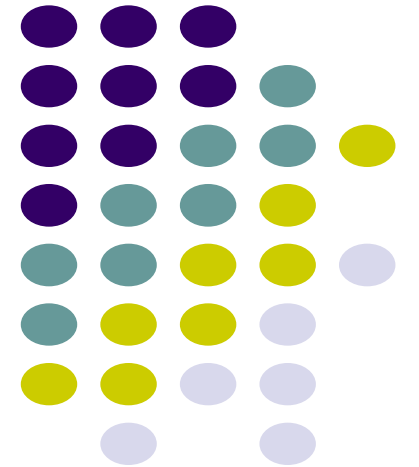


Visual Sutudio Codeによる Gitの基本操作



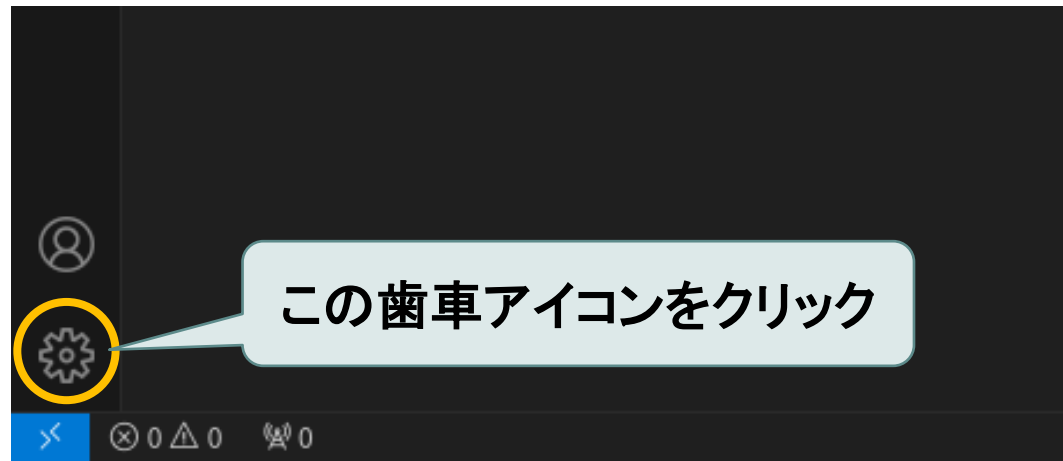
前準備

- Visual Studio Code(以下、vscode) の初期設定

- vscodeの起動

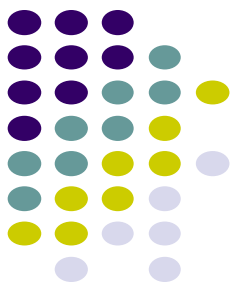
\$ code ↵

- 設定画面を開く



[設定]を選択





拡張機能の利用

- Git Graph のインストール

拡張機能 (Ctrl+Shift+X) をクリック

[git]を検索

[Git Graph]を選択し、[インストール]ボタンをクリックし、インストール

拡張機能: Git Graph - Visual Studio Code

Git Graph mhutchie.g
mhutchie | 793,348 |
View a Git Graph of your repository

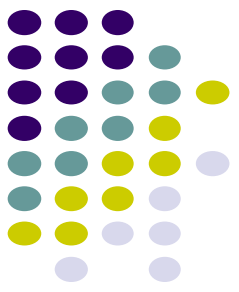
インストール

Git History 0.6.14
View git log, file history, compare branches
Don Jayamanne | インストール

GitLens — Git supercharged 11.0.2
Supercharge the Git capabilities built into Visual Studio
Eric Amodio | インストール

Git Graph 1.27.0
View a Git Graph of your repository, compare branches, and more
mhutchie | インストール

Git Blame 6.0.0
See git blame info in the editor
Wade Anderson



拡張機能の利用

- C/C++拡張機能のインストール

拡張機能 (Ctrl+Shift+X) を開く

[拡張機能]アイコンをクリック

[c]を検索

[C/C++]を選択し、[インストール]ボタンをクリックし、インストール

拡張機能: C/C++ - prog - Vi

拡張機能: マーケットプレ...

C/C++
C/C++ IntelliSense, debugging, ...
Microsoft
インストール

C/C++ Themes
UI Themes for C/C++
Microsoft
インストール

C/C++ Extension
Popular extension
Microsoft

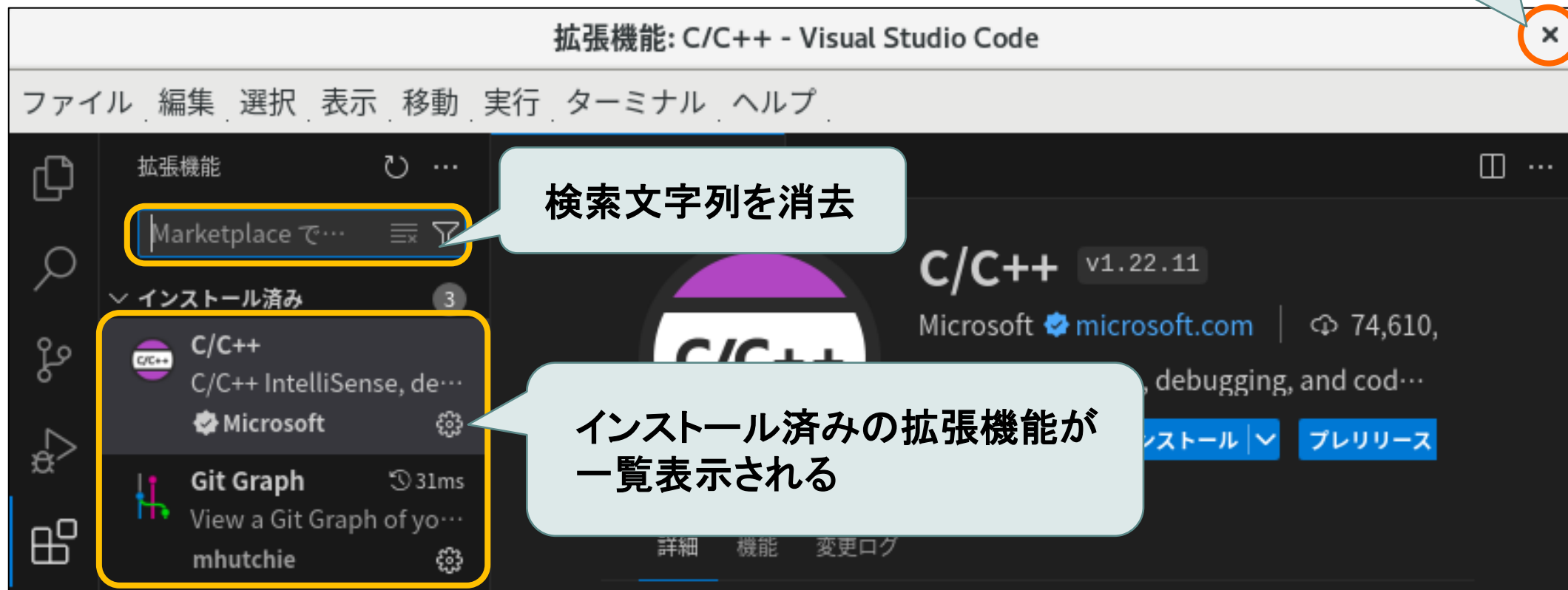
C/C++ Compile ...
Compile & Run single c/c++ fil...

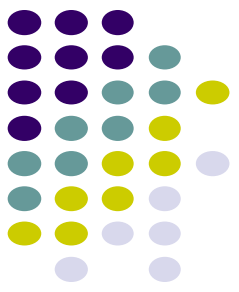
C/C++ for Visual St

拡張機能の利用

- インストールされた拡張機能の確認

インストールの確認ができれば、
一旦 VSCode を終了

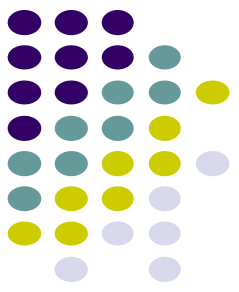




ここからは、すべて vscode 内で作業します

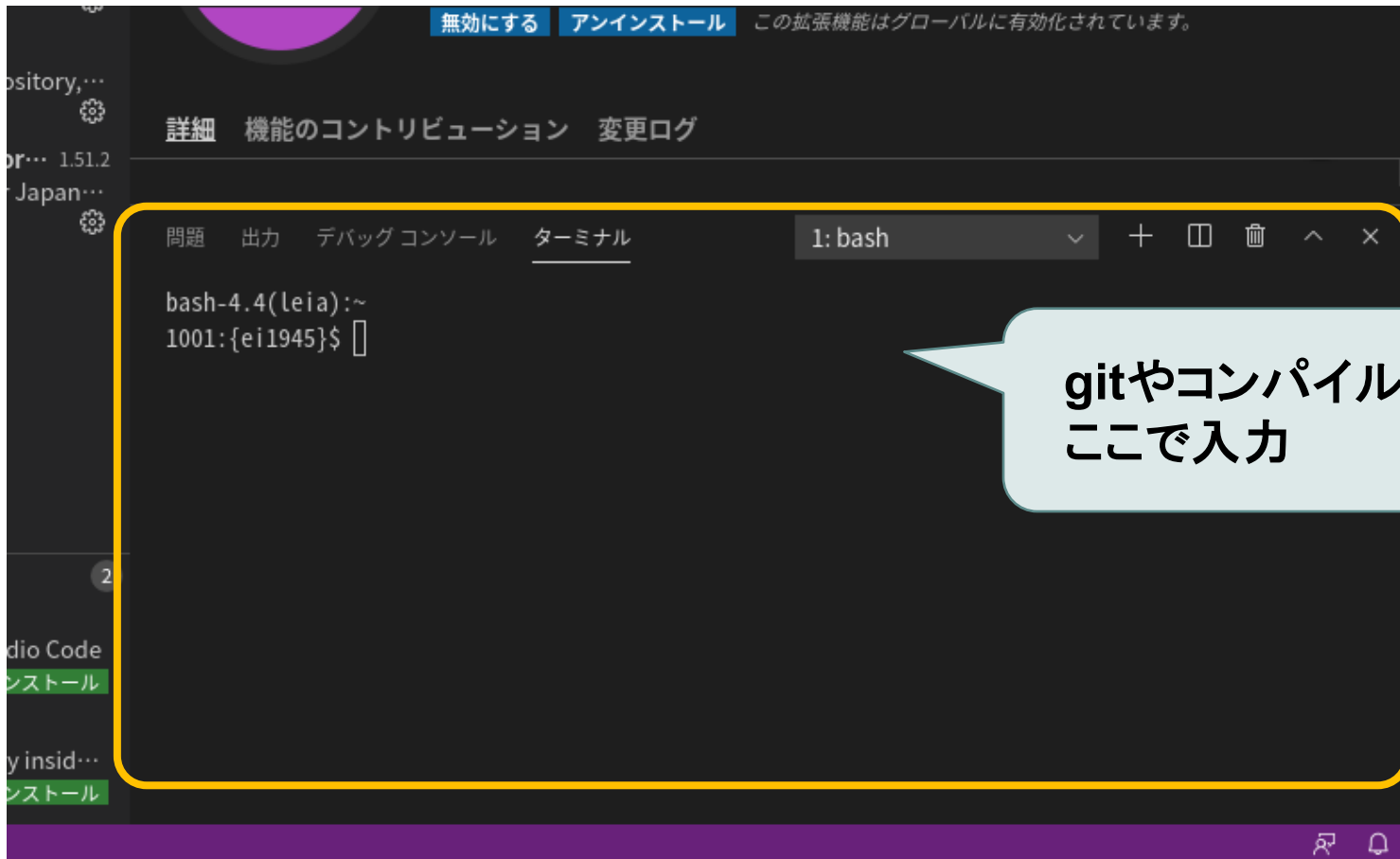
- 新しいターミナルを開きます



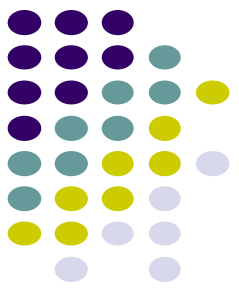


ここからは、すべて vscode 内で作業します

- Unixのコマンドは右下の[ターミナル]ペインで入力



gitやコンパイルなどのUnixコマンドは、
ここで入力

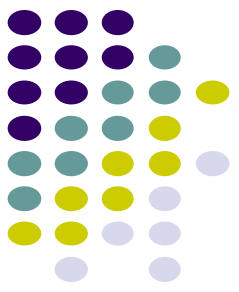


Gitを使うための準備(この作業は端末で実施)

- ユーザー名とメールアドレスの登録
 - git config --global user.name “eixxxx”
 - git config --global user.email “eixxxx@hamako-ths.ed.jp”
- pullのデフォルト動作をマージに設定
 - git config --global pull.rebase false

実行例

```
$ git config --global user.name “ei2345” ↵  
$ git config --global user.email “ei2345@hamako-ths.ed.jp” ↵  
$ git config --global pull.rebase false ↵
```

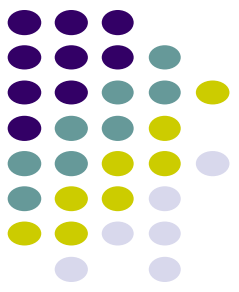



Gitを使うための準備(この作業は端末で実施)

- 登録内容の確認
 - git config --global --list

実行例

```
$ git config --global --list ↵  
user.name=ei2345  
user.email=ei2345@hamako-ths.ed.jp  
pull.rebase=false
```



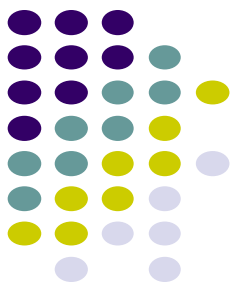
Gitを使うための準備（設定を間違えた場合）

- 以下のコマンドで登録内容を削除し、再登録
 - git config --global --unset user.name
 - git config --global --unset user.email
 - git config --global --unset pull.rebase false

実行例

```
$ git config --global --unset user.name ↵  
$ git config --global --unset user.email ↵  
$ git config --global --list ↵  
pull.rebase=false  
$
```

削除しなかった pull.rebase=false のみが表示された



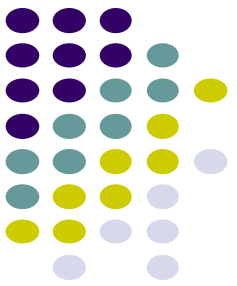
リポジトリの使用開始

メインリポジトリのクローンを作成します。

```
$ git clone -b ブランチ名 --single-branch https://github.com/Ei2437/ai-pumpkin-talk.git
```

```
$ cd ai-pumpkin-talk
```

Git操作



- 追加したいファイルを作成します。

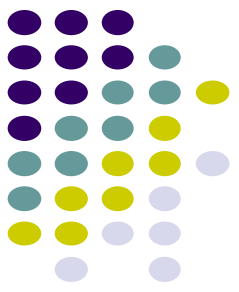
[エクスプローラー] アイコンをクリック

不要なタブは閉じてしまってよい

[新規ファイル]アイコンをクリックし、ファイル名を入力

このプログラムを入力し、
[Ctrl+S]で上書き保存

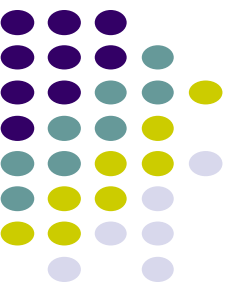
```
main.c - bookstore - Visual Studio  
ファイル(F) 編集(E) 選択(S) 表示(V) 移動(G) 実行(R) ターミナル(T) ヘルプ(H)  
エクスプローラー  
> 開いているエディター  
BOOKSTORE  
C main.c  
開始  
新しいファイル  
フォルダーを開く...  
ワークスペース  
main.c  
C main.c > main(void)  
1 #include <stdio.h>  
2  
3 int main(void)  
4 {  
5     printf("BookStore\n");  
6  
7     return(0);  
8 }
```



それでは、Git操作を始めましょう

```
C main.c  X
C main.c > main(void)
1  #include <stdio.h>
2
3  int main(void)
4  {
5      printf("BookStore\n");
6
7      return(0);
8  }
```

このプログラムを入力し、
[Ctrl+S]で上書き保存

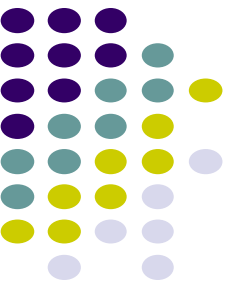


ファイルをコミット

\$ git add 編集したファイルがあるディレクトリのファイルパス
(大抵はカレントディレクトリで作業してるので、ファイルパスは . でOK?)

\$ git commit -m "何をするのかコメントを残す"

GUI 環境では、変なエラーみたいのが出てできませんでした。



ファイルをpush

\$ git push origin ブランチ名

初回時は、push時に、GitHubへのログインが求められます。
指示に従って、自分のGitHubアカウントでログインしてください。