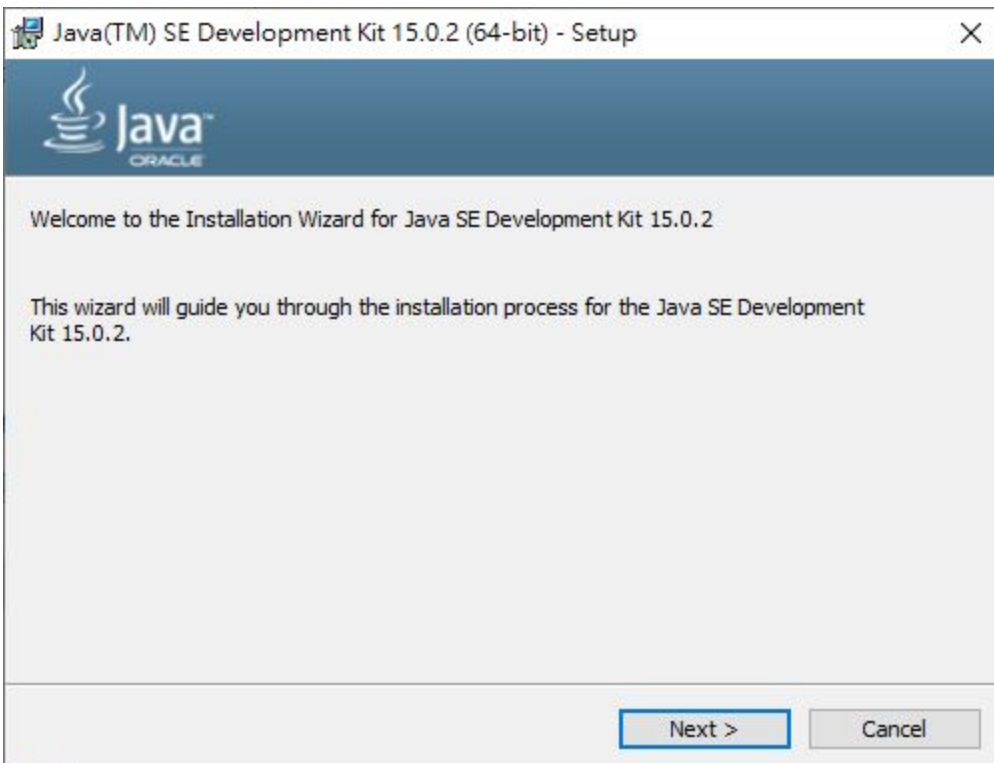
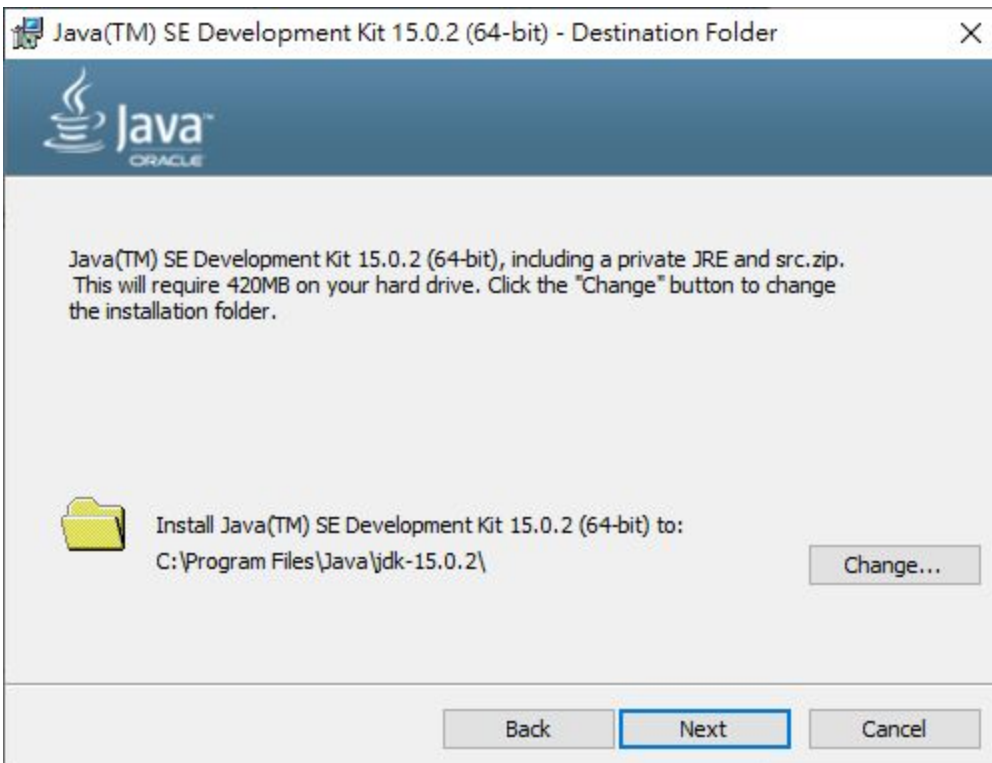


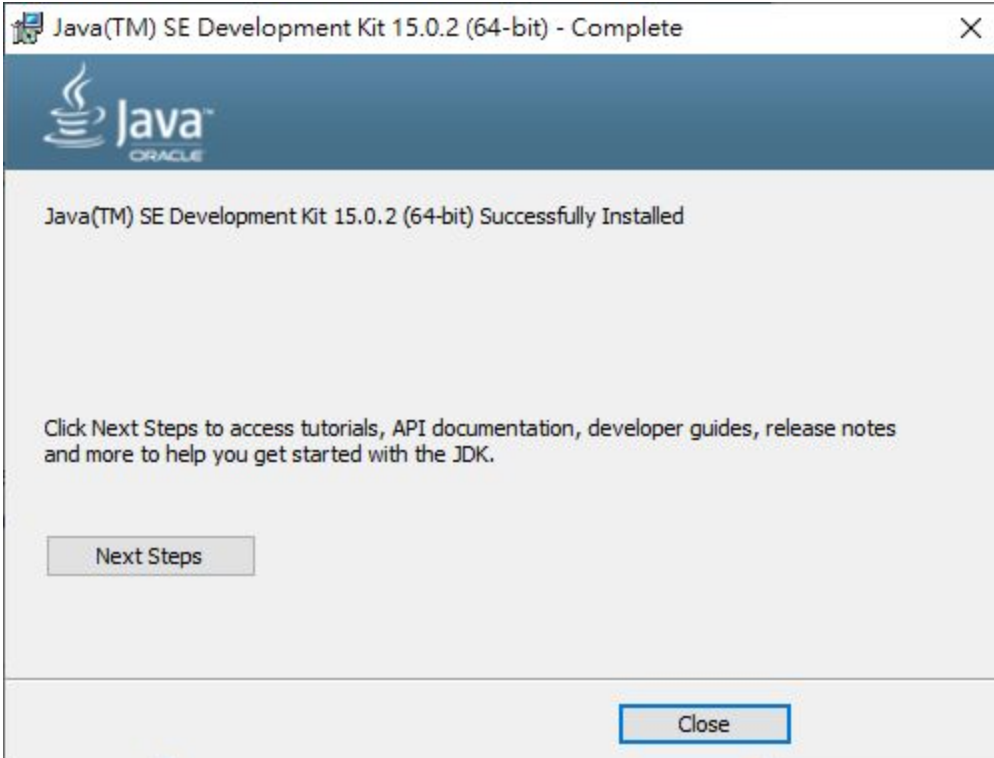
Java JDK

2021 02 25

<https://www.kjnotes.com/devtools/35>








JDK 15 Documentation - Home

docs.oracle.com/en/java/javase/15/index.html



Help Center

JDK 15 Documentation

Search

Home

API Documentation

Books

Related Resources


Home / Java / Java SE / 15

JDK 15 Documentation

Home


[Java Components page](#)

Looking for a different release? [Other releases](#)




Overview

- [Release Notes](#)
- [What's New](#)
- [Migration Guide](#)
- [Download the JDK](#)
- [Install Guide](#)
- [Version String](#)




Tools

- [JDK Tool Specifications](#)
- [JShell User's Guide](#)
- [Javadoc Guide](#)
- [Packaging Tool User Guide](#)



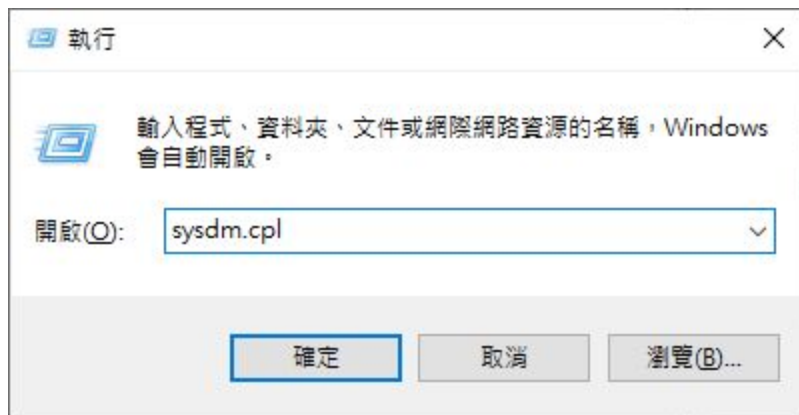
Language and Libraries

- [Language Updates](#)
- [Core Libraries](#)
- [JDK HTTP Client](#)
- [Java Tutorials](#)
- [Modular JDK](#)



Specifications

- [API Documentation](#)
- [Language and VM](#)
- [Java Security Standard Algorithm Names](#)
- [JAR](#)
- [Java Native Interface \(JNI\)](#)



PC 的使用者變數(U)

變數	值
OneDrive	C:\Users\PC\OneDrive
Path	C:\Users\PC\AppData\Local\Microsoft\W...
TEMP	C:\Users\PC\AppData\Local\Temp
TMP	C:\Users\PC\AppData\Local\Temp

新增(N)...

編輯(E)...

刪除(D)

系統變數(S)

變數	值
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PR...	4
OS	Windows_NT

新增(V)...

編輯(I)...

刪除(L)

確定

取消

環境變數

PC 的使用者變數(U)

新增系統變數

變數名稱(N): JAVA_HOME

變數值(V): C:\Program Files\Java\jdk-15.0.2

瀏覽目錄(D)...

瀏覽檔案(F)...

確定

取消

系統變數(S)

變數	值
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PR...	4
OS	Windows_NT

新增(W)...

編輯(I)...

刪除(L)

確定

取消

JAVA_HOME

C:\Program Files\Java\jdk-15.0.2

PC 的使用者變數(U)

變數	值
OneDrive	C:\Users\PC\OneDrive
Path	C:\Users\PC\AppData\Local\Microsoft\W...
TEMP	C:\Users\PC\AppData\Local\Temp
TMP	C:\Users\PC\AppData\Local\Temp

新增(N)...

編輯(E)...

刪除(D)

系統變數(S)

變數	值
NUMBER_OF_PR...	4
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\J...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;...

新增(W)...

編輯(O)...

刪除(L)

確定

取消

編輯環境變數



C:\Program Files\Common Files\Oracle\Java\javapath

%SystemRoot%\system32

%SystemRoot%

%SystemRoot%\System32\Wbem

%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\

%SYSTEMROOT%\System32\OpenSSH\

%JAVA_HOME%\bin

新增(N)

編輯(E)

瀏覽(B)...

刪除(D)

上移(U)

下移(O)

編輯文字(T)...

確定

取消

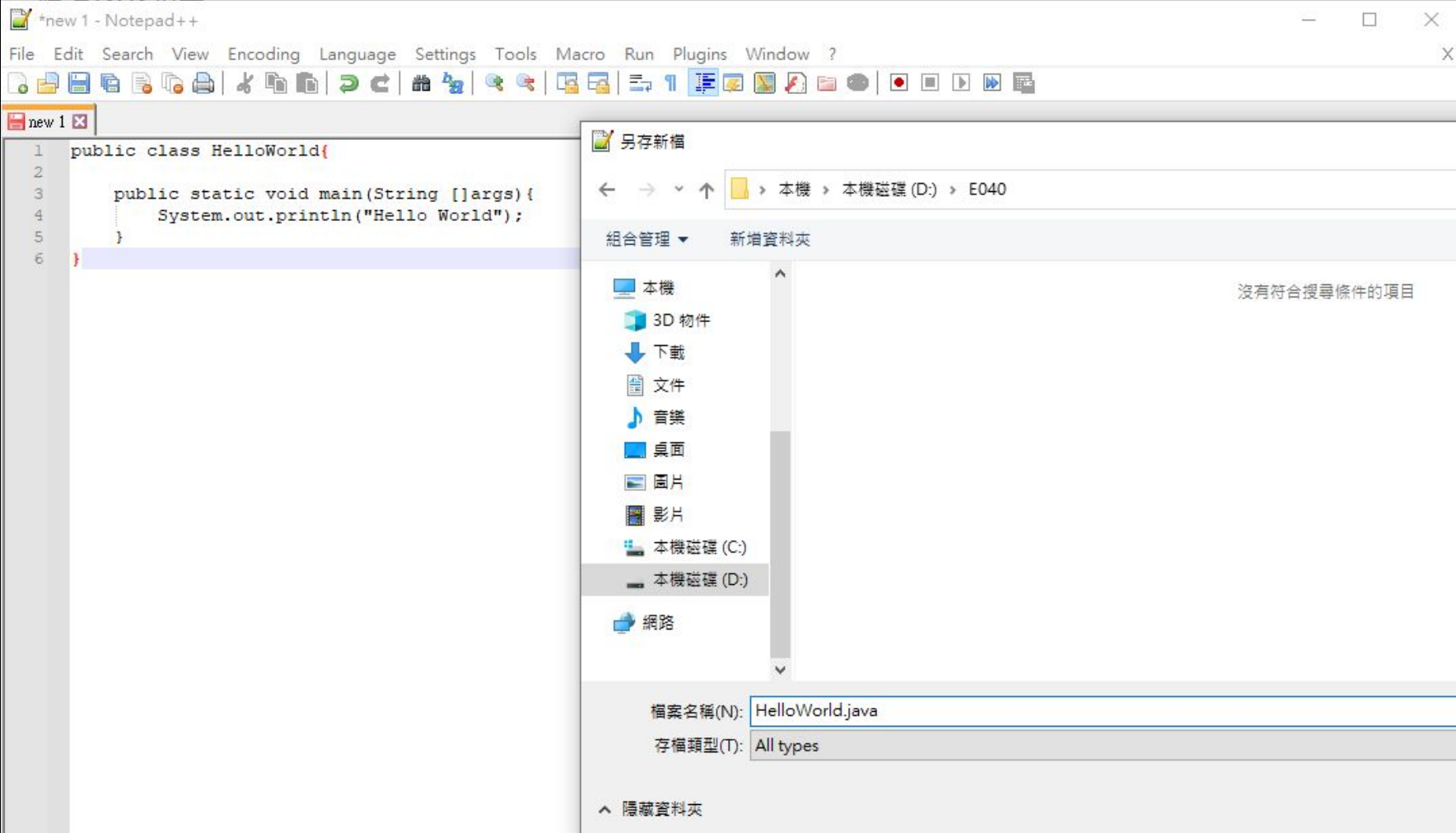
%JAVA_HOME%\bin

C:\Users\VC>javac

Usage: javac <options> <source files>

where possible options include:

```
@<filename>           Read options and filenames from file
-Akey[=value]          Options to pass to annotation processors
--add-modules <module>(<module>)*
    Root modules to resolve in addition to the initial modules, or all modules
    on the module path if <module> is ALL-MODULE-PATH.
--boot-class-path <path>, -bootclasspath <path>
    Override location of bootstrap class files
--class-path <path>, -classpath <path>, -cp <path>
    Specify where to find user class files and annotation processors
-d <directory>         Specify where to place generated class files
-deprecation
    Output source locations where deprecated APIs are used
--enable-preview
    Enable preview language features. To be used in conjunction with either -source or --release.
-encoding <encoding>   Specify character encoding used by source files
-endorseddirs <dirs>   Override location of endorsed standards path
-extdirs <dirs>        Override location of installed extensions
-g                    Generate all debugging info
-g:{lines,vars,source} Generate only some debugging info
-g:none              Generate no debugging info
-h <directory>
    Specify where to place generated native header files
--help, -help, -?      Print this help message
--help-extra, -X       Print help on extra options
-implicit:{none,class}
    Specify whether or not to generate class files for implicitly referenced files
-J<flag>               Pass <flag> directly to the runtime system
--limit-modules <module>(<module>)*
    Limit the universe of observable modules
--module <modules>(<module>)* -m <module>(<module>)*
    Compile only the specified module(s), check timestamps
--module-path <path>, -p <path>
    Specify where to find application modules
--module-source-path <module-source-path>
    Specify where to find input source files for multiple modules
--module-version <version>
    Specify version of modules that are being compiled
-novarn               Generate no warnings
-parameters
    Generate metadata for reflection on method parameters
-proc:{none,only}
    Control whether annotation processing and/or compilation is done.
-processor <class1>[,<class2>,<class3>...]
    Names of the annotation processors to run; bypasses default discovery process
--processor-module-path <path>
    Specify a module path where to find annotation processors
--processor-path <path>, -processorpath <path>
    Specify where to find annotation processors
-profile <profile>
    Check that API used is available in the specified profile
--release <release>
    Compile for the specified Java SE release. Supported releases: 7, 8, 9, 10, 11, 12, 13, 14, 15
-s <directory>         Specify where to place generated source files
--source <release>, -source <release>
    Provide source compatibility with the specified Java SE release. Supported releases: 7, 8, 9, 10, 11, 12, 13, 14, 15
--source-path <path>, -sourcepath <path>
    Specify where to find input source files
--system <jdk>{none}   Override location of system modules
```





管理

E040

檔案

常用

共用

檢視

圖片工具



本機 > 本機磁碟 (D:) > E040



搜尋 E040

快速存取

OneDrive

本機

3D 物件

下載

文件

音樂

桌面

圖片

影片

本機磁碟 (C:)

本機磁碟 (D:)

網路



HelloWorld.class



HelloWorld.java

```
cmd 命令提示字元
--target <release>, -target <release>
    Generate class files suitable for the specified Java SE release. Supported releases: 7, 8, 9, 10, 11, 12, 13, 14, 15
--upgrade-module-path <path>
    Override location of upgradeable modules
-verbose
    Output messages about what the compiler is doing
--version, -version
    Version information
-Werror
    Terminate compilation if warnings occur

C:\Users\PC>D:
D:\>cd E040
D:\E040>javac HelloWorld.java
D:\E040>java HelloWorld
Hello World
D:\E040>
```

Execute	Share	Source File	STDIN	Result
<pre>1 public class HelloWorld{ 2 3 public static void main(String []args){ 4 System.out.println("Hello World"); 5 } 6 }</pre>				<pre>\$javac HelloWorld.java \$java -Xmx128M -Xms16M HelloWorld Hello World</pre>

println()和print()有何差別？

Execute	Share	Source File	STDIN	Result
<pre>1- public class HelloWorld2{ 2 3- public static void main(String []args){ 4 System.out.print("Hello"); 5 System.out.print("Mydeargreatteacher"); 6 System.out.println("Hello World"); 7 } 8 }</pre>				<pre>\$javac HelloWorld2.java \$java -Xmx128M -Xms16M HelloWorld2 HelloMydeargreatteacherHello World</pre>

```
D:\E040>javac HelloWorld2.java
D:\E040>java HelloWorld2
HelloMydeargreatteacherHello World
```

Execute	Share	Source File	STDIN	Result
<pre>1- public class HelloWorld21{ 2 3- public static void main(String []args){ 4 System.out.print("Hello"); 5 System.out.println("Mydeargreatteacher"); 6 System.out.println("Hello World"); 7 } 8 }</pre>				<pre>\$javac HelloWorld21.java \$java -Xmx128M -Xms16M HelloWorld21 HelloMydeargreatteacher Hello World</pre>

```
D:\E040>java HelloWorld21
HelloMydeargreatteacher
Hello World
```

D:\E040\HelloWorld2.java - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Hello World.java x HelloWorld2.java x HelloWorld21.java x

```
1 public class HelloWorld2{
2
3     public static void main(String []args){
4         System.out.print("Hello");
5         System.out.print("Mydeargreatteacher");
6         System.out.println("Hello World");
7     }
8 }
```

另存新檔

← → ↕ ↑ > 本機 > 本機磁碟 (D:) > E040

組合管理 ▾

新增資料夾

OneDrive

本機

3D 物件

下載

文件

音樂

桌面

圖片

影片

本機磁碟 (C:)

本機磁碟 (D:)

網路

HelloWorld.clas
s

HelloWorld.java

HelloWorld2.cl
ss

HelloWorld2.jav
a

檔案名稱(N): HelloWorld2

檔案類型(T): All types

```
D:\E040>javac HelloWorld2.java
```

```
D:\E040>java HelloWorld2.java
```

```
Hello MydeargreatteacherHello World
```

D:\E040\HelloWorld21.java - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?



HelloWorld.java x HelloWorld2.java x HelloWorld21.java x

```
1 public class HelloWorld21{
2
3     public static void main(String []args){
4         System.out.print("Hello");
5         System.out.println("Mydeargreatteacher");
6         System.out.println("Hello World");
7     }
8 }
```

另存新檔

← → ↶ ↷ 本機 > 本機磁碟 (D:) > E040

組合管理 ▾

新增資料夾

OneDrive

本機

3D 物件

下載

文件

音樂

桌面

圖片

影片

本機磁碟 (C:)

本機磁碟 (D:)

網路



HelloWorld.clas
s



HelloWorld.java



HelloWorld2.cl
ss



HelloWorld2.jav
a

檔案名稱(N): HelloWorld21

存檔類型(T): All types

▲ 隱藏資料夾

```
D:\E040>javac HelloWorld21.java
```

```
D:\E040>java HelloWorld21.java
```

```
Hello Mydeargreatteacher
```

```
Hello World
```

D:\E040\Test.java - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugin

Test.java

```
1 public class Test{
2
3     public static void main(String []args){
4         for(int x=1;x<args.length;x++){
5             System.out.print(args[x]+" ");
6         }
7     }
8 }
```

D:\E040>javac Test.java
D:\E040>java Test

Execute | Share | Source File | STDIN | Result

```
1 public class Test{
2
3     public static void main(String []args){
4         for(int x=1;x<args.length;x++){
5             System.out.print(args[x]+" ");
6         }
7     }
8 }
```

\$javac Test.java
\$java -Xmx128M -Xms16M Test



Fibonacci.java

```
1 public class Fibonacci{
2
3     public static void main(String[] args){
4
5         /*
6          * Write a program to find first 20 numbers of Fibonacci Series
7          * 0 1 1 2 3 5 8 13 21
8          */
9         int num1 = 0;
10        int num2 = 1;
11        int num3 = 0;
12
13        System.out.println(num1);
14        System.out.println(num2);
15
16        for(int i = 1; i <= 18; i++){
17            num3 = num1 + num2;
18            System.out.println(num3);
19            num1 = num2;
20            num2 = num3;
21        }
22    }
23 }
```

```
D:\E040>javac Fibonacci.java
D:\E040>java Fibonacci
0
1
1
2
3
5
8
13
21
34
55
89
144
233
377
610
987
1597
2584
4181
```

Execute	Share	Source File	STDIN	Result
<pre>1 public class Fibonacci{ 2 3 public static void main(String[] args){ 4 5 /* 6 * Write a program to find first 20 numbers of Fibonacci Series 7 * 0 1 1 2 3 5 8 13 21 8 */ 9 int num1 = 0; 10 int num2 = 1; 11 int num3 = 0; 12 13 System.out.println(num1); 14 System.out.println(num2); 15 16 for(int i = 1; i <= 18; i++){ 17 num3 = num1 + num2; 18 System.out.println(num3); 19 num1 = num2; 20 num2 = num3; 21 } 22 } 23 }</pre>				<pre>\$javac Fibonacci.java \$java -Xmx128M -Xms16M Fibonacci 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181</pre>

反編譯

```
C:\Users\PC>javap
Usage: javap <options> <classes>
where possible options include:
--help -help -h -?      Print this help message
-version                Version information
-v -verbose             Print additional information
-l                     Print line number and local variable tables
-public                Show only public classes and members
-protected             Show protected/public classes and members
-package               Show package/protected/public classes
                       and members (default)
-p -private            Show all classes and members
-c                     Disassemble the code
-s                     Print internal type signatures
-sysinfo                Show system info (path, size, date, SHA-256 hash)
                       of class being processed
-constants              Show final constants
--module <module>, -m <module> Specify module containing classes to be disassembled
-J<vm-option>           Specify a VM option
--module-path <path>    Specify where to find application modules
--system <jdk>           Specify where to find system modules
--class-path <path>     Specify where to find user class files
-classpath <path>       Specify where to find user class files
-cp <path>              Specify where to find user class files
-bootclasspath <path>   Override location of bootstrap class files
--multi-release <version> Specify the version to use in multi-release JAR files
```

GNU-style options may use = instead of whitespace to separate the name of an option from its value.

Each class to be shown may be specified by a filename, a URL, or by its fully qualified class name. Examples:

```
path/to/MyClass.class
jar:file:///path/to/MyJar.jar!/mypkg/MyClass.class
java.lang.Object
```

javap -v
不僅會輸出行號、
本地變亮表信息、
反編譯匯編代碼，
還會輸出當前用到的常量池等信息

```
D:\E040>javap -v HelloWorld.class
Classfile /D:/E040/HelloWorld.class
  Last modified 2021年2月26日; size 425 bytes
  SHA-256 checksum 6a6c660f2bdb2bb3da7df6b4bed2c0be5ea00e6bcdd47108f893f52221cdedc1c
  Compiled from "HelloWorld.java"
public class HelloWorld
  minor version: 0
  major version: 59
  flags: (0x0021) ACC_PUBLIC, ACC_SUPER
  this_class: #21
  super_class: #2
  interfaces: 0, fields: 0, methods: 2, attributes: 1
Constant pool:
 #1 = Methodref      #2.#3      // java/lang/Object."<init>":()V
 #2 = Class           #4         // java/lang/Object
 #3 = NameAndType     #5:#6      // "<init>":()V
 #4 = Utf8            java/lang/Object
 #5 = Utf8            <init>
 #6 = Utf8            ()V
 #7 = Fieldref        #8.#9      // java/lang/System.out:Ljava/io/PrintStream;
 #8 = Class           #10        // java/lang/System
 #9 = NameAndType     #11:#12     // out:Ljava/io/PrintStream;
 #10 = Utf8           java/lang/System
 #11 = Utf8           out
 #12 = Utf8           Ljava/io/PrintStream;
 #13 = String         #14        // Hello World
 #14 = Utf8           Hello World
 #15 = Methodref      #16.#17     // java/io/PrintStream.println:(Ljava/lang/String;)V
 #16 = Class           #18        // java/io/PrintStream
 #17 = NameAndType     #19:#20     // println:(Ljava/lang/String;)V
 #18 = Utf8           java/io/PrintStream
 #19 = Utf8           println
 #20 = Utf8           (Ljava/lang/String;)V
 #21 = Class           #22        // HelloWorld
 #22 = Utf8           HelloWorld
 #23 = Utf8           Code
 #24 = Utf8          LineNumberTable
 #25 = Utf8           main
 #26 = Utf8           ([Ljava/lang/String;)V
 #27 = Utf8           SourceFile
 #28 = Utf8           HelloWorld.java
{
  public HelloWorld();
    descriptor: ()V
    flags: (0x0001) ACC_PUBLIC
    Code:
      stack=1, locals=1, args_size=1
      0: aload_0
      1: invokespecial #1      // Method java/lang/Object."<init>":()V
      4: return
    LineNumberTable:
      line 1: 0
  public static void main(java.lang.String[]);
    descriptor: ([Ljava/lang/String;)V
    flags: (0x0009) ACC_PUBLIC, ACC_STATIC
    Code:
      stack=2, locals=1, args_size=1
      0: getstatic      #7      // Field java/lang/System.out:Ljava/io/PrintStream;
      3: ldc            #13      // String Hello World
      5: invokevirtual #15      // Method java/io/PrintStream.println:(Ljava/lang/String;)V
      8: return
    LineNumberTable:
      line 4: 0
      line 5: 8
}
SourceFile: "HelloWorld.java"
```

javap -l

會輸出行號和本地變量表信息

```
D:\E040>javap -l HelloWorld.class
Compiled from "HelloWorld.java"
public class HelloWorld {
    public HelloWorld();
        lineNumberTable:
            line 1: 0

    public static void main(java.lang.String[]);
        lineNumberTable:
            line 4: 0
            line 5: 8
}
```

javap -c

會對當前class字結碼進行反編譯生成匯編代碼

```
D:\E040>javap -c HelloWorld.class
Compiled from "HelloWorld.java"
public class HelloWorld {
    public HelloWorld();
        Code:
            0: aload_0
            1: invokespecial #1                  // Method java/lang/Object."<init>":()V
            4: return

    public static void main(java.lang.String[]);
        Code:
            0: getstatic     #7                  // Field java/lang/System.out:Ljava/io/PrintStream;
            3: ldc           #13                 // String Hello World
            5: invokevirtual #15                 // Method java/io/PrintStream.println:(Ljava/lang/String;)V
            8: return
}
```

javap的選項

--help, -help, -h, 或者 ~?
打印javap命令的幫助消息。

-version
打印發行信息。

-verbose 或者 -v
打印有關所選類的其他信息。

-l
打印行和局部變量表。

-public
僅顯示公共班級和成員。

-protected
僅顯示受保護的和公開的班級和成員。

-package
顯示軟件包/受保護的/公共的類和成員（默認）。

-private 或者 -p
顯示所有班級和成員。

-c
為該類中的每個方法打印反彙編的代碼，例如構成Java字節碼的指令。

-s
打印內部類型簽名。

-sysinfo
顯示正在處理的類的系統信息（路徑，大小，日期，SHA-256雜希）。

-constants
顯示static final常數。

--module *模塊*或-m *模塊*
指定包含要反彙編的類的模塊。

--module-path *小路*
指定在哪裡找到應用程序模塊。

--system *模塊*
指定在何處查找系統模塊。

--class-path *路徑*, -classpath *路徑*或-cp *路徑*
指定javap命令用於查找用戶類文件的路徑。設置後，它將覆蓋默認值或CLASSPATH環境變量。

-bootclasspath *小路*
覆蓋引導類文件的位置。

--multi-release *版本*
指定要在多版本JAR文件中選擇的版本。

-J*選項*
將指定的選項傳遞給JVM，例如：

```
javap -J-version
```

```
javap -J-Djava.security.manager -J-Djava.security.policy=MyPolicy MyClassName
```

見*Java 選項的概述*中的Java。

<https://docs.oracle.com/en/java/javase/15/docs/specs/man/javap.html#javap-example>