

JDK 18以降のインストール／デスクトップ編 (Windows 10/11 (64 bit))

「見ひらきで学べるJavaプログラミング」補足資料
(C) 2019 古井陽之助, 神屋郁子, 下川俊彦, 合志和晃.

<https://mihirakijava.github.io/support/>

2023/09版

この資料の使用について

- 本資料は下記書籍の補足資料です。
 - 「見ひらきで学べるJavaプログラミング」, 近代科学社, (2019).
- 本資料の著作権は著者が所有します。
 - ただし、引用されている著作物の著作権はその著作権者のものです。
- 本資料の改変・配布は、学校・企業等の団体内部での利用に限り可能です。
 - ただし、団体外部への配布は禁止します。

改版履歴

- 2019/08/31: 2019/08版。
- 2021/02/16: 軽微な更新。
- 2022/02/11: 軽微な更新。
- 2022/03/31: 2022/03版。
 - JDK 18から既定の文字コードがUTF-8になったことを反映。
- 2023/04/03: 軽微な更新。
- 2023/09/23: 「圧縮ファイルを展開(続き)」(p.9)右にフキダシ1個追加。

はじめに

この資料の概要

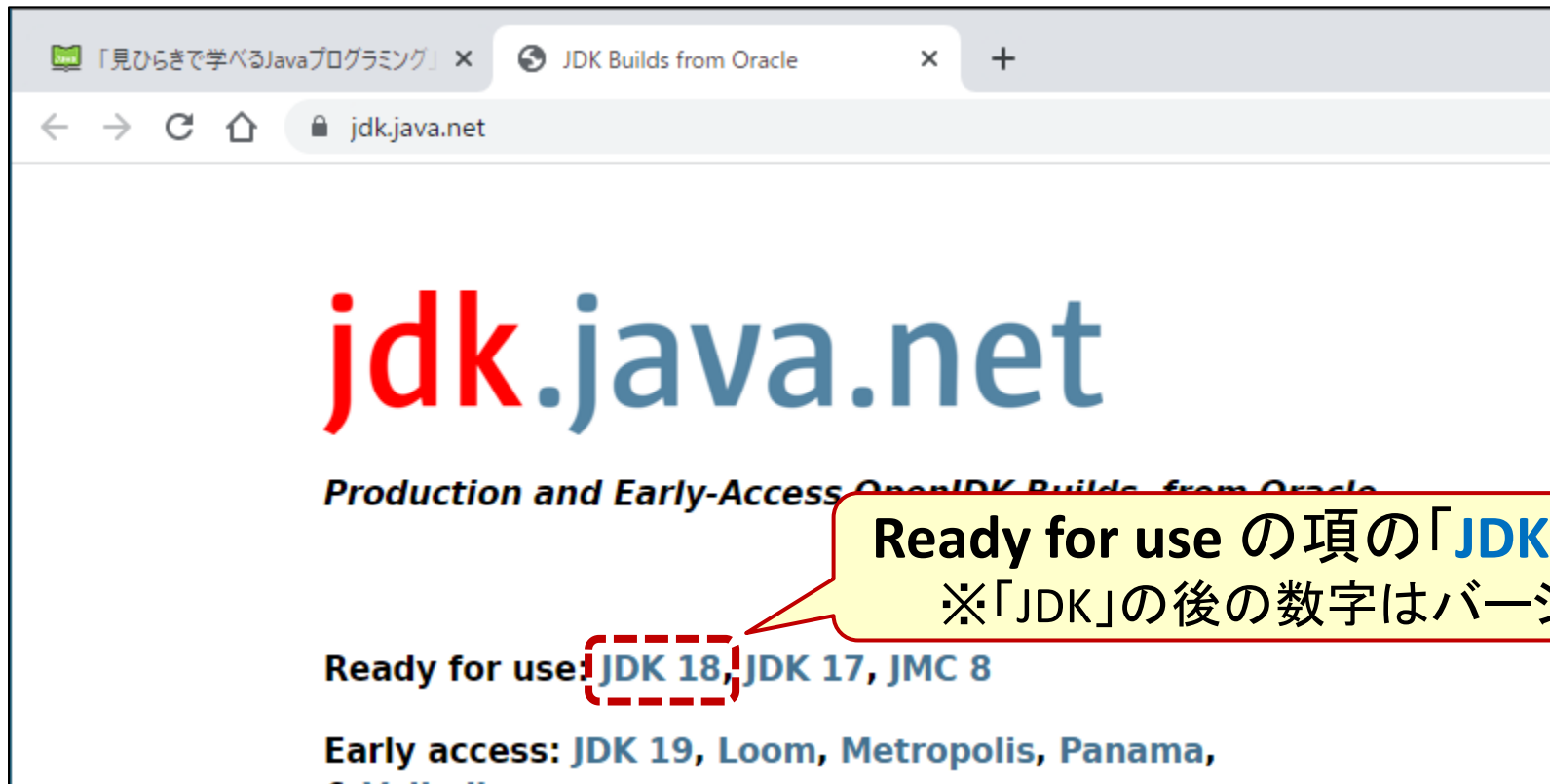
- Java Development Kit (JDK)のインストール方法など
 - Javaプログラミングのための基本的な開発環境
- 前提
 - オペレーティングシステムとして Windows 10/11 (64 bit) を使用
 - **JDK 18 以降の版を使用**
 - 必要に応じて「デスクトップ」を「ダウンロード」などに読み換えること
- 注意事項
 - この資料中の画面は Windows 10 Home (64 bit) のもの
 - ソフトウェアのアップデート等に伴い手順や画面は変わることがある

JDKのインストール

JDKのインストール

(1) 公式サイトにアクセス

- <https://jdk.java.net/> にアクセス



JDKのインストール

(2) ダウンロード

- Windows x64用のZIP形式ファイル(圧縮ファイル)をダウンロード

jdk.java.net

OpenJDK JDK 18 General-Availability Release

This page provides production-ready open-source builds of the Java SE 18 Platform under the General Public License, version 2, with the Classpath Exception.

Commercial builds of JDK 18 from Oracle, under a non-open-source license, can be found at the Oracle Technology Network.

Documentation

- Features
- Release notes
- API Javadoc

Builds

Linux/AArch64	tar.gz (sha256)	186983593 bytes
Linux/x64	tar.gz (sha256)	188173501
macOS/AArch64	tar.gz (sha256)	183221810
macOS/x64	tar.gz (sha256)	
Windows/x64	zip	

「zip」をクリック

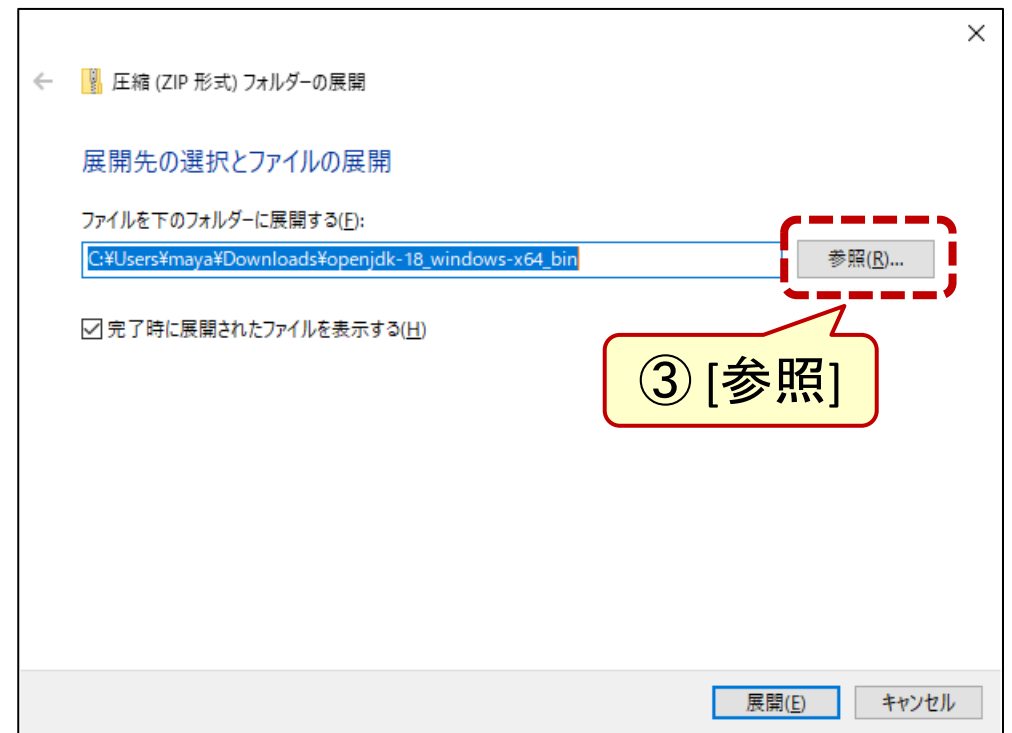
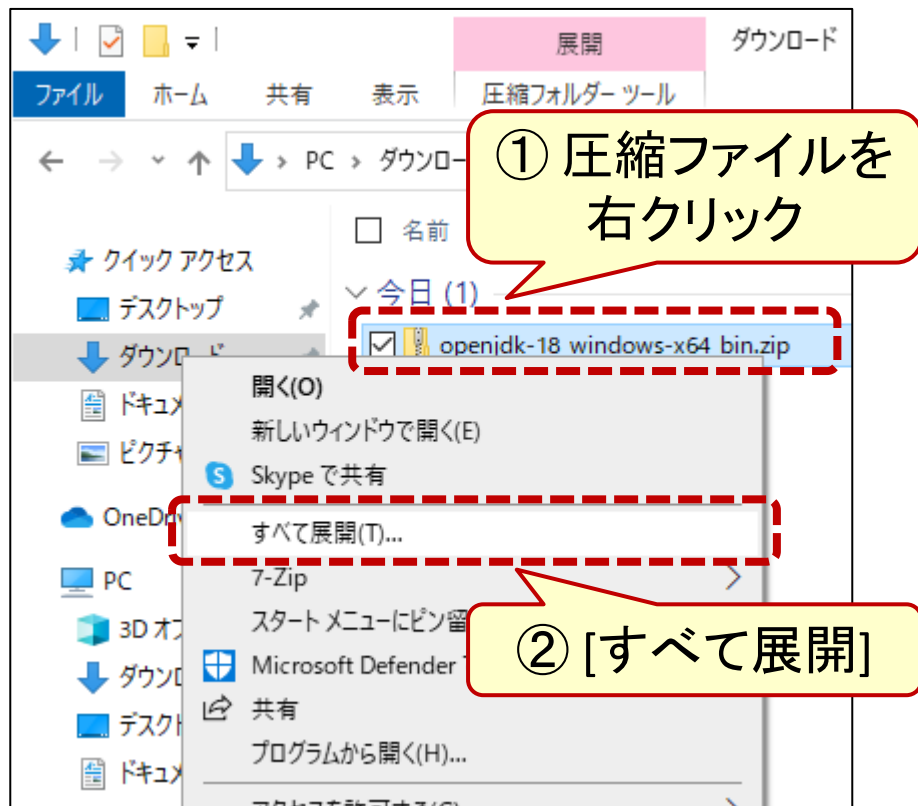
画面表示は版によって異なる。
これはJDK 18の例

※オラクルのJDKサイト <https://jdk.java.net/18/> (2022/03/23閲覧)

JDKのインストール

(3) 圧縮ファイルを展開

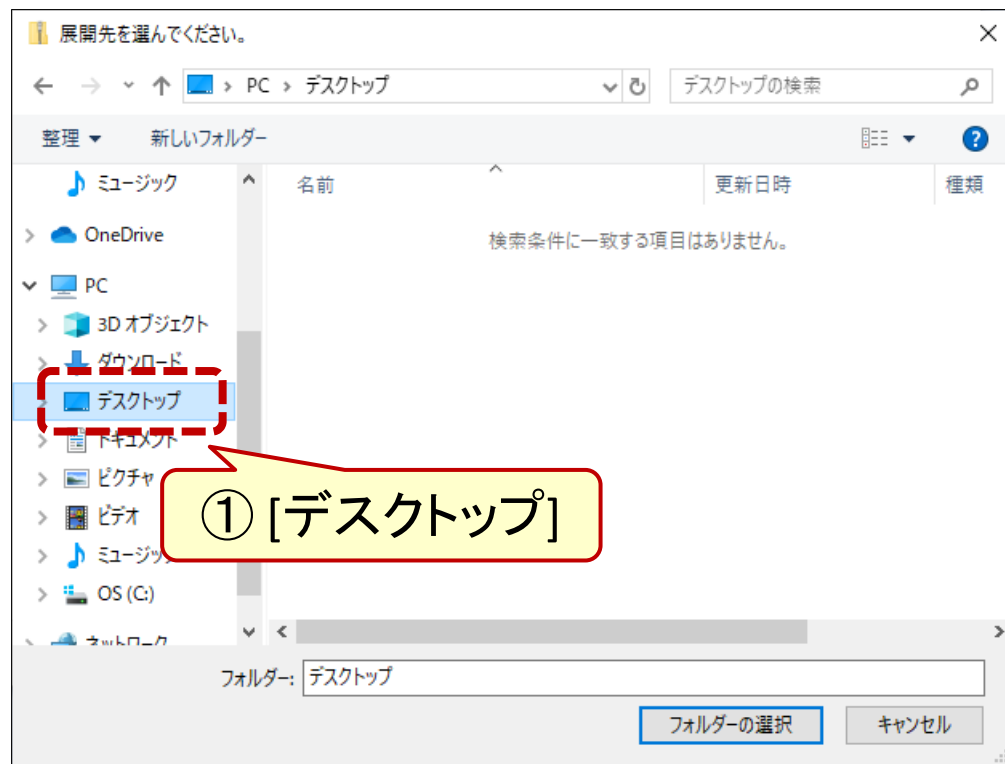
- 圧縮ファイルを右クリック
- [すべて展開] を選択
- 展開先フォルダを選択するため [参照] をクリック



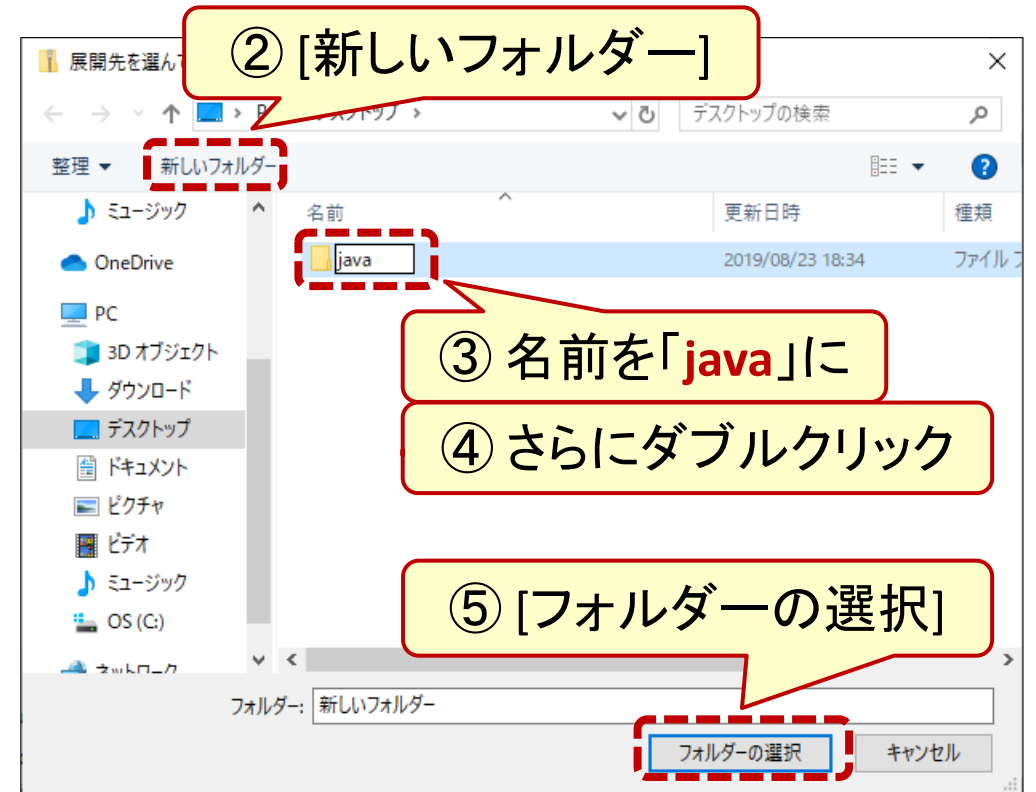
JDKのインストール

(4) 圧縮ファイルを展開(続き)

- 展開先を選択するダイアログでまずデスクトップへ移動



- [新しいフォルダー] をクリック
- 新フォルダの名前を「java」に



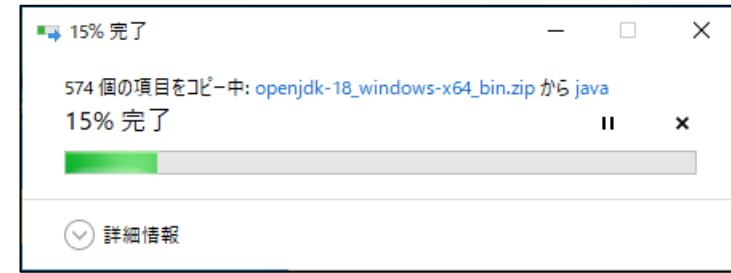
JDKのインストール

(5) 圧縮ファイルを展開(続き)

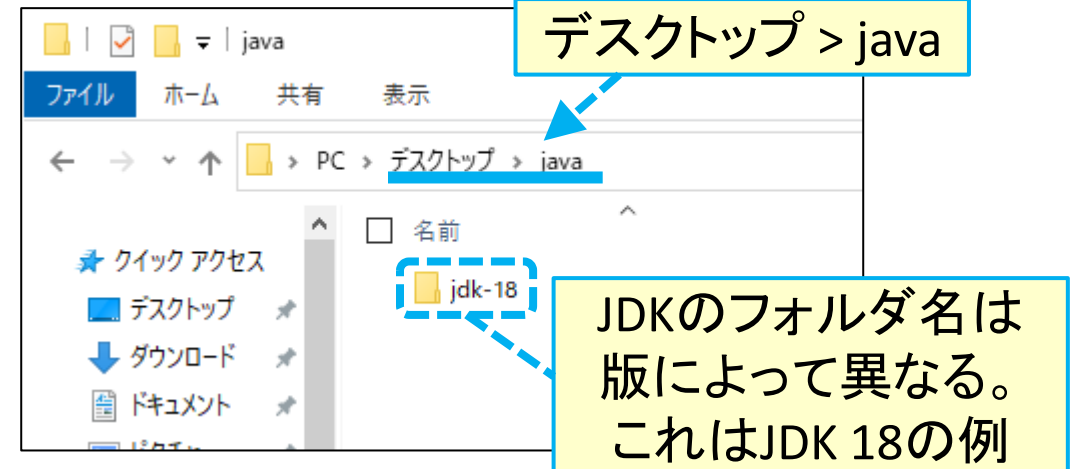
- 入力内容を確認したうえで展開を開始



- 展開処理の完了を待つ



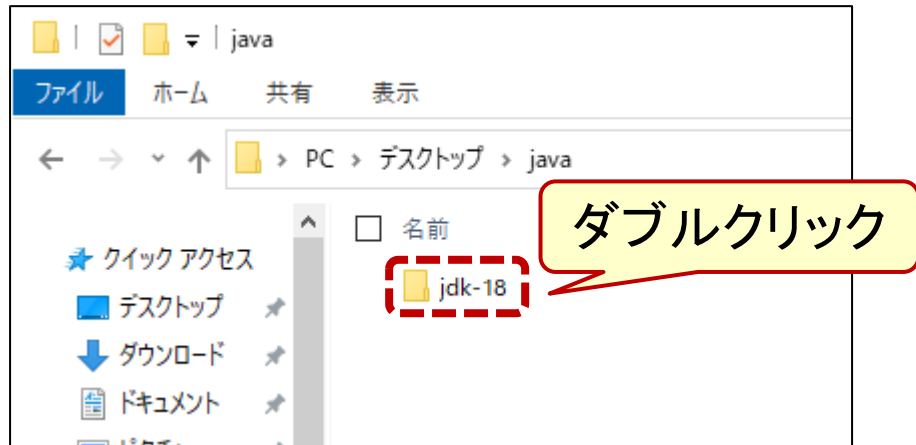
- フォルダが開く



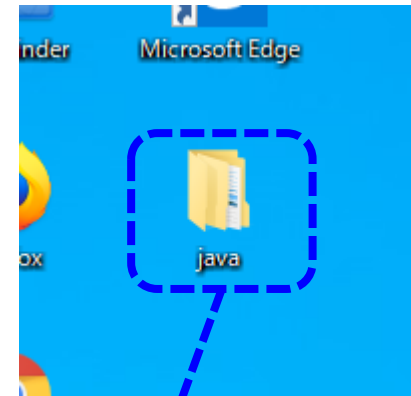
JDKのインストール

(6) 展開結果を確認

- JDKのフォルダを開く

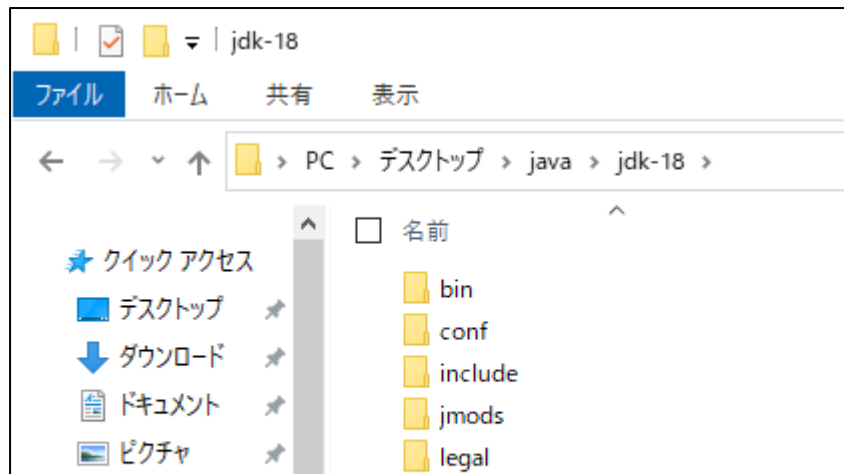


- デスクトップにフォルダ java のアイコンが現れたことも確認



フォルダ java を開くときには
このアイコンをダブルクリック

- JDKのフォルダの中身を確認

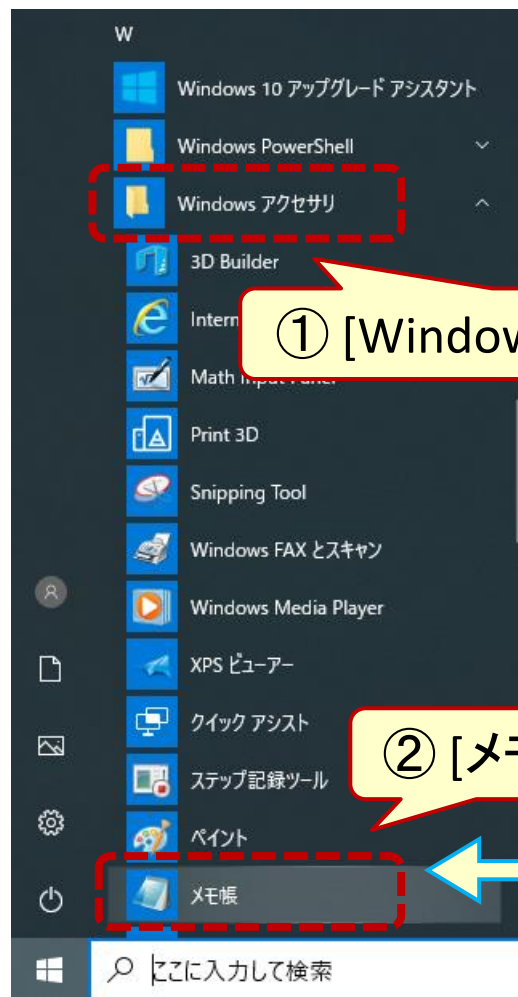


JDKの使用環境の設定

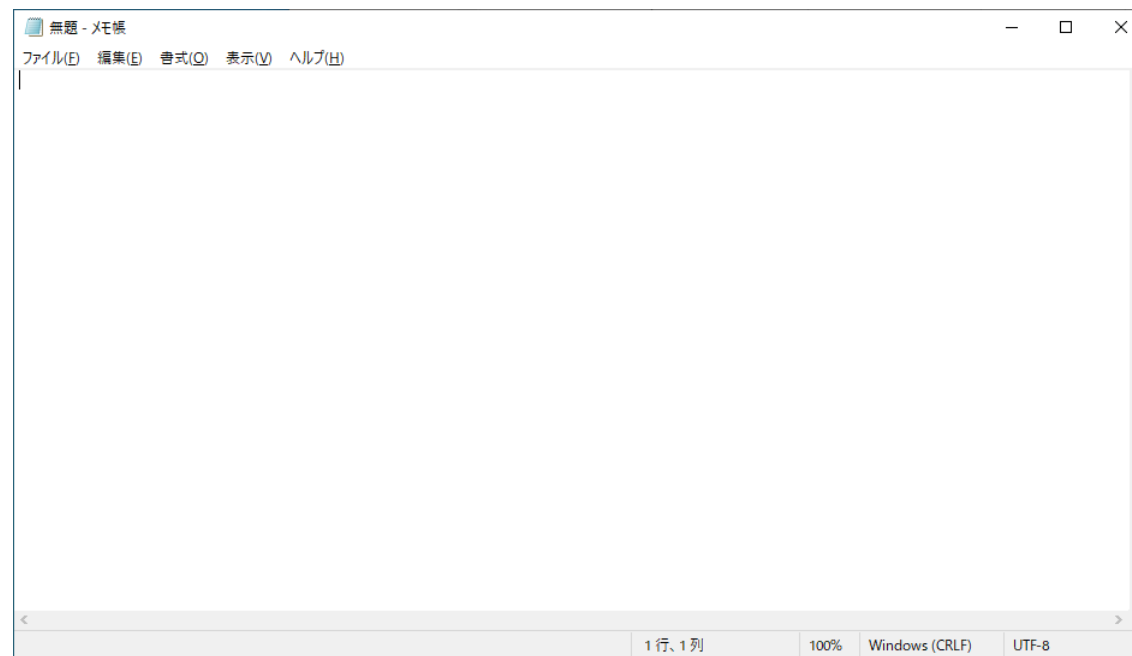
JDKの使用環境の設定

(1) テキストエディタを開く

- スタートメニューから



- 「メモ帳」が開く

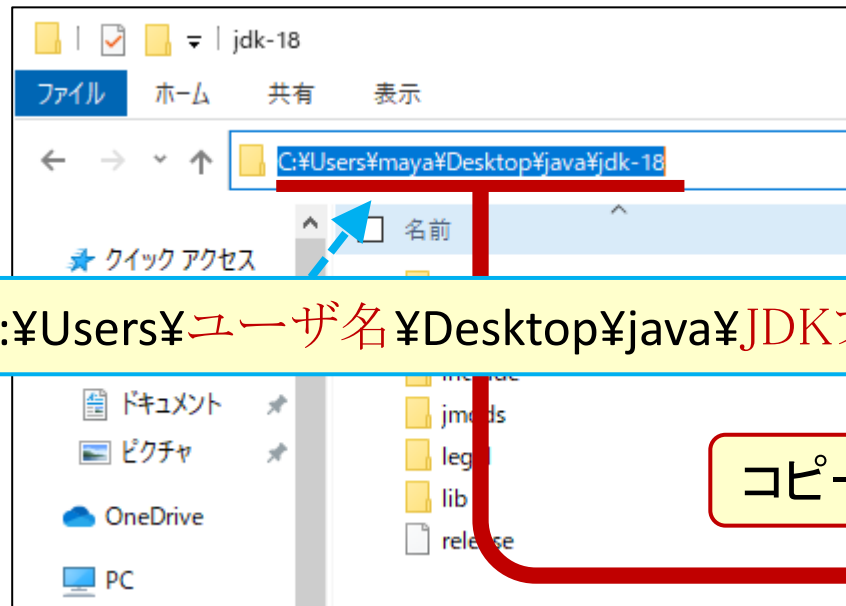


(参考) [メモ帳]を右クリックすると
スタート画面へのピン止めもできる

JDKの使用環境の設定

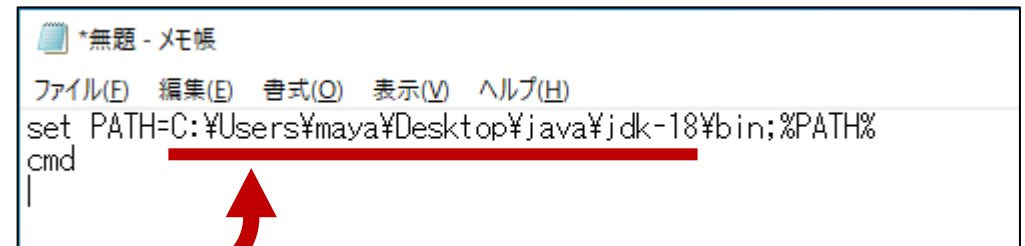
(2) バッチ処理を記述

- JDKのフォルダを開いておく
- アドレス部分をクリックし全選択
- コピー（[Ctrl]+[C] キー）
- 「メモ帳」に貼り付け（[Ctrl]+[V] キー）
- 次のように2行記述



```
set PATH=貼り付け部分¥bin;%PATH%
cmd
```

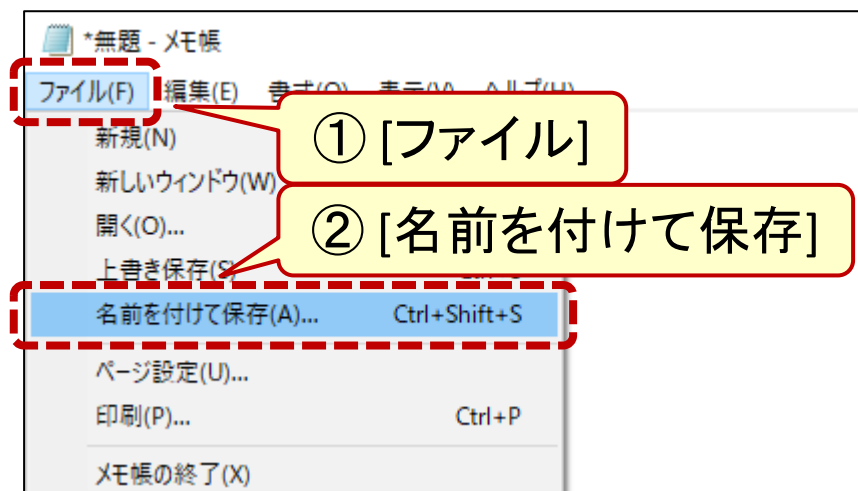
コロン「:」ではなく
セミコロン「;」



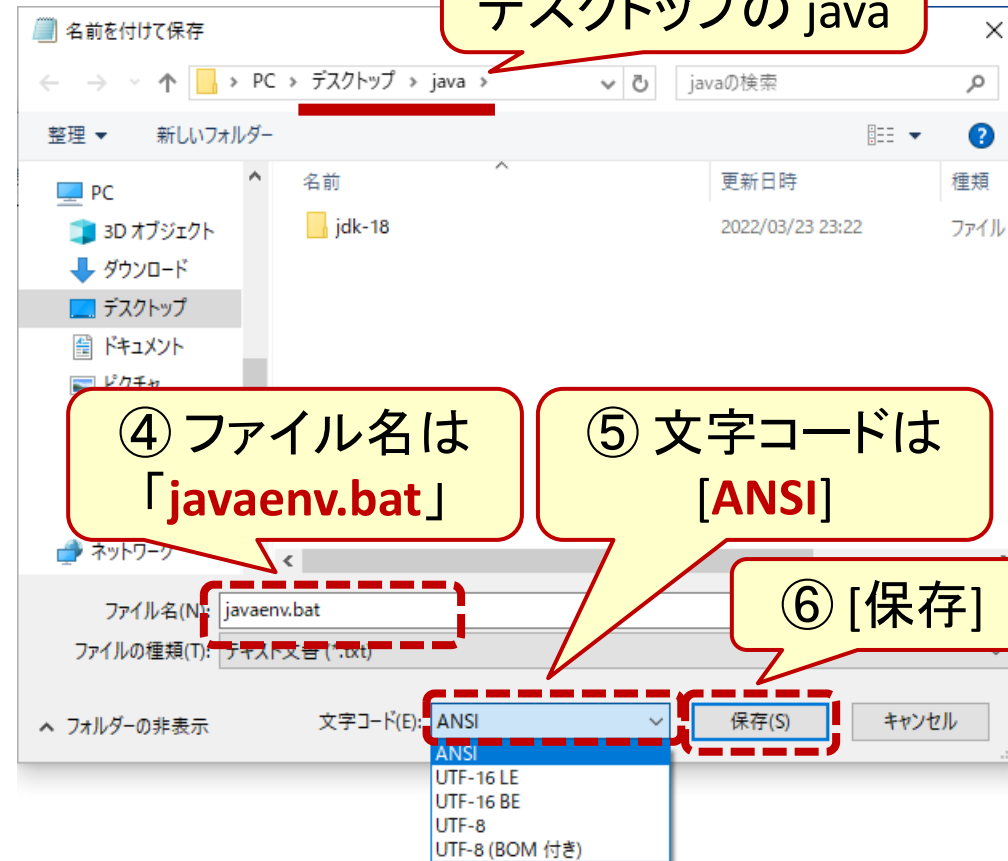
JDKの使用環境を設定

(3) バッチファイルを保存

- メニューから
[ファイル]-[名前を付けて保存]



- フォルダ java に「**javaenv.bat**」
として保存

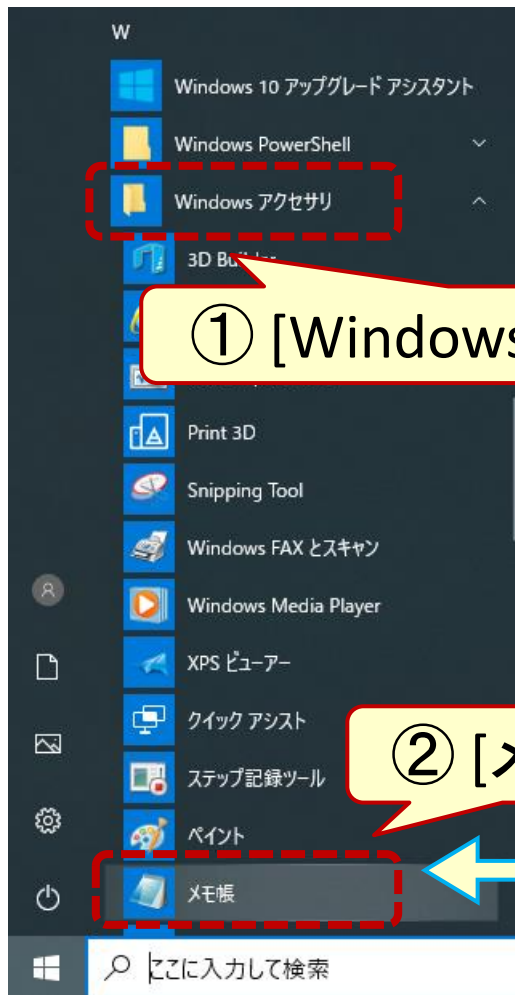


プログラムの入力

プログラムの入力

(1) テキストエディタを開く

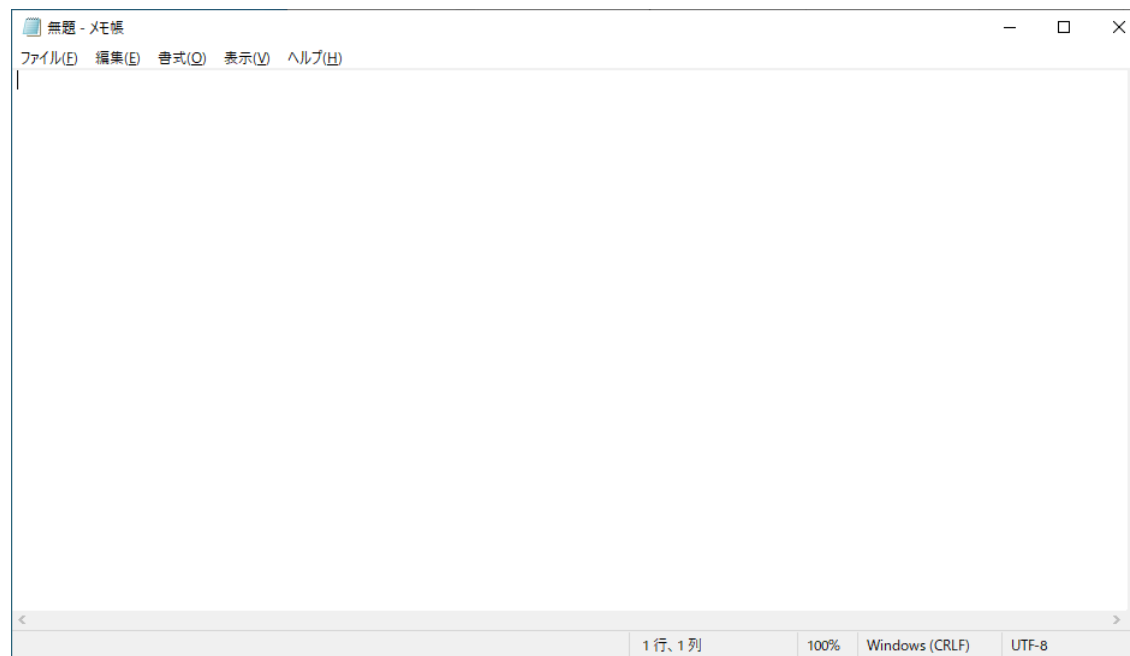
- スタートメニューから



① [Windowsアクセサリ]

② [メモ帳]

- 「メモ帳」を開く



(参考) [メモ帳]を右クリックすると
スタート画面へのピン止めもできる

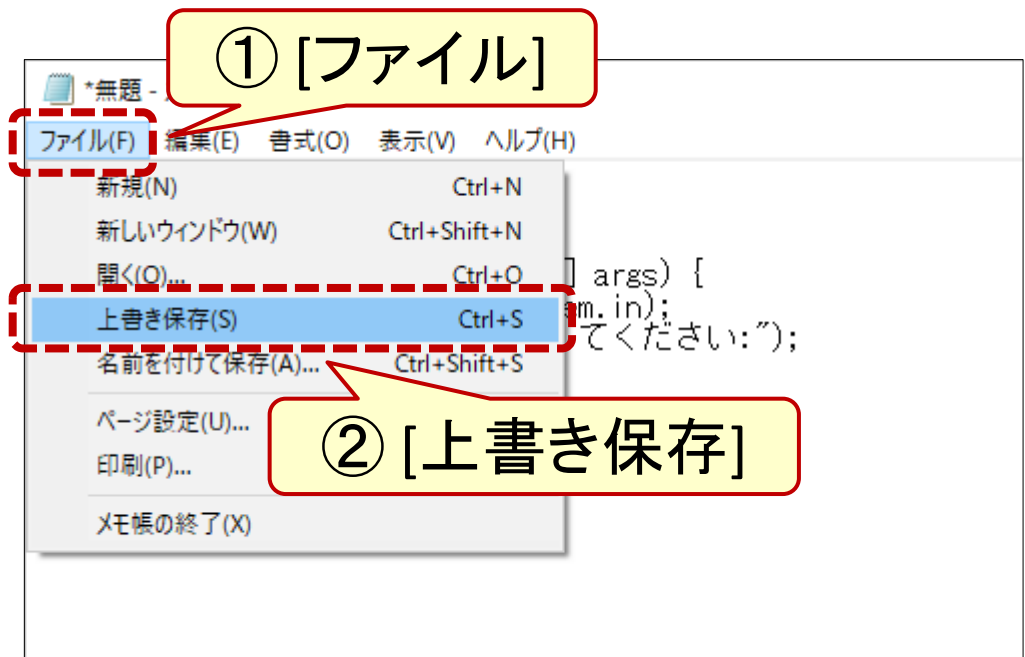
プログラムの入力

(2) プログラムの入力と保存

- プログラムを入力
 - この例は3.4節のリスト3.16
- ファイルに上書き保存
 - [Ctrl]+[S] でもOK

```
*無題 - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
import java.util.*;

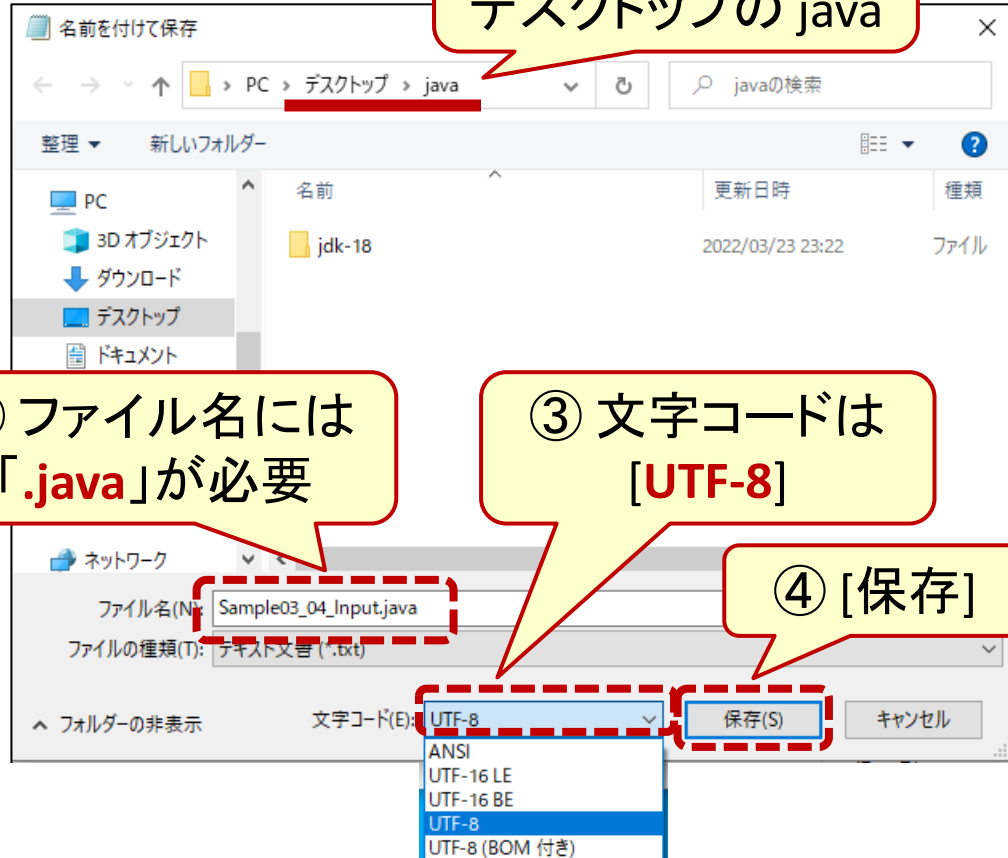
public class Sample03_04_Input {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("数値を入力してください:");
        int a;
        a = sc.nextInt();
        System.out.println("入力された数値:" + a);
    }
}
```



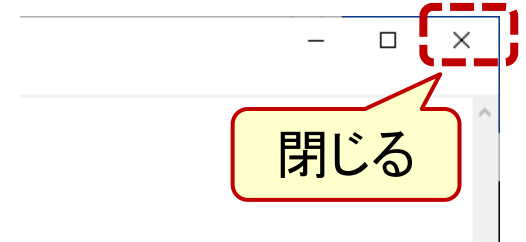
プログラムの入力

(3) 入力の終了と編集再開

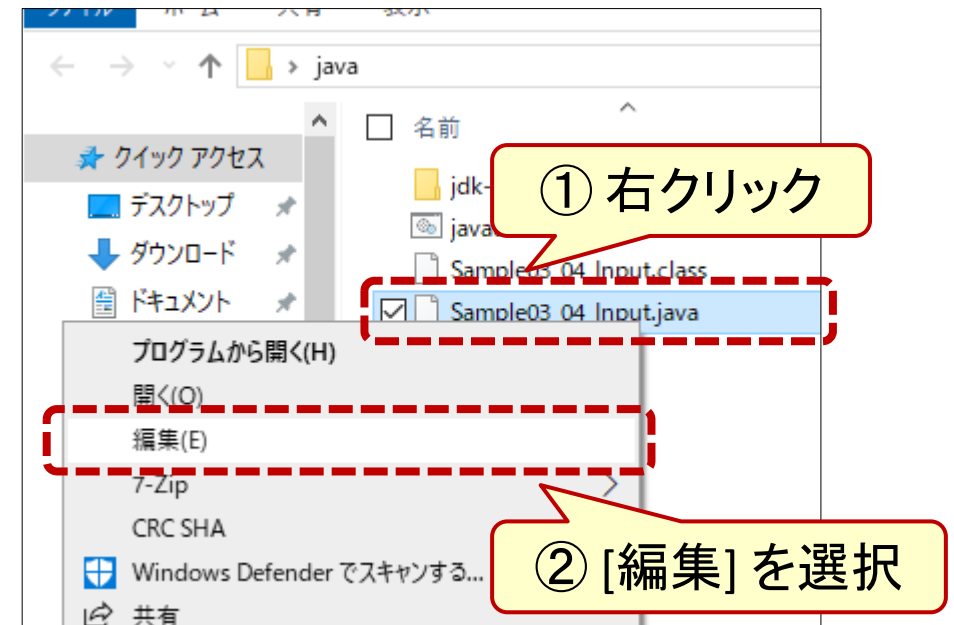
- 新規保存時のみ保存先を指定



- 「メモ帳」を閉じる



- ファイルを再び開きたいときは



プログラムの入力

(参考) テキストエディタ使用上の注意

- **JDK 18**以降を使用する場合
 - Javaプログラムの文字コードは**UTF-8**が既定(デフォルト)
 - Byte Order Mark (BOM)は無し
 - ソフトにより表記が違うので注意
 - (例)
 - 「UTF-8」「UTF-8(BOM付き)」があるときは前者
 - 「UTF-8」「UTF-8N」があるときは後者
- **JDK 17**までを使用する場合
 - Javaプログラムの文字コードは**シフトJIS**が既定(デフォルト)
- バッチファイルは**シフトJIS**に
 - ただし、半角英数のみ使うなら文字コードはどれでも同じ
- Windows付属のメモ帳では**ANSI**がシフトJISに相当

JDKの使用／トラブル対応

JDKの使用

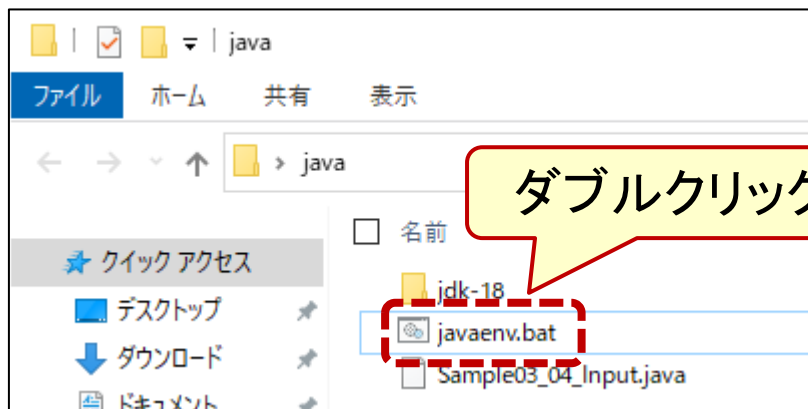
コンパイル(javac)と実行(java)

- デスクトップの java を開く



ダブルクリック

- javaenv.bat をダブルクリック



ダブルクリック

- コンパイル(javac)と実行(java)

javac ファイル名.java

```
C:¥Users¥taro¥Desktop¥java>javac Sample03_04_Input.java  
C:¥Users¥taro¥Desktop¥java>java Sample03_04_Input  
数値を入力してください:5  
入力された数値:5
```

java クラス名

※後ろに「java」は付けない

- 実行結果が表示される

(注意) プログラムを修正したら必ず

上書き保存 → javac → java

トラブル対応

(1) 文字コードのエラー、文字化け

[Q] 文字化けが発生したら？

文字化けの例

```
C:\Users\maya\Desktop\java>javac Sample03_04_Input.java
Sample03_04_Input.java:6: エラー: この文字(0x85)は、エンコーディングwindows-31jにマップ
System.out.print("誤-蛟、纒貞?・蛭呖@纒7纒上口纒輔>");
エラー1個
```

```
C:\Users\taro\Desktop\java>javac Sample03_04_Input.java
Sample03_04_Input.java:7: エラー: この文字(0x90)は、エンコーディングUTF-8にマップできま
System.out.print("???|????????????????");
Sample03_04_Input.java:7: エラー: この文字(0x94)は、エンコーディングUTF-8にマップできま
System.out.print("???|????????????????");
```

```
C:\work>javac Hello.java
```

```
C:\work>java Hello
纒連英 纒 纒連 纒連。纒連ッ
```

[A] 次のどちらかうまくいくほうで再コンパイルして、

```
javac -encoding utf8 ファイル名.java
```

```
javac -encoding ms932 ファイル名.java
```

java で実行して結果を確認

ms932 でうまくいった例

```
C:\Users\maya\Desktop\java>javac -encoding ms932 Sample03_04_Input.java
C:\Users\maya\Desktop\java>java Sample03_04_Input
数値を入力してください:5
入力された数値:5
```

詳細については「正誤表」の付録Aを参照

トラブル対応

(2) プログラム実行の強制終了

[Q] プログラムの実行が終了しなくなったら？

```
public class Sample05_02_For1 {  
    public static void main(String[] args) {  
        for (int i = 0; i < 5; i--) {  
            System.out.print("Hello, ");  
        }  
    }  
}
```

i++ を誤って i-- にした
プログラム例

実行すると延々と繰り返し

[A] [Ctrl]+[C] キーで強制終了

- [Ctrl] キーを押さえながら
- [C] キーをポンと押す

[Ctrl]+[C] で強制終了