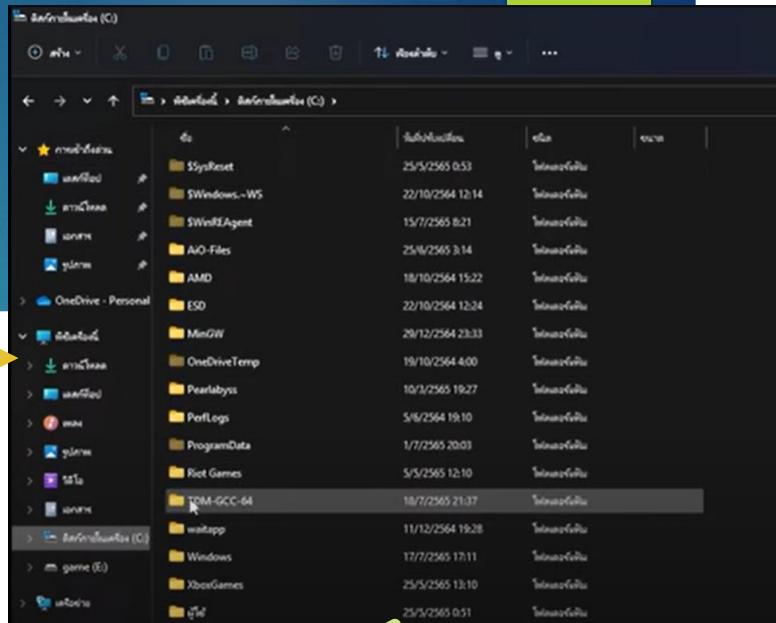
**JavaScript (ES6)** v0.8.0 charalampos karypis JavaScript snippets for Visual Studio Code. [Install](#)

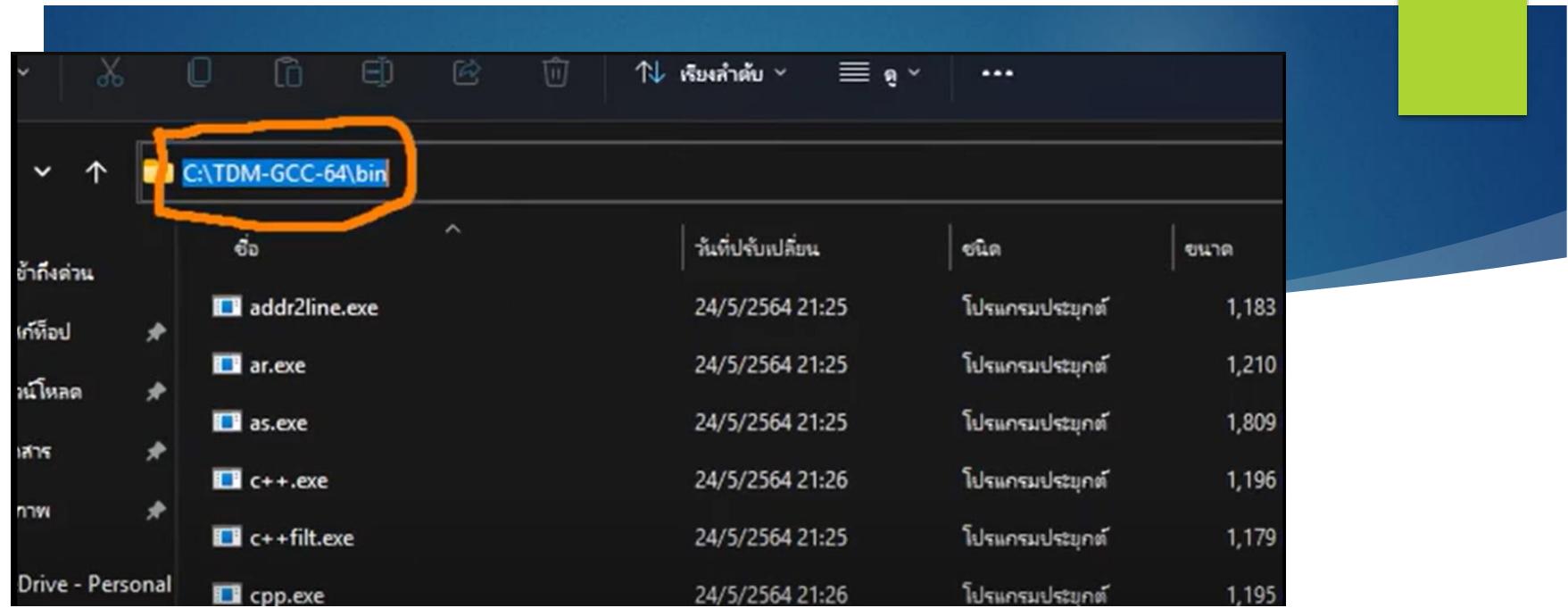
**Prettier - Code formatter** v2.2.1 Prettier Code formatter using prettier. [Install](#)

**Code Spell Checker** v0.4.1 Street Side Software Spelling checker for source code. [Install](#)

**Chinese (Simplified)** v0.17.0 Chinese (Simplified) localization for Visual Studio Code. [Install](#)







# Copy

## Compiler path

The full path to the compiler you use to build your project, e.g. /usr/bin/gcc, to enable more accurate IntelliSense. The extension will query the compiler to determine the system include paths and default defines to use for IntelliSense.

Specify a compiler path or select a detected compiler path from the drop-down list.

C:\TDM-GCC-64\bin\gcc.exe

The image shows a screenshot of the Visual Studio C/C++ Configuration dialog. The top section displays the configuration path: C/C++ Configurations - work - Visual Studio 2019. Below this, the 'IntelliSense mode' dropdown is set to 'windows-msvc-x64'. A red callout box highlights this selection, and a red arrow points from it to a larger screenshot of the same configuration dialog where 'gcc-x64 (legacy)' is selected. Another red callout box highlights the message 'IntelliSense mode windows-msvc-x64 is incompatible with compiler path.' A green arrow points from this message to the 'C standard' dropdown at the bottom right, which is set to 'c17'. A third green arrow points from the 'C standard' dropdown to the 'IntelliSense mode' dropdown, indicating a dependency or relationship between the two settings.

**C/C++ Configurations - work - Visual Studio 2019**

**IntelliSense mode**

The IntelliSense mode to use that maps to a platform and architecture. \${default}, the extension will choose the default for that platform. Windows defaults to windows-msvc-x64, Linux defaults to linux-gcc-x64, and macOS defaults to macos-clang-x64. Select a specific mode. IntelliSense modes that only specify <compiler>-<architecture> will be converted automatically to the <platform>-<compiler>-<architecture> mode.

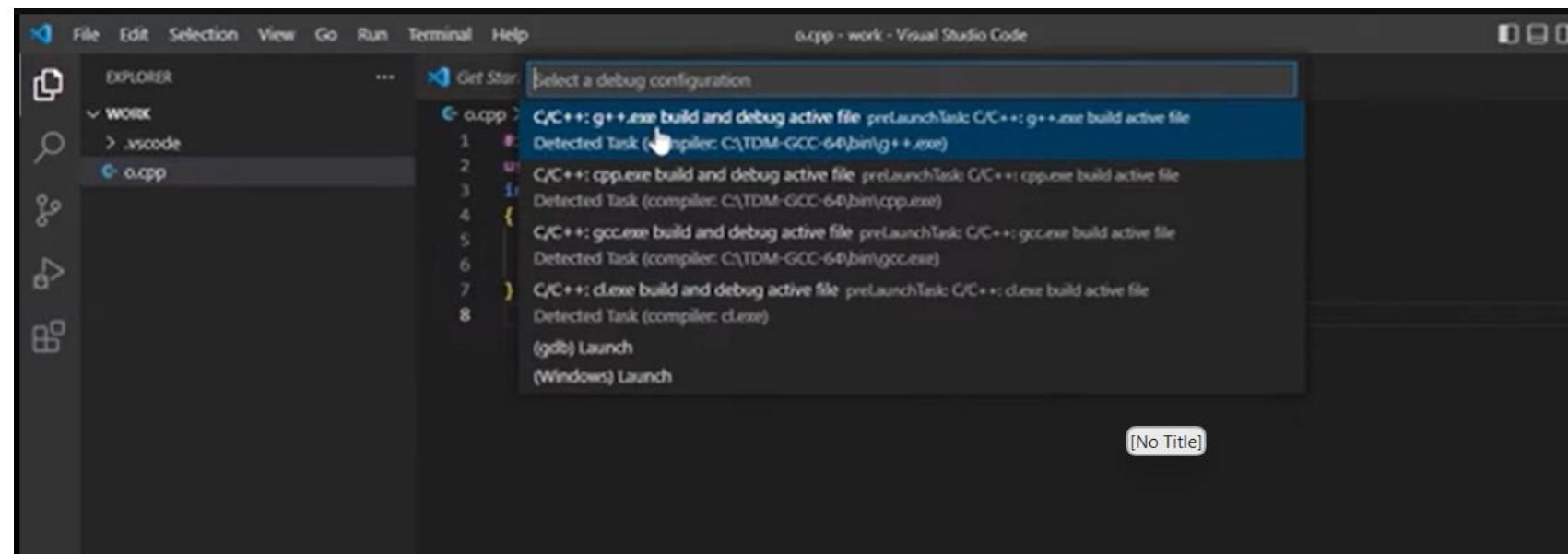
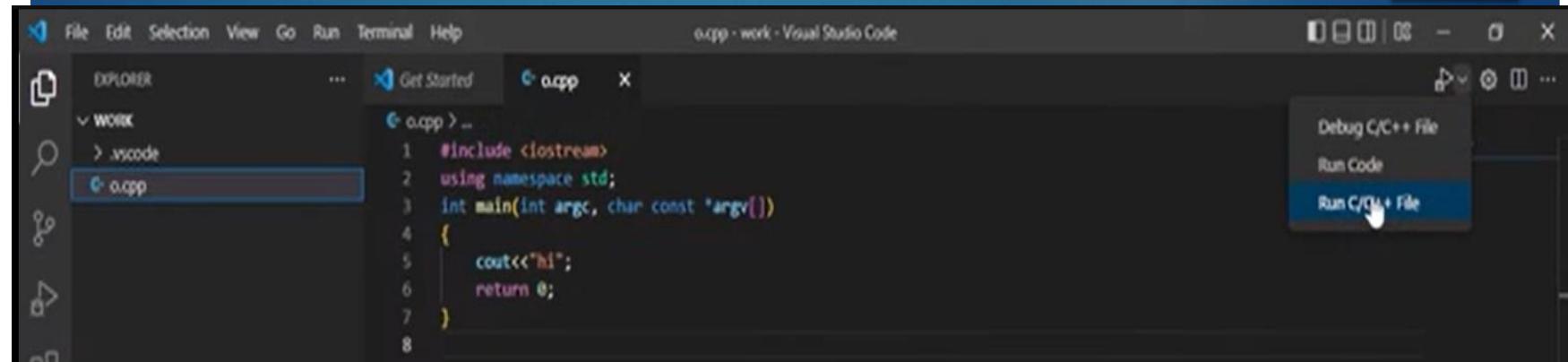
windows-msvc-x64

IntelliSense mode windows-msvc-x64 is incompatible with compiler path.

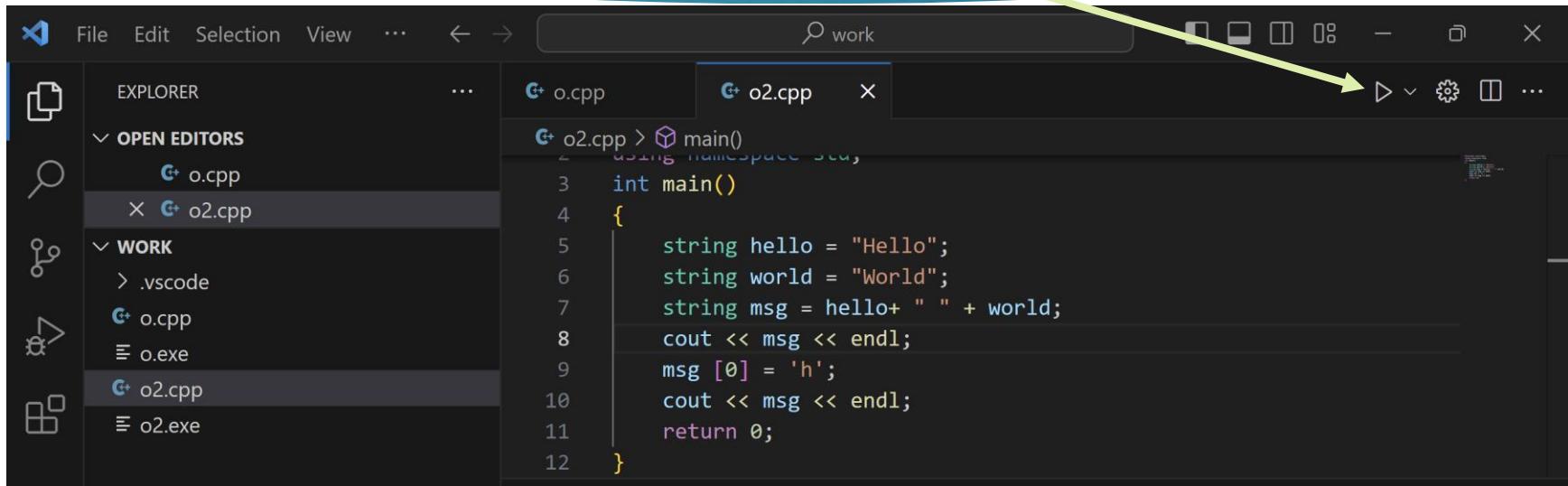
**C standard**

The version of the C language standard to use for IntelliSense. Note: GNU standards are only supported for Linux and macOS. For Windows, the standard is determined by the compiler. You can use the standard to get GNU defines, and IntelliSense will emulate the equivalent C standard version.

c17



## Run C\C++ file



Console window:

A screenshot of the VS Code terminal window. The title bar says "work". The terminal shows the output of running the C++ program. A yellow arrow points from the text "Console window:" to the terminal window.

```
PS D:\work> & 'c:\Users\Asus\.vscode\extensions\ms-vscode.cpptools-1.18.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-irh3q0om.cac' '--stdout=Microsoft-MIEngine-Out-5omasmgmt.b2b' '--stderr=Microsoft-MIEngine-Error-hnl2yea0.j4v' '--pid=Microsoft-MIEngine-Pid-vuqcah2h.rop' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Hello World
Hello World
PS D:\work>
```



Sublime Text

The screenshot shows a Google search results page for the query "sublime text 3". The search bar at the top contains the query. Below the search bar, there are several navigation links: ทั้งหมด (selected), ค้นรูป, วีดีโอ, ข่าวสาร, หนังสือ, เพิ่มเติม, and การตั้งค่า. The main search results section displays a list of links. The first result is highlighted with a red box and contains the title "Download - Sublime Text" and a link to "https://www.sublimetext.com/3". Below this, a snippet of text reads: "Sublime Text 3 is the current version of Sublime Text... Sublime Text may be downloaded and evaluated for free, however a license must be purchased for...". To the right of this result, there are two more entries: "Sublime Text 3.0" with a snippet about the new version, and "Sublime Text - Download" with a snippet about the download process. At the bottom of the page, there is a note about activating Windows.

Secure | https://www.google.co.th/search?q=sublime+text+3&oq=sublime+&aqs=chrome.1.69

sublime text 3

ทั้งหมด ค้นรูป วีดีโอ ข่าวสาร หนังสือ เพิ่มเติม การตั้งค่า

ผลการค้นหาประมาณ 89,300,000 รายการ (0.30 วินาที)

**Download - Sublime Text**  
<https://www.sublimetext.com/3> ▾ แปลงหน้านี้

Sublime Text 3 is the current version of Sublime Text... Sublime Text may be downloaded and evaluated for free, however a license must be purchased for...

**Sublime Text 3.0**

Sublime Text 3.0 is out! Compared to the last beta, 3.0 brings ...

**Documentation**

Column Selection - Auto Complete - Multiple Selection - Packages

**Sublime Text - Download**

Download. Sublime Text 3 may be downloaded from the Sublime ...

**Sublime Text 2**

Sublime Text 2 may be downloaded and evaluated for ...

Activate Windows



sublime text 3 package control install

ห้องน้ำด วิดีโอ ค้นรูป ข่าวสาร แผนที่ เพิ่มเติม การตั้งค่า

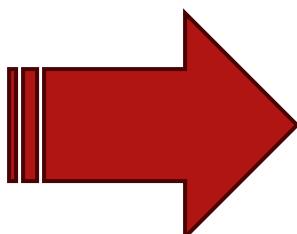
ผลการค้นหาประมาณ 423,000 รายการ (0.47 วินาที)

## Installation - Package Control

<https://packagecontrol.io/installation> แปลหน้า

The simplest method of **installation** is through the **Sublime Text** console. The console is accessed via the **ctrl+`** shortcut or the **View > Show Console** menu. Once open, paste the appropriate Python code for your version of Sublime Text into the console.

Sublime Text 3Sublime Text 2. import urllib.request,os ...



# Package Control

## INSTALLATION

### Simple

The simplest method of installation is through the Sublime Text console. The console is accessed via the **ctrl+`** shortcut or the **View > Show Console** menu. Once open, paste the appropriate Python code for your version of Sublime Text into the console.

```
SUBLIME TEXT 3 SUBLIME TEXT 2
import urllib.request,os,hashlib; h = 
'6f4c264a24d933ce70df5dedcf1dcae' + 
'ebbe013ee18cced0ef93d5f746d880ef60'; pf = 'Package' +
Control.sublime-package'; ipp = 
sublime.installed_packages_path(); 
urllib.request.install_opener(
urllib.request.build_opener(
urllib.request.ProxyHandler())); by = 
urllib.request.urlopen("http://packagecontrol.io/" + 
pf.replace(' ', '%20')).read(); dh = 
hashlib.sha256(by).hexdigest(); print('Error validating' + 
'download (got %s instead of %s), please try manual' + 
'install' % (dh, h)) if dh != h else open(os.path.join( 
ipp, pf), 'wb').write(by)
```

This code creates the **Installed Packages** folder for you (if necessary), and then downloads the **Package Control.sublime-package** into it. The download will be done over HTTP instead of HTTPS due to Python standard library limitations, however the file will be validated using SHA-256.

**WARNING:** Please do not redistribute the install code via another website. It will change with every release. Instead, please link to this page.

Search

Usage Instructions

Installation

Browse

Search

Docs

News

Stats

About

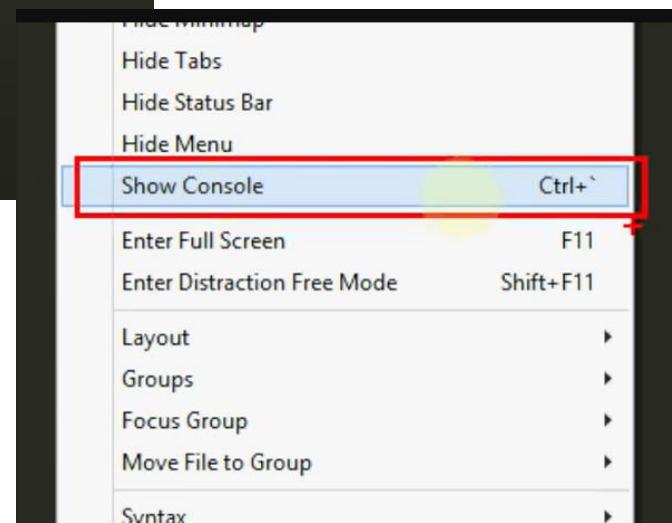
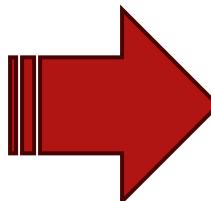
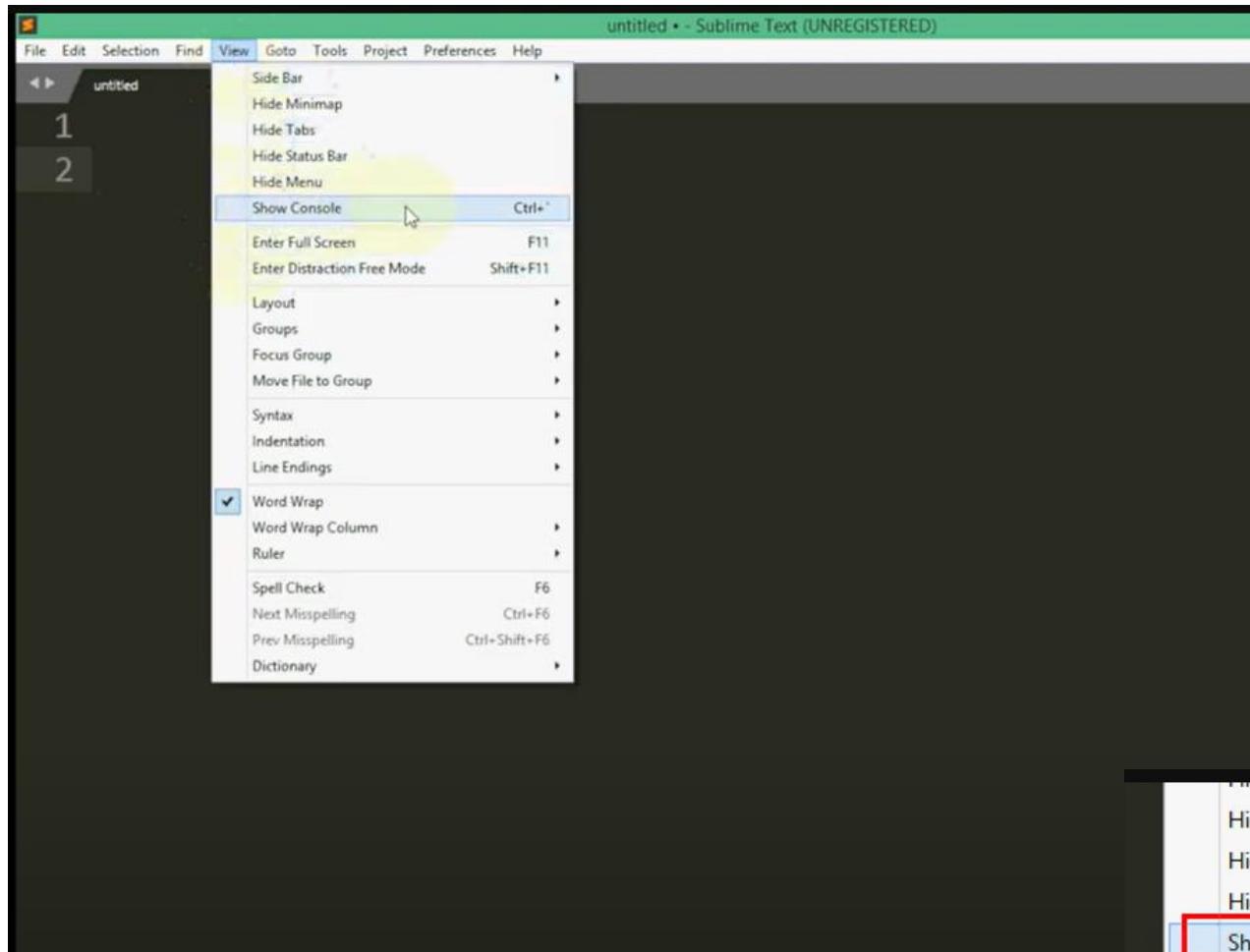
Say Thanks

SUPPORTED VIA



Activate Windows

Go to PC settings to activate Windows



A screenshot of a Sublime Text window titled "untitled • - Sublime Text (UNREGISTERED)". The window has a dark theme. In the top left, there's a small orange icon. The menu bar includes File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, and Help. Below the menu is a toolbar with icons for back, forward, and search. The main area shows two lines of code: "1" and "2". At the bottom of the window, a terminal session is displayed:

```
reloading settings Packages/User/Preferences.sublime-settings
>>> import urllib.request,os,hashlib; h = '6f4c264a24d933ce70df5dedcf1dcaee' + 'ebe013ee18cced0ef93d5f746d80ef60'; pf = 'Package Control.sublime-package'; ipp = sublime.installed_packages_path(); urllib.request.install_opener( urllib.request.build_opener(urllib.request.ProxyHandler()) ); by = urllib.request.urlopen( 'http://packagecontrol.io/' + pf.replace(' ', '%20')).read(); dh = hashlib.sha256(by).hexdigest(); print('Error validating download (got %s instead of %s), please try manual install' % (dh, h)) if dh != h else open(os.path.join( ipp, pf), 'wb' ).write(by)
286331
```

# Package Control

Search

Install Now

The Sublime Text package manager that makes it exceedingly simple to find, install and keep packages up-to-date.

## Trending 🌏

A recent, relative, increase in installs

Wolf Theme  
rem-unit  
Sublime Input  
PHP Form Builder  
Acme Color Scheme  
Placeholders  
NodeRequirer  
CSS Color Converter  
PackageSync  
SublimeLinter-flake8

## New 🌱

Just added to Package Control

- Software
- SublimeLinter-contrib-rosline
- PureScript Syntax
- CAOS Syntax Highlighter
- BreakLines
- Mellanox Syntax Highlighter
- Regex Escape
- Mattermost Post
- PaletteFile
- JavaScriptExtractFunction

## Popular ⚡

Randomly selected from the top 100

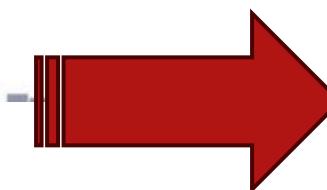
- Emmet Style Reflector
- SFTP
- Javascript Beautify
- HTML5
- Predawn
- JavaScript Completions
- SublimeCodeIntel
- Phpcs
- FTPSync
- Seti\_UI

## Labels 🎯

Labels with the biggest selection

- language syntax 542
- snippets 427
- color scheme 308
- linting 227
- theme 171
- auto-complete 171
- text manipulation 107
- javascript 93
- formatting 91
- build system 66

Realtime



## Popular ⚡

Randomly selected from the top 100

- Emmet Style Reflector
- SFTP
- Javascript Beautify
- HTML5
- Predawn
- JavaScript Completions
- SublimeCodeIntel
- Phpcs
- FTPSync
- Seti\_UI

+

# Popular

Package Control *by Will Bond (wbond)*

A full-featured package manager

Emmet *by sergeche*

Emmet for Sublime Text

SideBarEnhancements *by SideBarEnhancements-org* ST3

Enhancements to Sublime Text sidebar. Files and folders.

SublimeLinter *by SublimeLinter*

The code linting framework for Sublime Text 3



HTML5 *by mrmartineau*

HTML5 bundle for Sublime Text

Alignment *by Will Bond (wbond)*

Easy alignment of multiple selections and multi-line selections

ColorPicker *by Weslly H.*

A multi-platform color picker plugin



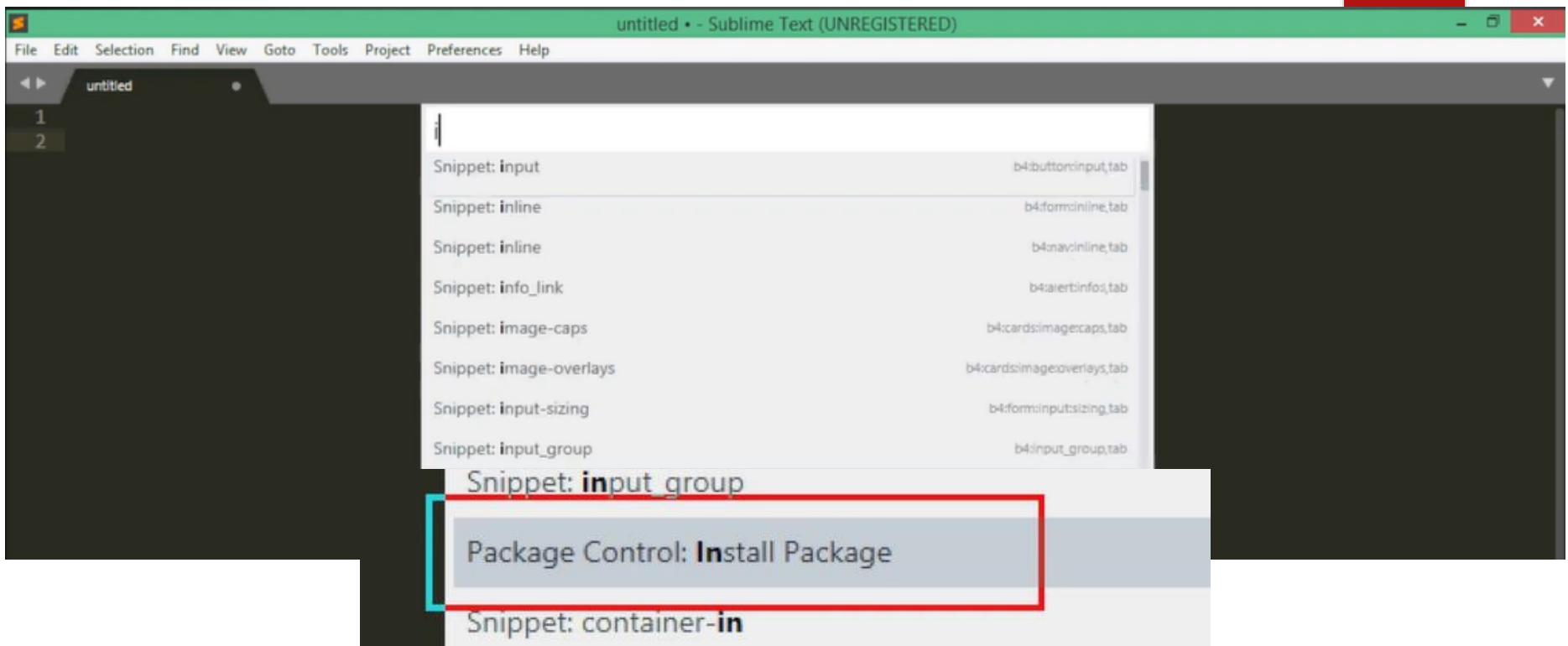
SASS *by P233*

A Sublime Text 3 package for highlighting both Sass and SCSS syntax.

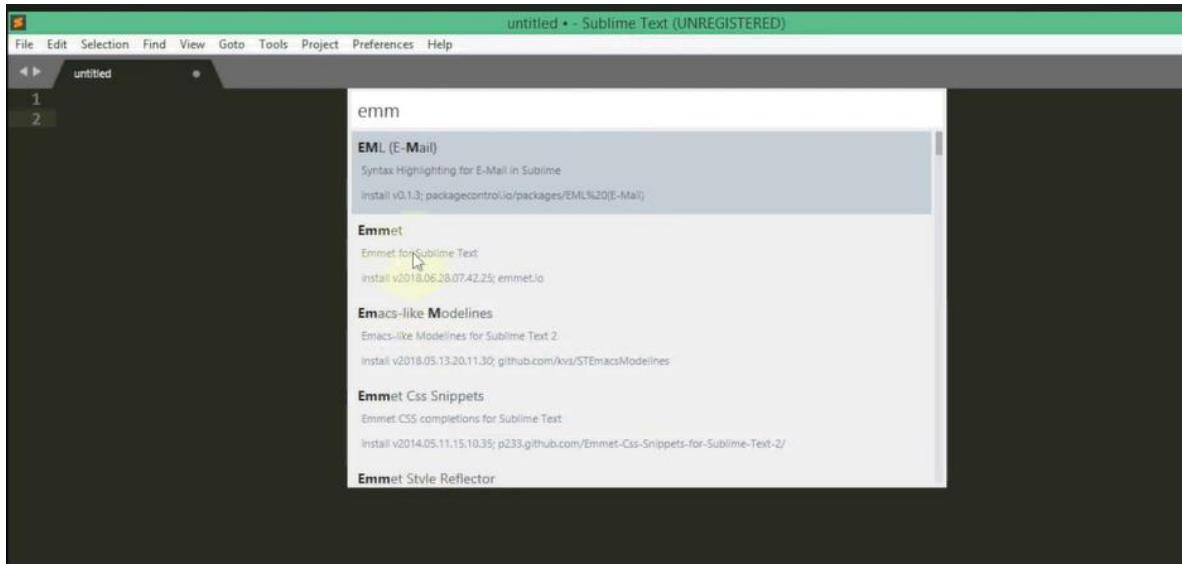
jQuery *by SublimeText*

Sublime Text package bundle for jQuery

Activate VI  
Go to PC setti



กด **ctrl+shift+p**



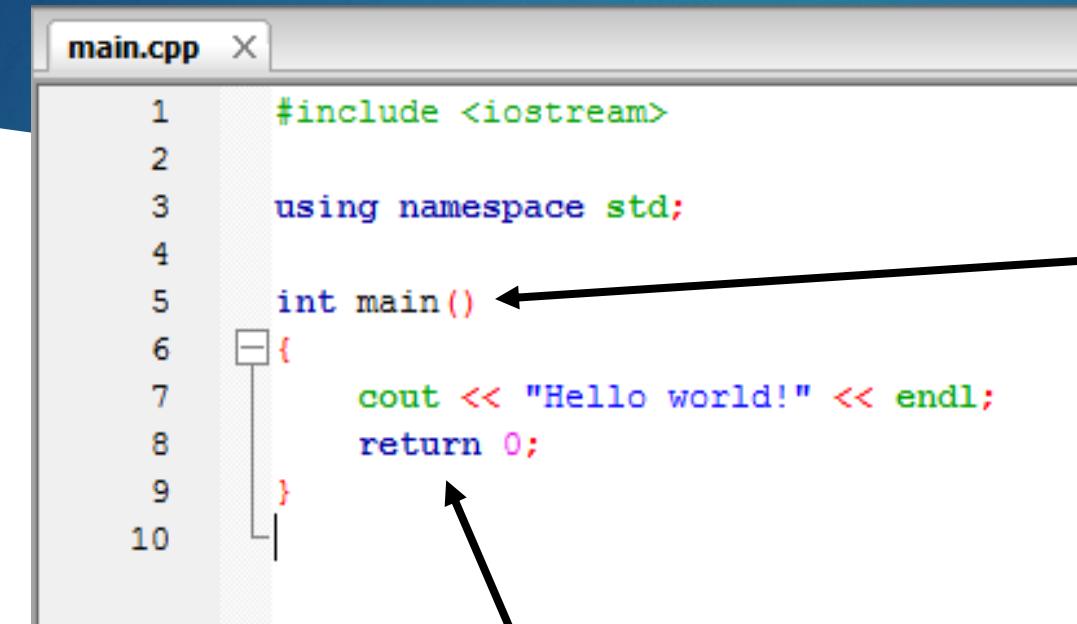
The screenshot shows the Sublime Text interface with the title bar "Package Control Messages - Sublime Text (UNREGISTERED)". The menu bar includes File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, Help, and a "Package Control Messages" item. The main area displays the following text:

```
1 Package Control Messages
2 =====
3
4 Emmet
5 -----
6
7 Thank you for installing Emmet -- a toolkit that can greatly improve your workflow. Note that this plugin
8 automatically downloads and installs PyV8 binary (see status bar message).
9 -----
10 Please restart editor
11 to finish installation process
12 after PyV8 was downloaded.
13 -----
14 Tab key handler
15 -----
16 -----
17 By default, Emmet allows you to expand abbreviations with Tab key in HTML, XML, HAML and CSS/SASS/LESS/Stylus
18 documents. As a side effect, you can't use some of your ST2 snippets.
19
20 Please read https://github.com/sergeche/emmet-sublime#tab-key-handler about how Tab handler works and how to tweak
21 its behavior to match your needs.
22 Enter key
23 -----
24
25 In HTML and XML documents, Emmet overrides Enter key to insert formatted line breaks between opening and closing
tags. In some cases it will break character input (for example, in Japanese language).
26
27 To disable Enter key handler, simply add the following option in your user's Preferences file:
28
29 "disable_formatted_linebreak": true
30
31 Actions shortcuts
32 -----
33
34 Emmet has a number of actions with keyboard shortcuts that may override ones you're using commonly (for example,
Ctrl+F or Ctrl+Down). Please read the package documentation for list of available actions and keyboard shortcuts.
```

# C++

- ▶ Introduction C++
- ▶ Array
- ▶ Pointer
- ▶ Vector

## Hello, World! explained



```
main.cpp X
1 #include <iostream>
2
3 using namespace std;
4
5 int main() { ←
6     cout << "Hello world!" << endl;
7     return 0; ↑
8 }
9
10 }
```

main – เป็นจุดเริ่มต้นของทุกๆ โปรแกรมใน C++ มันจะทำการ returns integer value ให้กับระบบปฏิบัติการ และ (ในกรณีนี้) ไม่รับ arguments: main()

คำสั่ง return จะส่งค่าจำนวนเต็มกลับไปยังระบบปฏิบัติการหลังจากทำงานเสร็จสิ้น ค่า 0 หมายถึง 'ไม่มีข้อผิดพลาด' โปรแกรม C++ จะต้องส่งค่าจำนวนเต็มกลับ.

# Hello, World! explained

loads header file containing function and class definitions

```
main.cpp X
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      cout << "Hello world!" << endl;
8
9  }
10
```

โหลด namespace ชื่อ std โดย namespace ใช้เพื่อความสะดวกของโปรแกรมเมอร์

- cout เป็น object ที่เขียนข้อความไปยังอุปกรณ์ เช่น console window.
- เป็นส่วนหนึ่งของ librariematrฐาน ในC++ ถ้าไม่มี “using namespace std;” เราจะต้องพิมพ์ว่า std::cout. ซึ่งถูกนิยามไว้ใน iostream header file.
- "<<" เป็นตัวดำเนินการ การแทรกช่องมูลใน C++ โดยใช้ส่งตัวอักษรจากด้านขวาไปยังวัตถุทางซ้าย ส่วน endl เป็นตัวอักษรขึ้นบรรทัดใหม่ใน C++

# Basic Syntax

- **Syntax** ของ **C++** มีความคล้ายกับ **C, Java, และ C#**มาก เริ่มต้นโปรแกรมด้วยการใช้ฟังก์ชัน **main()**

- ปิดท้ายแต่ละบรรทัดด้วย ;

```
int a ;
a = 1 + 3 ;
```

```
// this is a comment.
/* everything in here
   is a comment */
```

- **Comments** แต่ละบรรทัดด้วย // และหลาย ๆ บรรทัดใช้ /\*... \*/

- ตัวแปรสามารถประกาศได้ตลอดเวลาภาคใน **block** ของ **code**.

```
void my_function() {
    int a ;
    a=1 ;
    int b;
}
```

## Functions

## arguments

```
int add(int x, int y) {  
    int z = x + y ;  
    return z ;  
}
```

```
// No arguments? Still need ():  
void my_function() {  
    /* do something...  
       but a void value means the  
       return statement can be  
       skipped.*/  
}
```

**void** ไม่มีการ **return** ค่ากลับ

ตัวแปร จะถูกประกาศ ด้วยการใช้ ชนิดของตัวแปร และ ชื่อของตัวแปร

```
// Specify the type  
int x = 100;  
float y;  
vector<string> vec ;  
// Sometimes types can be inferred  
auto z = x;
```

## Built-in (aka primitive or intrinsic) Types

Name	Name	Value
char	unsigned char	8-bit integer
short	unsigned short	16-bit integer
int	unsigned int	32-bit integer
long	unsigned long	64-bit integer
bool		true or false

Name	Value
float	32-bit floating point
double	64-bit floating point
long long	128-bit integer
long double	128-bit floating point

# Example Program

```
#include <iostream>
using namespace std;

int main()
{
    double num1 = 5,
           num2,
    sum; num2 = 12;

    sum = num1 + num2;
    cout << "The sum is " << sum; return 0;
}
```

# The Parts of a C++ Program

Statement	Purpose
// sample C++ program	comment
#include <iostream>	preprocessor directive
using namespace std;	namespace to use
int main()	beginning of function named main
{	beginning of block for main
cout << "Hello, there!" ;	output statement
return 0;	send 0 back to the operating system
}	end of block for main

# Special Characters

Character	Name	Description
//	Double Slash	Begins a comment
#	Pound Sign	Begins preprocessor directive
< >	Open, Close Brackets	Encloses filename used in #include directive
( )	Open, Close Parentheses	Used when naming function
{ }	Open, Close Braces	Encloses a group of statements
" "	Open, Close Quote Marks	Encloses string of characters
;	Semicolon	Ends a programming statement

# Important Details

- C++ is case-sensitive. Uppercase และ lowercase นั่นคือ ‘Main’ ไม่เหมือนกับ ‘main’.
- ทุกๆ โปรแกรมต้องมี { ... }

# The cout Object

- แสดงข้อมูลบนหน้าจอคอมพิวเตอร์
- ใช้ << เพื่อส่งข้อมูลไปยัง cout
  - `cout << "Hello, there!" ;`
- สามารถใช้ << เพื่อส่งหลายรายการไปยัง cout ได้
  - `cout << "Hello, " << "there!" ;`
  - หรือ
  - `cout << "Hello, " ;`
  - `cout << "there!" ;`

# Starting a New Line

- เพื่อให้ได้ผลลัพธ์หลายบรรทัดบนหน้าจอ

- Use `endl`

```
cout << "Hello, there!" << endl;
```

- Use `\n` in an output string

```
cout << "Hello, there!\n";
```

# Escape Sequences – More Control Over Output

Escape Sequence	Name	Description
\n	Newline	Causes the cursor to go to the next line for subsequent printing.
\t	Horizontal tab	Causes the cursor to skip over to the next tab stop.
\a	Alarm	Causes the computer to beep.
\b	Backspace	Causes the cursor to back up, or move left one position.
\r	Return	Causes the cursor to go to the beginning of the current line, not the next line.
\\\	Backslash	Causes a backslash to be printed.
\'	Single quote	Causes a single quotation mark to be printed.
\"	Double quote	Causes a double quotation mark to be printed.

# Variables, Literals, and the Assignment Statement

- Variable

- ประกอบไปด้วยชื่อและประเภทของข้อมูล



- ใช้เพื่ออ้างอิงถึงตำแหน่งในหน่วยความจำที่สามารถเก็บค่าได้
  - ต้องกำหนดก่อนที่จะใช้งานได้
  - ค่าที่เก็บไว้สามารถเปลี่ยนแปลงได้

# Variables

- หากมีการเก็บค่าใหม่ในตัวแปร ค่าก่อนหน้านั้นจะถูกแทนที่ด้วยค่าใหม่
  - ค่าก่อนหน้านั้นถูกเขียนทับและไม่สามารถเรียกคืนได้อีกต่อไป
- ```
int age;  
age = 17;      // age is 17  
cout << age; // Displays 17  
age = 18;      // Now age is 18  
cout << age; // Displays 18
```

# Constants

## Literal

- ค่าที่ไม่เปลี่ยนแปลงระหว่างการ execute โปรแกรม
- เรียกว่า constant

'A' // character constant

"Hello" // string literal

12 // integer constant

3.14 // floating-point constant

# Multi-word Variable Names

- ชื่อตัวแปรที่ใช้คำอธิบายอาจประกอบไปด้วยคำหลายคำ
- มีสองแนวทางในการตั้งชื่อตัวแปรดังนี้:
- ใช้ตัวพิมพ์ใหญ่ในอักษรแรกของคำ

quantityOnOrder

totalSales

- ใช้ underscore \_ แทนช่องว่าง:

quantity\_on\_order

total\_sales

# Valid and Invalid Identifiers

| IDENTIFIER  | VALID? | REASON IF INVALID       |
|-------------|--------|-------------------------|
| totalSales  | Yes    |                         |
| total_Sales | Yes    |                         |
| total.Sales | No     | Cannot contain period   |
| 4thQtrSales | No     | Cannot begin with digit |
| totalSale\$ | No     | Cannot contain \$       |

# Integer Data Types

- **Integer** ถูกออกแบบเพื่อเก็บเลขจำนวนเต็ม ไม่มีทศนิยม
- สามารถเป็นได้ทั้งแบบ **signed** หรือ **unsigned**  
**12**            **-6**            **+3**
- สามารถกำหนดให้มีขนาดที่แตกต่างกันได้ (*i.e.*, number of bytes):  
**short, int, and long**
- โดยขนาดของ **Short**  $\leq$  size of **int**  $\leq$  size of **long**

# Signed vs. Unsigned Integers

- C++ จะจัดสรร 1 bit สำหรับการกำหนด ของตัวเลข และ bit ที่เหลือจะแทนข้อมูล
- ถ้าโปรแกรมที่เราสร้างไม่เคยที่จะเป็นลบเลย เราสามารถประกาศตัวแปรเป็น `unsigned` ได้เลย
- โดยปกติถ้าไม่มีการกำหนดอย่างไร ตัวแปรจะเป็น `signed`

# Defining Variables

- ตัวแปรที่เป็น type เดียวกัน สามารถนิยามได้เป็น
  - แบบคำสั่งแยก

```
int length;
```

```
int width;
```
  - อุปในคำสั่งเดียวกัน

```
int length, width;
```
- ตัวแปรต่างชนิดกันจะต้องนิยามแยกกันเท่านั้น

# Integral Constants

- เพื่อที่จะเก็บค่าคงที่จำนวนเต็มในตัวแปรชนิด long ให้ใส่ 'L'  
ต่อท้ายตัวเลข **1234L**
- ค่าคงที่ที่ขึ้นต้นด้วย '0' (zero) คือเลขฐาน 8: **075**
- ค่าคงที่ ที่ขึ้นต้นด้วย '0x' จะเป็นเลขฐาน 16: **0x75A**

# Floating-Point Data Types

- ออกแบบมาเพื่อทำการเก็บเลขจำนวนจริง  
**12.45                          -3.8**
- ทุกจำนวนเป็น **signed**
- มีขนาดให้เลือกหลายรูปแบบ  
**float, double, และ long double**
- ขนาดของ **float**  $\leq$  size of **double**  
 $\leq$  size of **long double**

# Floating-point Constants

- ทศนิยมปกติ:

**31.4159**

**0.0000625**

- รูปที่มี e :

**3.14159E1**

**6.25e-5**

- ค่าเริ่มต้นของตัวแปรที่เก็บจำนวนจริงใน C++ คือ double
- แต่สามารถกำหนดให้เป็น float ด้วยการใส่ตัวอักษร 'F' ต่อท้ายตัวเลข เช่น 3.14159F
- หรือกำหนดให้เป็น long double ด้วยการใส่ตัวอักษร 'L' ต่อท้ายตัวเลข เช่น 0.0000625L

# Assigning Floating-point Values to Integer Variables

ถ้ามีการกำหนดค่าจำนวนจริงให้กับตัวแปรจำนวนเต็มใน C++  
จะทำการตัดทศนิยมออก (หรือก็คือตัดเศษทิ้งไป) และใช้เฉพาะจำนวนเต็ม  
เท่านั้น  
ค่าจำนวนจริงจะไม่ถูกปัดเศษทิ้งไป แต่จะถูกตัดไปโดยทั้งหมด

```
int rainfall = 3.88;  
cout << rainfall;           // Displays 3
```

# The **char** Data Type

- ใช้เพื่อเก็บตัวอักษรเดี่ยว
- ใช้พื้นที่เก็บ 1 byte
- การเก็บในหน่วยความจำจะเก็บเป็นรหัสตัวเลขที่แทนตัวอักษร

SOURCE CODE

```
char letter = 'C'; letter
```

MEMORY

67

# Character Literal

- เมื่ออ้างอิงถึงตัวอักษรสัญลักษณ์ดังกล่าวในโปรแกรม จะต้องครอบคลุมด้วยเครื่องหมายอัญประกาศเดี่ยว (single quotation marks) ดังนี้:
  - `cout << 'Y' << endl;`
  - เครื่องหมายอัญประกาศไม่ใช่ส่วนหนึ่งของตัวอักษร สัญลักษณ์เอง และจะไม่ถูกแสดงผลในการแสดงผลทางหน้าจอ

# String Literals

- ข้อความ (string) สามารถเก็บได้เป็นชุดของตัวอักษรติดต่อกันในหน่วยความจำต่อเนื่องกันได้ เช่น "Hello"
- แต่ในการเก็บข้อมูลแบบนี้จะต้องมีการเพิ่มตัวอักษรหนึ่งตัวเพิ่มเติมเรียกว่า null terminator เพื่อบอกว่าตัวอักษรที่เก็บจบลงแล้ว ตัวอักษร null terminator จะมีค่าเป็น \0 และจะถูกเพิ่มอัตโนมัติต่อท้ายข้อความที่ถูกเก็บในหน่วยความจำ
- ข้อความ (string) ประกอบด้วยตัวอักษรที่อยู่ระหว่างเครื่องหมายคำพูดคู่ (double quotation marks) " " เช่น "Hello World" เป็นต้น

|   |   |   |   |   |    |
|---|---|---|---|---|----|
| H | e | l | l | o | \0 |
|---|---|---|---|---|----|

# The C++ **string** Class

- `#include <string>` ใช้ในการสร้าง **string object**
- นิยามตัวแปร **string**  
`string name;`
- สามารถกำหนดค่าของตัวแปรได้  
`name = "George";`
- สามารถใช้ `cout` ในการแสดงผลลัพธ์ได้  
`cout << "My name is " << name;`

# The **bool** Data Type

- แทนค่าในรูปแบบ **true** หรือ **false**
- **bool** ค่าจะถูกเก็บในรูปแบบของ **integer**
- **false** ถูกแทนด้วย 0, **true** แทนด้วย 1

```
bool allDone = true;  
bool finished = false;
```

| allDone | finished |
|---------|----------|
| 1       | 0        |

# Determining the Size of a Data Type

คำสั่ง sizeof เป็นตัวดำเนินการ (operator) ในภาษาโปรแกรม C++ ที่ใช้เพื่อหาขนาดของข้อมูลหรือตัวแปรใดๆ ในหน่วยความจำ (memory) โดยขนาดจะถูกตีความเป็นจำนวน byte (ไบต์) ที่ใช้ในการเก็บข้อมูลนั้นๆ ตัวอย่างการใช้ sizeof operator ในการหาขนาดของตัวแปรชนิด int:

```
int x;  
std::cout << "Size of int variable x is: " << sizeof(x)  
<< " bytes\n";
```

ผลลัพธ์ที่ได้จะแสดงขนาดของตัวแปร x ในหน่วย byte บนหน้าจอเช่น "Size of int variable x is: 4 bytes" (หมายความว่าตัวแปรชนิด int ใช้พื้นที่ในหน่วยความจำ 4 ไบต์)

# Binary Arithmetic Operators

| SYMBOL | OPERATION             | EXAMPLE             | ans       |
|--------|-----------------------|---------------------|-----------|
| +      | <b>addition</b>       | <b>ans = 7 + 3;</b> | <b>10</b> |
| -      | <b>subtraction</b>    | <b>ans = 7 - 3;</b> | <b>4</b>  |
| *      | <b>multiplication</b> | <b>ans = 7 * 3;</b> | <b>21</b> |
| /      | <b>division</b>       | <b>ans = 7 / 3;</b> | <b>2</b>  |
| %      | <b>modulus</b>        | <b>ans = 7 % 3;</b> | <b>1</b>  |

# / Operator

- C++ division operator (/) การหารเป็นจำนวนเต็ม ผลลัพธ์ของการหารจะเป็นจำนวนเต็มเสมอ

```
cout << 13 / 5;      // displays 2  
cout << 2 / 4;      // displays 0
```

- แต่ถ้าทั้งสองตัวดำเนินการเป็นจำนวนทศนิยม ผลการหารจะเป็นจำนวนทศนิยมเหมือนกัน

```
cout << 13 / 5.0;    // displays 2.6  
cout << 2.0 / 4;     // displays 0.5
```

# % Operator

- C++ modulus operator (%) คำนวณเอาเศษ

```
cout << 9 % 2; // displays 1
```

- % cout << 9 % 2.0; // error

- ต้องเป็น integer ทั้งคู่

# Single-Line Comments

- Begin with `//` and continue to the end of line

```
int length = 12; // length in inches
int width = 15; // width in inches
int area; // calculated area

// Calculate rectangle area
area = length * width;
```

# Multi-Line Comments

- Begin with `/*` and end with `*/`
- Can span multiple lines

```
/*-----  
   Here's a multi-line comment  
-----*/
```

- Can also be used as single-line comments

```
int area; /* Calculated area */
```



## ► Expressions and Interactivity

# The **cin** Object

- มาตรฐาน input object
- เหมือนกับ **cout**, ต้องการการเรียกใช้ **iostream** file
- ใช้อ่าน input จาก keyboard
- ข้อมูลดึงจาก **cin >>** เพื่อไปเก็บในตัวแปร

# The `cin` Object

- `cin` สามารถใช้รับค่าหlays ๆ ตัวแปรพร้อมกันได้
- `cin >> height >> width;`
- ค่าที่รับเข้ามาไม่จำเป็นต้องเป็นชนิดเดียวกันทั้งหมด แต่ ลำดับของค่าจะมีความสำคัญ ค่าแรกที่ป้อนเข้ามาจะถูกเก็บไว้ในตัวแปรแรก ค่าถัดไปจะถูกเก็บไว้ในตัวแปรถัดไป ตามลำดับ

# Mathematical Expressions

- นิพจน์ (expression) สามารถเป็นค่าคงที่ (constant), ตัวแปร (variable), หรือสมการทางคณิตศาสตร์ที่ประกอบด้วยค่าคงที่และตัวแปรซึ่งรวมกันด้วยตัวดำเนินการ (operator) ต่าง ๆ สามารถสร้างนิพจน์ที่ซับซ้อนได้โดยใช้ตัวดำเนินการคณิตศาสตร์หลายตัวเข้าด้วยกัน เช่น

2

height

a + b / c

# Using Mathematical Expressions

- สามารถใช้ Mathematic Expression กับ คำสั่ง cout ได้

- Examples:

```
area = 2 * PI * radius;  
cout << "border is: " << (2*(l+w));
```



This is an  
expression

# Order of Operations

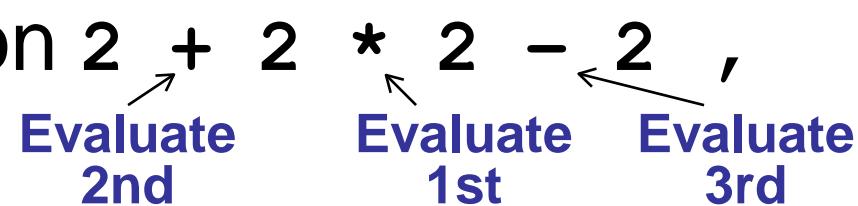
- ถ้า expression มี > 1 operator, จะต้องทำงานตามลำดับ

**Do first:** ( ) expressions in parentheses

**Do next:** - (unary negation) in order, left to right

**Do next:** \* / % in order, left to right

**Do last:** + - in order, left to right

- In the expression  $2 + 2 * 2 - 2$ ,  
  
Evaluate 2nd      Evaluate 1st      Evaluate 3rd

# Multiple and Combined Assignment

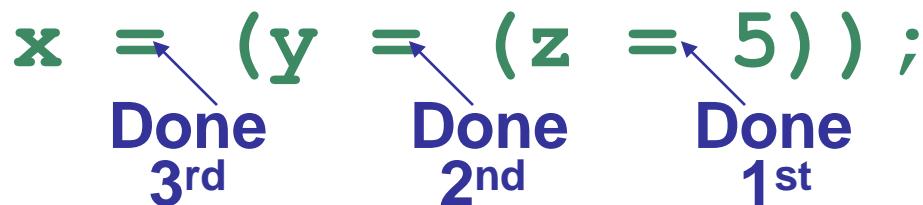
- การใช้ operator (=) สามารถใส่ได้หลายครั้งใน 1 expression

`x = y = z = 5;`

- Associates right to left

`x = (y = (z = 5)) ;`

Done 3rd      Done 2nd      Done 1st



## More Examples

`x += 5;` means `x = x + 5;`

`x -= 5;` means `x = x - 5;`

`x *= 5;` means `x = x * 5;`

`x /= 5;` means `x = x / 5;`

`x %= 5;` means `x = x % 5;`

The right hand side is evaluated before the combined assignment operation is done.

`x *= a + b;` means `x = x * (a + b);`

# Working with Characters and Strings

- **char**: เก็บอักขระตัวเดียว
- **string**: เก็บสายของอักขระ

# String Input

Reading in a string object

```
string str;
```

```
cin >> str;           // Reads in a string  
                      // with no blanks
```

# String Operators

= การกำหนดค่าของ string

**string words;**

**words = "Tasty ";**

+ Joins two strings together

**string s1 = "hot", s2 = "dog";**

**string food = s1 + s2; // food = "hotdog"**

+= Concatenates a string onto the end of another one

**words += food; // words now = "Tasty hotdog"**

# string Member Functions

- **length()** – จำนวนอักขระใน string

```
string firstPrez="George Washington";
int size=firstPrez.length(); // size is 17
```

- **assign()** – ใช้เมื่อต้องการกำหนดอักขระซ้ำๆ

```
string equals;
equals.assign(80, '=');
cout << equals << endl;
```

```
#include <iostream>

using namespace std;

const int NUMBER = 12;

int main()
{
    int firstNum;
    int secondNum;

    firstNum = 18;
    cout << "Line 9: firstNum = " << firstNum
        << endl;

    cout << "Line 10: Enter an integer: ";
    cin >> secondNum;
    cout << endl;

    cout << "Line 13: secondNum = " << secondNum
        << endl;

    firstNum = firstNum + NUMBER + 2 * secondNum;

    cout << "Line 15: The new value of "
        << "firstNum = " << firstNum << endl;

    return 0;
}
```

## Exercise 1

```
using namespace std;

    //Named constants
const double CENTIMETERS_PER_INCH = 2.54;
const int INCHES_PER_FOOT = 12;

int main ()
{
    //Declare variables
    int feet, inches;
    int totalInches;
    double centimeter;

    //Statements: Step 1 - Step 7
    cout << "Enter two integers, one for feet and "
        << "one for inches: ";
    cin >> feet >> inches;
    cout << endl;
    cout << "The numbers you entered are " << feet
        << " for feet and " << inches
        << " for inches. " << endl;

    totalInches = INCHES_PER_FOOT * feet + inches;

    cout << "The total number of inches = "
        << totalInches << endl;

    centimeter = CENTIMETERS_PER_INCH * totalInches;

    cout << "The number of centimeters = "
        << centimeter << endl;

    return 0;
}
```

## Exercise 2

```
#include<iostream>
using namespace std;
const int NUMBER = 12;
int main()
{
    int firstNum;
    int secondNum;
    firstNum = 18;
    cout<<"Firstnum = "<<firstNum<<endl;
    cout<<"Enter an integer : ";
    cin >> secondNum;
    cout<<"Secondnum = " <<secondNum << endl;
    firstNum = firstNum+NUMBER+2*secondNum;
    cout<<"The new value of ";
    firstNum = firstNum+NUMBER+2*secondNum;
    cout<< "The new value of "
        <<"Firstnum = "<< firstNum << endl;
    return 0;
}
```

## Exercise 3

Firstnum = 18  
Enter an integer : 2  
Secondnum = 2  
The new value of The new value of Firstnum = 50

```
/* C++ Program to Calculate Multiplication of two Numbers */
```

```
Enter 1st number :: 23
```

```
Enter 2nd number :: 12
```

```
Product of Two Numbers [ 23 * 12 ] = 276
```

## Exercise 4

```
/* C++ Program to Find Sum and Average of three numbers */
```

## Exercise 5

```
Enter 1st number :: 12
```

```
Enter 2nd number :: 22
```

```
Enter 3rd number :: 33
```

```
The SUM of 3 Numbers [ 12 + 22 + 33 ] = 67
```

```
The AVERAGE of 3 Numbers [ 12, 22, 33 ] = 22.3333
```

## Exercise 6

Enter two integers, one for feet and one for inches: 6 8

The numbers you entered are 6 for feet and 8 for inches.

The total number of inches = 80

The number of centimeters = 203.2



# ► ARRAY

# Arrays Hold Multiple Values

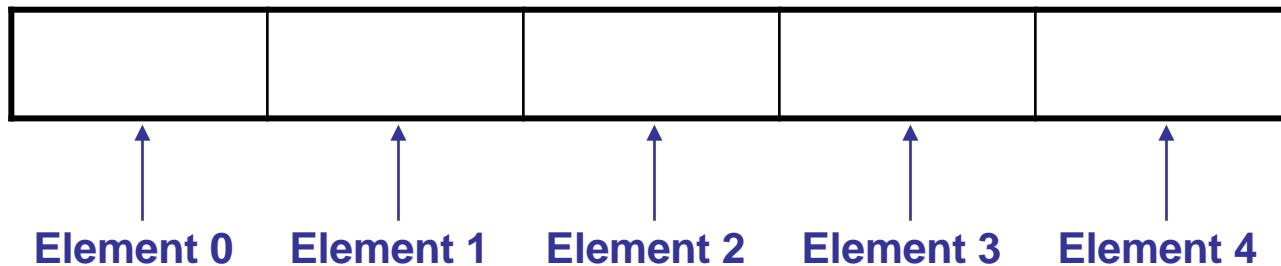
- Array: ตัวแปรที่เก็บค่าประเภทเดียวกันได้หลายค่า ค่าจะถูกเก็บไว้ในตำแหน่งหน่วยความจำที่ต่อเนื่องกันประกาศโดยใช้ตัวดำเนินการ [ ]
- `const int ISIZE = 5; int tests[ISIZE];`

# Array Storage in Memory

ถ้าให้

```
int tests[ISIZE]; // ISIZE is 5
```

ค่าจะถูกจัดสรรลงหน่วยความจำดังนี้



# Array Terminology

ถ้าให้ **int tests[ISIZE];**

- Int คือชนิดของข้อมูลที่จะเป็นสมาชิกของ array
- tests คือ name ของ array
- ISIZE ใน [ISIZE] คือขนาดของ array ที่ถูกประกาศ แสดงถึงจำนวนสมาชิกใน array
- โดย size ของ array คือ จำนวน byte ที่ถูกจัดสรรให้  
*(number of elements) \* (bytes needed for each element)*

# Array Terminology Examples

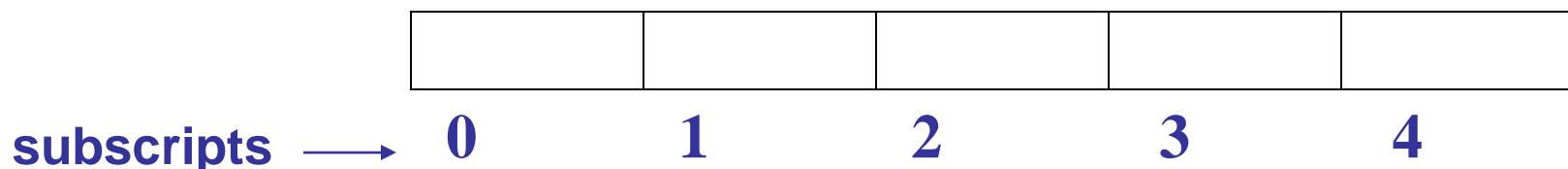
Examples:

สมมติให้ int ใช้ 4 bytes และ double ใช้ 8 bytes

```
const int ISIZE = 5, DSIZE = 10;  
  
int tests[ISIZE]; // holds 5 ints, array  
                  // occupies 20 bytes  
  
double volumes[DSIZE]; // holds 10 doubles,  
                      // array occupies  
                      // 80 bytes
```

# Accessing Array Elements

- แต่ละสมาชิกของ array จะมี subscript หรือ index เป็นตัวเข้าถึงสมาชิก
- โดย subscript จะเริ่มต้นที่ 0



# Accessing Array Elements

สมาชิกของ Array (จะถูกเข้าถึงโดย array name และ subscript)

tests



0                  1                  2                  3                  4

```
tests[0] = 79;  
cout << tests[0];  
cin >> tests[1];  
tests[4] = tests[0] + tests[1];
```

# Inputting and Displaying Array Contents

cout และ cin สามารถนำมาใช้ในการแสดงผลและเก็บค่าของ Array ได้

```
const int ISIZE = 5;
```

```
int tests[ISIZE]; // Define 5-elt. array cout <<  
"Enter first test score "; cin>> tests[0];
```

# Array Subscripts

- Array subscript สามารถเป็น integer constant, integer variable, หรือ integer expression
- Examples:

| <u>Subscript is</u>                    |                |
|----------------------------------------|----------------|
| <code>cin &gt;&gt; tests[3];</code>    | int constant   |
| <code>cout &lt;&lt; tests[i];</code>   | int variable   |
| <code>cout &lt;&lt; tests[i+j];</code> | int expression |

# Accessing All Array Elements

การเข้าถึงสมาชิกใน array

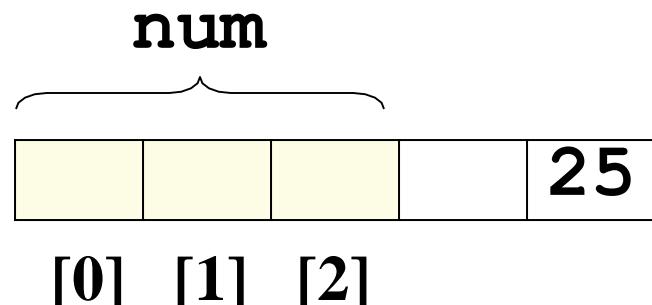
- ใช้ loop
- ให้ loop ควบคุมตัวแปรที่เป็น array subscript

```
for (i = 0; i < 5; i++)  
    cout << tests[i] << endl;
```

# No Bounds Checking

- ใน C++ จะไม่มีการตรวจเช็ค array subscript ที่ถูกต้อง
- ถ้ามีการใช้ array subscript ที่ผิด สามารถเขียนโปรแกรมทับลงในหน่วยความจำได้เลย
- Example:

```
const int ISIZE = 3;  
int i = 4;  
int num[ISIZE];  
num[i] = 25;
```



# Array Initialization

- สามารถเริ่มต้นในระหว่างการดำเนินการโปรแกรมได้

```
tests[0] = 79;  
tests[1] = 82; // etc.
```

- สามารถเริ่มต้นได้ที่การกำหนดอาร์เรย์ตั้งแต่เริ่มต้น
- `const int ISIZE = 5;`  
`int tests[ISIZE] = {79,82,91,77,84};`

# Partial Array Initialization

- การกำหนดค่า array เริ่มต้น ถ้ามีการกำหนดค่าน้อยกว่าขนาดที่ประกาศไว้ใน array ค่าスマชิกที่เหลือจะถูกตั้งค่าให้เป็น 0 หรือ เป็น empty string

```
int tests[ISIZE] = {79, 82};
```

|    |    |   |   |   |
|----|----|---|---|---|
| 79 | 82 | 0 | 0 | 0 |
|----|----|---|---|---|

# Implicit Array Sizing

- สามารถกำหนดขนาดของ array ได้โดยการใช้ขนาดของรายการข้อมูลเริ่มต้น

```
short quizzes[] = {12, 17, 15, 11};
```

|    |    |    |    |
|----|----|----|----|
| 12 | 17 | 15 | 11 |
|----|----|----|----|

# Using Increment and Decrement Operators with Array Elements

When using `++` and `--` operators, don't confuse the element with the subscript

```
tests[i]++; // adds 1 to tests[i]
tests[i++]; // increments i, but has
            // no effect on tests
```

# Copying One Array to Another

- ไม่สามารถกำหนดค่าสั่งแบบนี้เพื่อ copy array ได้

```
tests2 = tests; //won't work
```
- ต้องใช้ loop ในการ copy ค่าของสมาชิก แบบ element-by-element:

```
for (int idx=0; idx < ISIZE; idx++)  
    tests2[idx] = tests[idx];
```

# Sum, Average of Array Elements

- ใช้ loop เพื่อ add ค่าภายใน array element ไว้ด้วยกัน

```
float average, sum = 0;
```

```
for (int tnum=0; tnum< ISIZE; tnum++)
```

```
    sum += tests[tnum];
```

- หารแล้วจากได้ค่า sum, คำนวนค่า average

```
average = sum/ISIZE;
```

# Largest Array Element

- ใช้ loop ในการตรวจสอบเพื่อหา ค่า largest element
- (i.e., one with the largest value)

```
int largest = tests[0];
for (int tnum = 1; tnum < ISIZE; tnum++)
{   if (tests[tnum] > largest)
    largest = tests[tnum];
}
cout << "Highest score is " << largest;
```

- smallest element ทำเหมือนกัน

# C-Strings and **string** Objects

```
string city;  
cout << "Enter city name: ";  
cin >> city;
```

|     |     |     |     |     |
|-----|-----|-----|-----|-----|
| 'S' | 'a' | 'l' | 'e' | 'm' |
|-----|-----|-----|-----|-----|

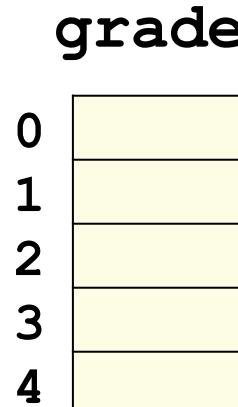
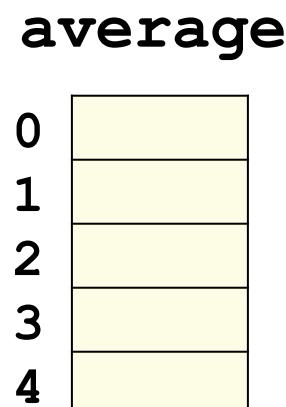
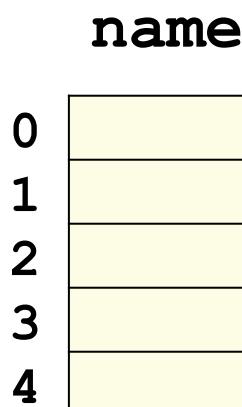
city[0] city[1] city[2] city[3] city[4]

# Using Parallel Arrays

- Parallel arrays: array ตั้งแต่สองอันขึ้นไปที่เก็บค่าที่สัมพันธ์กัน
- Subscript เพื่อเข้ามายอิง arrays
  - elements ที่มี subscript เดียวกันจะสัมพันธ์กัน
- โดยอาร์เรย์แต่ละอันไม่จำเป็นต้องเก็บข้อมูลประเภทเดียวกัน

# Parallel Array Example

```
const int ISIZE = 5;  
string name[ISIZE];    // student name  
float average[ISIZE]; // course average  
char grade[ISIZE];    // course grade
```



# Parallel Array Processing

```
const int ISIZE = 5;  
  
string name[ISIZE];    // student name  
float average[ISIZE]; // course average  
char grade[ISIZE];    // course grade  
  
...  
  
for (int i = 0; i < ISIZE; i++)  
    cout << " Student: " << name[i]  
        << " Average: " << average[i]  
        << " Grade: "    << grade[i]  
        << endl;
```

# 2D Array Traversal

- Use nested loops, one for row and one for column, to visit each array element.
- Accumulators can be used to sum the elements row-by-row, column-by-column, or over the entire array.

# 2D Array Traversal

- To initialize
  - Default of 0
  - Initializers grouped by row in braces

```
int b[ 2 ][ 2 ] = { { 1, 2 }, { 3, 4 } };
```

Row 0              Row 1

|   |   |
|---|---|
| 1 | 2 |
| 3 | 4 |

```
int b[ 2 ][ 2 ] = { { 1 }, { 3, 4 } };
```

|   |   |
|---|---|
| 1 | 0 |
| 3 | 4 |

# 2D Array Traversal

- Referenced like normal

```
cout << b[ 0 ][ 1 ];
```

- Outputs 0
- Cannot reference using commas

```
cout << b[ 0 , 1 ];
```

- Syntax error

|   |   |
|---|---|
| 1 | 0 |
| 3 | 4 |

- Function prototypes

- Must specify sizes of subscripts
  - First subscript not necessary, as with single-scripted arrays
- **void printArray( int [] [ 3 ] );**

## Exercise 7

```
// Fig. 4.22: fig04_22.cpp
// Initializing multidimensional arrays.

#include <iostream>

using std::cout;
using std::endl;

void printArray( int [][] [ 3 ] );

int main()
{
    int array1[ 2 ][ 3 ] = { { 1, 2, 3 }, { 4, 5, 6 } };
    int array2[ 2 ][ 3 ] = { 1, 2, 3, 4, 5 };
    int array3[ 2 ][ 3 ] = { { 1, 2 }, { 4 } };

    cout << "Values in array1 by row are:" << endl;
    printArray( array1 );

    cout << "Values in array2 by row are:" << endl;
    printArray( array2 );

    cout << "Values in array3 by row are:" << endl;
    printArray( array3 );

    return 0; // indicates successful termination
} // end main
```

Note the format of the prototype.

Note the various initialization styles. The elements in **array2** are assigned to the first row and then the second.

```
// function to output array with two rows
void printArray( int a[][][ 3 ] )
{
    for ( int i = 0; i < 2; i++ ) {      // f
        for ( int j = 0; j < 3; j++ )     // output column values
            cout << a[ i ][ j ] << ' ';

        cout << endl; // start new line of output
    } // end outer for structure
} // end function printArray
```

For loops are often used to iterate through arrays. Nested loops are helpful with multiple-subscripted arrays.

# Program to print one dimensional array

## Exercise 8

How Many Elements You Want to Store  
into an Array?

4

Enter 4 Elements to Store into an Array :

3

4

5

6

The Elements in the Array are :

3    4    5    6

# Program to calculate arithmetic mean of numbers

## Exercise 9

How Many Numbers You Want to Enter?

4

Enter 4 Numbers :

5

6

7

8

Arithmetic Mean = 6

# Calculate average and percentage marks of a student

## Exercise 10

Enter Marks of Student

Thai : 89

English : 76

Math : 56

History : 98

Science : 68

Average Marks = 77.4

Percentage = 77.4 %

# Calculate grade of a student on the basis of his/her total marks

## Exercise 11

```
Enter Marks of Student
```

```
-----  
Thai : 60
```

```
English : 70
```

```
Maths : 45
```

```
History : 68
```

```
Science : 40
```

```
-----  
Total Marks of Student = 283
```

```
Average = 56.6
```

```
Grade = C
```

# Search an element in array

## Exercise 12

```
Enter Array Size : 4
```

```
Enter Array Elements :
```

```
34
```

```
56
```

```
78
```

```
90
```

```
Enter Element to be Searched : 56
```

```
Element 56 Found At Position 2
```



# Pointer

# Pointers and the Address Operator

- แต่ละตัวแปรในโปรแกรมจะถูกเก็บไว้ในหน่วยความจำที่ถูกอ้างอิงตำแหน่งโดย Address
- ใช้ operator & เพื่อให้ได้ Address ของตัวแปร
- ```
int num = -23;  
cout << &num; // prints address  
// in hexadecimal
```
- โดย address ของ memory location ถูกเข้าโดยตัวแปร pointer

# Pointer Variables

- Definition:

```
int      *intptr;      //ตัวแปร pointer
```

“intptr can hold the address of an int” or “the variable that intptr points to has type int” สามารถเก็บที่อยู่ของตัวแปรที่เป็น integer ได้ หรือ ตัวแปร intptr ซึ่งไปที่ตัวแปรประเภท int นั่นเอง

- ช่องว่างในการเว้นวรคไม่สำคัญ:

```
int * intptr; int* intptr;
```

- \* เรียกว่าเครื่องหมาย indirection operator

# Pointer Variables

- Assignment:  
`int num = 25; int *intptr; intptr = &num;`



- Memory layout:

**address of num: 0x4a00**

- สามารถเข้าถึง num ได้โดยการใช้ intptr และ indirection operator \*:

```
cout << intptr;           // prints 0x4a00
cout << *intptr;          // prints 25
*intptr = 20;             // puts 20 in num
```

# The Relationship Between Arrays and Pointers

An array name is the starting address of the array

```
int vals[] = {4, 7, 11};
```

4	7	11
---	---	----

**starting address of vals: 0x4a00**

```
cout << vals;      // displays 0x4a00
cout << vals[0]; // displays 4
```

# The Relationship Between Arrays and Pointers

- ชื่อของ array สามารถถูกใช้เป็น pointer ได้
- ```
int vals[] = {4, 7, 11};  
cout << *vals;           // displays 4
```
- pointer สามารถถูกใช้เป็นชื่อ array ได้  

```
int *valptr = vals;  
cout << valptr[1]; // displays 7
```

# Pointers in Expressions

- ถ้าให้ :

```
int vals[] = {4, 7, 11};  
int *valptr = vals;
```

- อะไรคือ **valptr + 1**?
- นั่นคือ (address in **valptr**) + (1 \* size of an **int**)

```
cout << * (valptr+1); // displays 7  
cout << * (valptr+2); // displays 11
```

# Array Access

Array elements can be accessed in many ways

| Array access method                       | Example                         |
|-------------------------------------------|---------------------------------|
| array name and [ ]                        | <code>vals[2] = 17;</code>      |
| pointer to array and [ ]                  | <code>valptr[2] = 17;</code>    |
| array name and subscript arithmetic       | <code>* (vals+2) = 17;</code>   |
| pointer to array and subscript arithmetic | <code>* (valptr+2) = 17;</code> |

# Array Access

- Array notation

`vals[i]`

is equivalent to the pointer notation

`* (vals + i)`

- No bounds checking is performed on array access

# Pointer Arithmetic

Some arithmetic operators can be used with pointers:

- Increment and decrement operators `++`, `--`
- Integers can be added to or subtracted from pointers using the operators `+`, `-`, `+=`, and `-=`
- One pointer can be subtracted from another by using the subtraction operator `-`

# Pointer Arithmetic

Assume the variable definitions

```
int vals[] = {4, 7, 11};  
int *valptr = vals;
```

Examples of use of ++ and --

```
valptr++; // points at 7  
valptr--; // now points at 4
```

# More on Pointer Arithmetic

Assume the variable definitions:

```
int vals[] = {4, 7, 11};  
int *valptr = vals;
```

Example of the use of + to add an int to a pointer:

```
cout << *(valptr + 2)
```

This statement will print 11

# More on Pointer Arithmetic

Assume the variable definitions:

```
int vals[] = {4, 7, 11};  
int *valptr = vals;
```

Example of use of +=:

```
valptr = vals; // points at 4  
valptr += 2; // points at 11
```

# Initializing Pointers

- Can initialize to NULL or 0 (zero)  
`int *ptr = NULL;`
- Can initialize to addresses of other variables  
`int num, *numPtr = &num;`  
`int val[ISIZE], *valptr = val;`
- Initial value must have correct type  
`float cost;`  
`int *ptr = &cost; // won't work`

# Comparing Pointers

```
if (ptr1 == ptr2)      // compares
                      // addresses
if (*ptr1 == *ptr2)  // compares
                      // contents
```

# Exercise 13

```
#include <iostream>
using namespace std;
int main() {
    int x = 27;
    int *ip;
    ip = &x;
    cout << "Value of x is : ";
    cout << x << endl;
    cout << "Value of ip is : ";
    cout << ip << endl;
    cout << "Value of *ip is : ";
    cout << *ip << endl;
    return 0;
}
```

```
Value of x is : 27
Value of ip is : 0x5ffe94
Value of *ip is : 27
```

# Exercise 14

```
#include <iostream>
using namespace std;
int main() {
    int *ip;
    int arr[] = { 10, 34, 13, 76, 5, 46
};
    ip = arr;
    for (int x = 0; x < 6; x++) {
        cout << *ip << endl;
        ip++;
    }
    return 0;
}
```

10  
34  
13  
76  
5  
46

# Exercise 15

```
#include <iostream>
using namespace std;
void test(int*, int*);
int main() {
    int a = 5, b = 5;
    cout << "Before changing:" <<
endl;
    cout << "a = " << a << endl;
    cout << "b = " << b << endl;

    test(&a, &b);

    cout << "\nAfter changing" <<
endl;
    cout << "a = " << a << endl;
    cout << "b = " << b << endl;
    return 0;
}
```

Before changing:

a = 5

b = 5

After changing

a = 10

b = 11

```
#include <iostream>
using namespace std;
void geeks(){
    int var = 20;
    int *ptr, *ptr2;
    int val[3] = {5,9,12};
    ptr = &var;
    ptr2 = val;
    cout << ("Value of ptr = ") <<
ptr << endl;
    cout << ("Value of var = ") <<
var << endl;
    cout << ("Value of *ptr = ")
<< *ptr << endl;
    cout << "Element of the array
are : ";
    cout << ptr2[0] << " " <<
ptr2[1] << " " << ptr2[2];
}
int main()
{
    geeks();
    return 0;
}
```

## Exercise 16

```
Value of ptr = 0x5ffe5c
Value of var = 20
Value of *ptr = 20
Element of the array are : 5 9 12
```

# Vectors

- Holds a set of elements, like an array
- Flexible number of elements - can grow and shrink
  - No need to specify size when defined
  - Automatically adds more space as needed
- Defined in the Standard Template Library (STL)
  - Covered in a later chapter
- Must include **vector** header file to use vectors

```
#include <vector>
```

# Vectors

- Can hold values of any type
  - Type is specified when a vector is defined

```
vector<int> scores;
```

```
vector<double> volumes;
```
- Can use [] to access elements

# Defining Vectors

- Define a vector of integers (starts with 0 elements)  
`vector<int> scores;`
- Define `int` vector with initial size 30 elements  
`vector<int> scores(30);`
- Define 20-element `int` vector and initialize all elements to 0  
`vector<int> scores(20, 0);`
- Define `int` vector initialized to size and contents of vector `finals`  
`vector<int> scores(finals);`

# Growing a Vector's Size

- Use **push\_back** member function to add an element to a full array or to an array that had no defined size

```
// Add a new element holding a 75
scores.push_back(75);
```

- Use **size** member function to determine number of elements currently in a vector  
`howbig = scores.size();`

# Removing Vector Elements

- Use **pop\_back** member function to remove last element from vector

```
scores.pop_back();
```

- To remove all contents of vector, use **clear** member function

```
scores.clear();
```

- To determine if vector is empty, use **empty** member function

```
while (!scores.empty()) . . .
```

## Exercise 17

```
#include <iostream>
#include <vector>
using namespace std;
int main() {
    // initializer list
    vector<int> vector1 = {1, 2, 3, 4, 5};

    // uniform initialization
    vector<int> vector2{6, 7, 8, 9, 10};
    // method 3
    vector<int> vector3(5, 12);
    cout << "vector1 = ";
    // ranged loop
    for (const int& i : vector1) {
        cout << i << " ";
    }
    cout << "\nvector2 = ";
    // ranged loop
    for (const int& i : vector2) {
        cout << i << " ";
    }
    cout << "\nvector3 = ";
    // ranged loop
    for (int i : vector3) {
        cout << i << " ";
    }
    return 0;
}
```

```
vector1 = 1 2 3 4 5
vector2 = 6 7 8 9 10
vector3 = 12 12 12 12 12
```

```
#include <iostream>
#include <vector>
using namespace std;
int main()
{
    vector<int> num {1, 2, 3, 4, 5};
    cout << "Initial Vector: ";
    for (const int& i : num) {
        cout << i << " ";
    }
    // add the integers 6 and 7 to the
    vector
    num.push_back(6);
    num.push_back(7);
    cout << "\nUpdated Vector: ";
    for (const int& i : num) {
        cout << i << " ";
    }
    return 0;
}
```

## Add Elements to a Vector

### Exercise 18

Initial Vector: 1 2 3 4 5  
Updated Vector: 1 2 3 4 5 6 7

```
#include <iostream>
#include <vector>
using namespace std;
```

## Access Elements of a Vector

### Exercise 19

```
int main() {
    vector<int> num {1, 2, 3, 4, 5};

    cout << "Element at Index 0: " << num.at(0) <<
endl;
    cout << "Element at Index 2: " << num.at(2) <<
endl;
    cout << "Element at Index 4: " << num.at(4);

    return 0;
}
```

Element at Index 0: 1  
Element at Index 2: 3  
Element at Index 4: 5

```
#include <iostream>
#include <vector>
using namespace std;
int main()
{
    vector<int> num {1, 2, 3, 4, 5};
    cout << "Initial Vector: ";
    for (const int& i : num) {
        cout << i << " ";
    }
    // change elements at indexes 1 and 4
    num.at(1) = 9;
    num.at(4) = 7;
    cout << "\nUpdated Vector: ";
    for (const int& i : num) {
        cout << i << " ";
    }
    return 0;
}
```

## Change Vector Element

### Exercise 20

Initial Vector: 1 2 3 4 5  
Updated Vector: 1 9 3 4 7

```
#include <iostream>
#include <vector>
using namespace std;
int main()
{
```

```
    vector<int> prime_numbers{2, 3, 5, 7};
    // initial vector
    cout << "Initial Vector: ";
    for (int i : prime_numbers) {
        cout << i << " ";
    }
    // remove the last element
    prime_numbers.pop_back();
    // final vector
    cout << "\nUpdated Vector: ";
    for (int i : prime_numbers) {
        cout << i << " ";
    }
    return 0;
}
```

## Delete Elements from C++ Vectors Exercise 21

Initial Vector: 2 3 5 7  
Updated Vector: 2 3 5

```
#include <iostream>
#include <vector>
using namespace std;
int main()
{
    vector<int> num {1, 2, 3, 4, 5};
    // declare iterator
    vector<int>::iterator iter;
    // initialize the iterator with the first element
    iter = num.begin();
    // print the vector element
    cout << "num[0] = " << *iter << endl;
    // iterator points to the 3rd element
    iter = num.begin() + 2;
    cout << "num[2] = " << *iter;
    // iterator points to the last element
    iter = num.end() - 1;
    cout << "num[4] = " << *iter;
    return 0;
}
```

## C++ Vector Iterators

### Exercise 22

num[0] = 1  
num[2] = 3  
num[4] = 5