

Windows で C プログラミング

URL: <https://www.kkaneko.jp/pro/cintro/index.html>

金子邦彦



トピックス

- ソースコード
- コンパイル
- `getchar()` によるキーボード読み取り

ソースコード (source code)



- プログラムを、何らかのプログラミング言語で書いたもの
- 「ソフトウェアの設計図」ということも。
人間も読み書き、編集できる

C 言語プログラムの例



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

```
int main() {
```

```
    printf("Hello, World!¥n");
```

```
    return 0;
```

```
}
```

C 言語プログラムのソースコード

A screenshot of a text editor window titled "*無題 - メモ帳". The window contains the same C program code as shown on the left. The code is:

```
#include <stdio.h>
int main() {
    printf("Hello, World!¥n");
    return 0;
}
```

エディタでソースコードを編集



ファイルに保存

コンパイル



コンパイルは、ソースコードを実行形式ファイルに変換すること。

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

```
int main() {
```

```
    printf("Hello, World!¥n");
```

```
    return 0;
```

```
}
```



実行形式ファイル

コンパイラー

C 言語プログラムのソースコード
ファイル名: **hello.c**

コンパイルの方法



- **マイクロソフト C++ ビルドツール**

- 次のページからダウンロードしてインストール

<https://visualstudio.microsoft.com/ja/visual-cpp-build-tools/>

- インストール手順は次のページで説明

<https://www.kkaneko.jp/tools/win/buildtool.html>

- **WSL 2 のコンパイラー**

- **オンライン開発環境**

C Tutor <https://pythontutor.com/c.html>

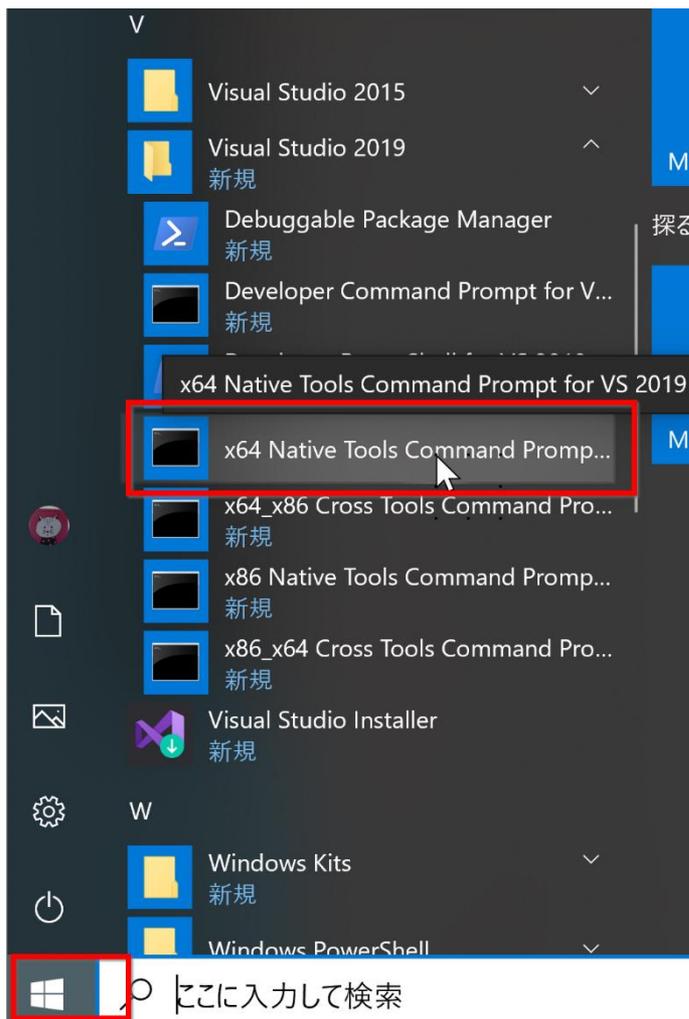
Online GDB <https://www.onlinegdb.com/>

Coding Ground

<https://www.tutorialspoint.com/codingground.htm>

など

マイクロソフト C++ ビルドツールの起動



```
C:\x64 Native Tools Command Prompt for VS 2019
*****
** Visual Studio 2019 Developer Command Prompt v16.4.5
** Copyright (c) 2019 Microsoft Corporation
*****
[vcvarsall.bat] Environment initialized for: 'x64'

C:\Program Files (x86)\Microsoft Visual Studio\2019\BuildTools>d:
D:\>cl hello.c
Microsoft(R) C/C++ Optimizing Compiler Version 19.24.28316 for x64
Copyright (C) Microsoft Corporation. All rights reserved.

hello.c
Microsoft (R) Incremental Linker Version 14.24.28316.0
Copyright (C) Microsoft Corporation. All rights reserved.

/out:hello.exe
hello.obj

D:\>
```

- コンパイルは、
cl <ソースコードのファイル名>
- 「/out:hello.exe」のように、**実行形式ファイル名**が表示される
 - コンパイルの前に **cd** コマンドで、**ソースコードのファイルのあるディレクトリに移動**

スタートメニューで、
x86 Native Tools Command Prompt
(Visual Studio 2019 の下)

getchar() で文字数を数える

- 半角 1 文字: **1 バイト**

- Enterキー: **1 バイト**

文字コードは、16進数の「0A」。C言語では '¥n'

- **getchar()** のふるまい

Enter キーを押した時点で、キーボードバッファからの読み取りが始まる。

```
#include <stdio.h>
int main() {
    int a;
    while (getchar () != '¥n') {
        a++;
    }
    printf ("%d¥n", a);
    return 0;
}
```

```
D:¥>. ¥hello.exe
abcd
4
D:¥>_
```

EOF (End of File)

- Windows では、EOF は CTRL + Z (同時押し)
- **行の先頭**で **CTRL + Z** とした場合

EOF (1バイトとはカウントされない)

- **行の先頭以外**で **CTRL + Z** とした場合

CTRL + Z は、文字コードが「1A」の1バイトのデータ
CTRL + Z 以降、当該行のすべてのデータが破棄される

```
#include <stdio.h>
int main() {
    int a;
    while(getchar() != EOF) {
        a++;
    }
    printf("%d\n", a);
    return 0;
}
```

```
D:\>. \hello.exe
abcd
^Z
5
D:\>
```

a b c d Enter CTRL+Z Enter
5 バイト EOF

```
D:\>. \hello.exe
abc^Zd
^Z
4
D:\>
```

a b c CTRL+Z d Enter CTRL+Z Enter
4 バイト EOF 9

ここで学んだこと



- while による繰り返し
- 半角文字は1バイト
- Enterキー: **1バイト**. C言語では '**\n**'
- **EOF** (ファイルの終わり) : Windows では CTRL + Z

※ 学習のため, `getchar()` を用いた. キーボードからの文字列読み取りは, `getchar()` でなく, 扱いが簡単な, `fgets()` を推奨する.