

# Hello C 使い方

Ver.2.0.2.2

2018.12.08 update

©2018 K.Kaida

## ○基本操作

### 1. Hello C の起動

- 1) [スタート]→アプリ一覧から[Hello C]をクリックすると起動します。



- 2) MinGW についてのメッセージボックスが出た場合

→ エラー文を日本語化したい場合は[はい]

→ 何もしない場合は[いいえ] ※デフォルトで英語になります。

※「(ドライブ名):¥」へ MinGW フォルダ、及びファイルを追加します。既に存在する場合上書き追加します。既存の変更に注意してください。

### 2. Hello C の終了

- 1) メニューバーの[ファイル]をクリックします。
- 2) [終了]をクリックします。
- 3) プロジェクトの保存が必要な場合は通知されます。  
必要な操作を選択してください。

## ○プロジェクト

ソースや実行ファイルはプロジェクトごとに管理されます。

### 1. プロジェクトの作成

- 1) メニューバーの[ファイル(F)]をクリックします。
- 2) [新しいプロジェクト]をクリックします。
- 3) 作成したいプロジェクト名を入力します。
- 4) プロジェクトが作成され、コードを打ち込めるようになります。

### 2. プロジェクトを開く

- 1) メニューバーの[ファイル(F)]をクリックします。
- 2) [プロジェクト一覧]をクリックします。
- 3) 開きたいプロジェクトを選択し、ダブルクリックします。  
※選択した状態で右クリック→[開く]でも開くことができます。

### 3. プロジェクトを閉じる

- 1) メニューバーの[ファイル(F)]をクリックします。
- 2) [プロジェクトを閉じる]をクリックします。

### 4. プロジェクトの名前を変更する

- 1) メニューバーの[ファイル(F)]をクリックします。
  - 2) [プロジェクト一覧]をクリックします。
  - 3) 名前を変更したいプロジェクトを選択します。
  - 4) 右クリック→[名前の変更]をクリックします。
  - 5) 新しい名前を入力して[OK]を押します。
- ※プロジェクトは閉じておく必要があります。

### 5. プロジェクトを削除する

- 1) メニューバーの[ファイル(F)]をクリックします。
  - 2) [プロジェクト一覧]をクリックします。
  - 3) 削除したいプロジェクトを選択します。
  - 4) 右クリック→[削除]をクリックします。
  - 5) [はい]をクリックします。
- ※プロジェクトは閉じておく必要があります。

## ○ソースファイル

ソースファイルの一覧は「ファイルビュー」から見ることができます。

ファイルビューを表示するには、メニューバーの[表示]から[ファイルビュー]を選択してください。

### 1. ソースファイルを追加する

- 1) メニューバーの[ファイル(F)]をクリックします。
- 2) [追加]から、追加したいファイルの種類を選択します。
- 3) ファイル名を入力します。

### 2. ソースファイルの名前を変更する

- 1) ファイルビューを表示しておきます。(前述)
- 2) 名前を変更したいファイルを選択します。
- 3) 右クリック→[名前の変更]をクリックします。
- 4) 新しい名前を入力して[OK]を押します。

### 3. ソースファイルを削除する

- 1) ファイルビューを表示しておきます。(前述)
- 2) 削除したいファイルを選択します。

3) 右クリック→[削除]をクリックします。

4) [はい]を押します。

※ファイルがすべて削除された場合でもプロジェクトは管理されます。

## ○実行

作成したコードを実行したり、バグを見つけるデバッグを行ったりすることができます。

### 1. プログラムを実行する

1) 実行したいプロジェクトを開いておきます。

2) ツールバーの[Debug]のチェックを外します。

3) ツールバーから[実行]を押します。

4) コンパイルに成功した場合はプログラムが実行されます。

失敗した場合はエラーや警告が「エラー一覧」に表示されます。

※「エラー一覧」を表示するには、メニューバー[表示]から[エラー一覧]を選択してください。

※「コンパイルに成功したが実行ファイルが見つからない」と出た場合や「アクセスが拒否されました」と出る場合は、ウイルスセキュリティに誤検知されている可能性があります。ご自身の責任の上で、ウイルスセキュリティの除外項目に登録してください。

### 2. デバッグ機能を使用する

#### A) デバッグの開始と終了

1) 実行したいプロジェクトを開いておきます。

2) ツールバーの[Debug]にチェックをします。

3) ツールバーから[実行]を押します。

4) コンパイルに成功した場合はデバッグが開始されます。

失敗した場合はエラーや警告が「エラー一覧」に表示されます。

コードを修正し再度実行してください。

※「エラー一覧」を表示するには、メニューバー[表示]から[エラー一覧]を選択してください。

5) プログラムが停止する原因があった場合は通知されます。

→ プログラムを続行するには[続行]

→ プログラムを強制終了するには[強制終了]

→ プログラムの停止位置を確認したい場合は[閉じる]

6) プログラムが終了した時点でデバッグは終了されます。

途中で終了したい場合は、ツールバーの[■ (停止)]をクリックしてください。

#### B) ブレイクポイントの設定・解除

ブレイクポイントを設定することで、プログラムを設定場所で一時停止させ、変数の中身を確認

認することができます。

- 1) デバッグしたいプロジェクトを開いておきます。
- 2) 行番号の右側をクリックすることでブレイクポイントを設定・解除できます。

```
13 int main( void )
14 {
15     →STUDENTS st[N];
16     →int i;
17     →
18     →for(i = 0; i < N; i++)
19     →→st[i].Japanese = rand() % 100;
20     →
21     →return 0;
22 }
23
24
```

- 3) ブレイクポイントで設定された行でプログラムが止まります。
- 4) ブレイクポイントで停止後、その時点で表示できる変数が「デバッグ・変数」に表示されます。見方については後述をご覧ください。  
※「デバッグ・変数」を表示するには、ツールバー[表示]から[デバッグ・変数]を選択してください。

### C) ステップイン、ステップアウト、ステップオーバー

これらは、ブレイクポイントで停止後、1行ずつ実行したい場合に使用します。

ステップイン：

現在のステートメントを実行し次のステートメントで停止します。

関数である場合は関数の内部へ移動します。

ステップアウト：

現在のステートメントを実行し次のステートメントで停止します。

関数である場合は、その関数全体を実行したのち、呼び出し後の次のステートメントで停止します。

ステップオーバー：

現在実行している関数全体を実行したのち、呼び出し元へ移動します。

- 1) 実行したいプロジェクトを開いておきます。
- 2) ブレイクポイントを設定します。
- 3) デバッグを開始します。
- 4) ブレイクポイントで停止した時点でこれらの操作が有効になります。
- 5) ツールバーの各アイコンをクリックしてください。



左から、[ステップイン]、[ステップアウト]、[ステップオーバー]になります。

効果は前述の通りです。

#### D) デバッグ・変数の見方

ブレークポイントやステップでプログラムが一時停止した場合、自動でその時点で表示可能な変数の中身が表示されます。

※「デバッグ・変数」を表示するには、ツールバーの[表示]から[デバッグ・変数]を選択してください。

- 変数名、値、アドレスの順で表示されます。
- 配列、構造体は階層構造で表示されます。  
中身を展開したい場合は、「+」ボタンをクリックしてください。
- 表示されるたびにすべて展開しておきたい場合は、ウィンドウ左上の「すべて展開」にチェックしてください。
- 前回から値が変更された場合は背景色が変わり、「以前の値→現在の値」が表示されます。

変数名	値	アドレス	
st			<input type="checkbox"/>
[0]			<input type="checkbox"/>
[1]			<input type="checkbox"/>
[2]			<input type="checkbox"/>
[3]			<input type="checkbox"/>
[4]			<input type="checkbox"/>
[5]			<input type="checkbox"/>
[6]			<input type="checkbox"/>
[7]			<input type="checkbox"/>
[8]			<input type="checkbox"/>
[9]			<input type="checkbox"/>
Japanese	6422420 → 64	0x61ff18	<input checked="" type="checkbox"/>
Math	4200843	0x61ff1c	<input checked="" type="checkbox"/>
English	4200752	0x61ff20	<input checked="" type="checkbox"/>
Science	0	0x61ff24	<input type="checkbox"/>
History	2809856	0x61ff28	<input type="checkbox"/>
i	4194432 → 10	0x61ff2c	<input type="checkbox"/>

#### ・指定したメモリのみ表示する場合

1. 表示したいメモリの一番右にあるチェックボックスをチェックします。  
※アドレスが表示されているメモリに限ります。
2. 上にある[選択表示]ボタンをクリックすることで選択されたメモリのみ表示されます。
3. 再度全体を表示するには[全体表示]ボタンをクリックしてください。

## ○オプション

オプションでは Hello C の設定が行えます。

オプションを表示するには、ツールバーの[ツール]から[オプション]を選択してください。

### ● 全般

#### ・配色テーマ

Hello C の全体の色を決めることができます。明色は白、暗色は灰色基調になります。

#### ・プロジェクトの場所

プロジェクトを管理するフォルダを指定します。

デフォルトはドキュメントです。

#### ・学習者モード

「;」のつけ忘れの通知や、エラーウィンドウのポップアップを行います。

### ● 起動時

#### ・最大化する

ウィンドウを最大化します。

#### ・前回のレイアウト配置で開く

ソフトを再起動しても、前回のレイアウト配置で起動します。

#### ・MinGW 日本語化についてのメッセージを出す

MinGW の設定について、再度問い合わせるようにします。

### ● バージョン情報

現在の Hello C のバージョンを確認することができます。

## ○エディタ設定

エディタ設定ではエディタに関する設定が行えます。

エディタ設定を表示するには、ツールバーの[ツール]から[エディタ設定]を選択してください。

### ● 全般

#### ・ユーザー補助

名称	説明
自動インデントを行う	改行時にインデントを揃えます。
段落ガイドを表示する	段落ごとに縦線を引きます。
タブを表示する	タブコードを表示します。
改行コードを表示する	改行コードを表示します。
強調表示を行う	括弧のペアを強調表示します。
オートコンプリートを使用する	オートコンプリートを有効にします。
ソースを探索し独自定義名を登録する	ユーザーが定義した関数・定義名を登録します。
中括弧を自動で閉じる	中括弧を入力時、閉じ括弧を追加します。
中括弧内部での改行でインデントする	中括弧内部で改行時にインデントします。

- ・環境

- ・外部によるファイル内容変更時、自動再読み込みを行う

ソースファイル、テキストファイル等が外部変更、または実行されたプログラムによる変更時、自動で再読み込みを行うかを設定できます。

手動で再読み込みを行う場合は、ファイルビューから操作してください。

- インデント整形

インデント整形機能についての設定を変更できます。

- フォント

フォントの種類とサイズを変更できます。

- テーマと色

配色テーマではエディタの基調となる色を設定できます。

各文字色はテーマ別に設定できます。上記の配色テーマから変更したいテーマを選び、左側の欄からさらに変更したい種類を選択し、[変更]をクリックしてください。

## ○コンパイル設定

- 全般

- ・コンパイル時に構文チェックを行う

コンパイル時に構文チェックを行うかを設定します。

- ・実行時に”pause”コマンドを適用する

プログラム終了時にウィンドウが閉じられないようにします。

※デバッグ時は適用されません。

- ・続行確認文字を表示しない

デフォルトで「続行するには何かキーを押してください...」と出ますが、これを表示しません。

- プロジェクト設定

プロジェクトごとに設定される設定項目です。

- ・コマンドライン引数

コマンドライン引数を設定できます。

- ・C++コンパイラを使用する

試験導入された機能です。g++コンパイラを使用します。

- ・-mwindows オプションを使用する

WinAPI を使用する際に設定します。

- ・socket オプションを使用する

Socket 通信プログラム用のオプションを追加します。

- ・追加のコンパイルオプション

コンパイラに直接渡されるオプションです。誤動作の原因となります。使用には注意してください。

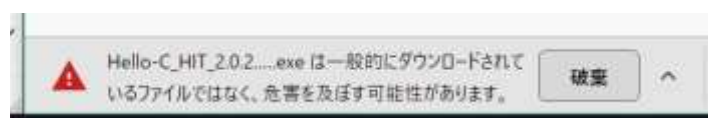
- ・追加のインクルード  
インクルードディレクトリを追加します。
- ・追加のライブラリ  
ライブラリディレクトリを追加します。

## ○Q&A

・インストール時に「Windows によって P C が保護されました」と出てインストールできない。

A. Hello C が安全であるご自身で認めた場合は「詳細情報」→「実行」をクリックしてください。

・ Chrome でダウンロード時に以下のような表示になる。



A. Hello C が安全であるご自身で認めた場合は、破棄ボタンの右側をクリックし、「継続」をクリックしてください。



・「この P C の設定では、ストアの確認済みアプリのみがインストールを許可されています」と出てインストールできない。

A. 「設定を開く」から設定を開き、アプリのインストール項目から、「ストアのアプリのみ許可」以外を選択して、再度インストールしてください。

・「承認されていない変更がブロックされました」と出てブロックされる。

A. 「Windows Defender セキュリティ センター」を開き、「ウイルスと脅威の防止」から「ウイルスと脅威の防止設定」をクリックします。「コントロールされたフォルダーアクセス」の項目を「オフ」にするか「除外の追加または削除」を選択し、該当のフォルダを追加してください。

・実行時に、ウイルス対策ソフトにウイルスと判定される。

A. Hello C が安全であるご自身で認めた場合は、ウイルス対策ソフトから、「ホワイトリストに追加」または「除外リスト」に追加してください。ウイルス対策ソフトの使い方については、各ソフトの説明書をご覧ください。



・コンパイルエラーが一部日本語ではない。

A. 本ソフトはコンパイラに gcc を使用しています。日本語訳はすべて gcc に依存しますのでご了承ください。

・作ったプロジェクトは Hello C のアンインストール時に消えますか？

A. 消えません。