

第01课

史蒂夫的第一句话

从一行输出开始，搭起自己的第一座代码小屋



程序不会猜你的意思。你写“欢迎”，它就说欢迎；少写一个符号，它可能连第一句话都说不出来。这一课，我们从一份完整程序出发，学会运行、控制换行、读懂报错，最后做出一个有自己风格的服务器欢迎屏。

阅读方式：先猜结果，再运行核对；每次只改一个条件。

使用 Java 11 或更新版本。故事与插画借用方块世界，本系列练习运行在普通 Java 控制台。

01 电脑怎样才会开口？



插画中的工作台提供故事背景；真正的输出会出现在开发工具的控制台。

史蒂夫站在工作台前，想告诉伙伴：“先带火把，再进矿洞。”可是，电脑并不知道哪句话重要。它需要一条写得清清楚楚的指令。这正是编程最有趣的地方：把一个模糊想法，变成别人也能执行的步骤。

带着三个问题往下读

- 程序从哪里开始执行？
- 文字为什么要放在引号里？
- 说完一句话后，怎样决定要不要换行？

先准备一个安装了 JDK 的 Java 开发工具，使用 Java 11 或更新版本。创建普通 Java 项目即可；这几课暂时不需要连接 Minecraft 或安装模组。

02 第一份完整程序：找到 main

新建 HelloWorld.java。文件名与 public class 后面的名字要一致，连大小写也要一致。如果工具已经生成 main，使用已有的入口，别再套一层模板。

完整程序 • HelloWorld.java

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("你好，方块世界！");
    }
}
```

运行核对

你好，方块世界！

读代码，先看结构

位置	作用
class HelloWorld	给这份代码中的类命名
main(...)	本课程的执行入口
System.out.println(...)	把文字送到控制台
两对 { }	外层是类，内层是 main 的范围

先不着急背 public、static 和 void。认出入口、找到两对花括号，并让第一句话成功出现，就是本页的目标。

03 从源文件到运行结果



你负责表达想法；编译器和 JVM 负责把规则落到机器上。

一段代码的四站旅程



保存文件后，开发工具通常会替你完成编译和运行。编译先检查代码是否符合 Java 规则；运行时，JVM 执行生成的字节码。程序仍可能在运行时遇到问题，因此“编译通过”只是第一关。

想看看工具替你做了什么？

选学：在文件所在目录执行的终端命令

```
javac -encoding UTF-8 HelloWorld.java
java HelloWorld
```

第一行编译源文件；第二行运行类，不加 .java 或 .class 后缀。若项目自动加入 package 声明，优先使用开发工具的运行按钮，本课终端示例按没有 package 的文件编写。

04 print 和 println：换行由谁负责？

把控制台想成一张不断向下写的便笺。print 写完内容后停在原处；println 写完内容，再把下一次输出送到新行。引号里的空格也会原样出现。

一行继续，还是下一行？

print

写内容
停在同一行

println

写内容
在结尾换行

下一次输出

从当前光标位置开始

片段：放在 main 中，单独运行

```
System.out.print("钻石");  
System.out.print(" + ");  
System.out.println("木棍");  
System.out.println("第一层：草方块");  
System.out.println("第二层：泥土");  
System.out.println();  
System.out.print("嘶.....");  
System.out.println("快跑！");
```

运行核对

钻石 + 木棍
第一层：草方块
第二层：泥土

嘶.....快跑！

不带内容的 println() 也会换行，所以这里出现一条空行。先画出每行的文字，再运行核对；“差一条空行”也是可以解释的实验结果。

05 小符号不是装饰



一条不起眼的连接缺失，就可能让整条红石线路停下来。

代码中的符号各有职责。分号结束本块的声明或输出语句；花括号划定范围；圆括号装着方法需要的内容；英文双引号标出文字；双斜杠开始一条注释。

三份有问题的片段

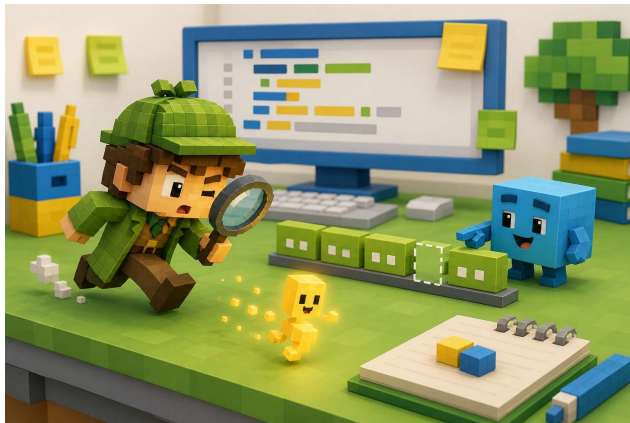
故意写错：分别检查，不要整段复制运行

```
System.out.println("苦力怕来了")
System.out.println( "快跑! " );
System.out.Println("欢迎! ");
```

- 第一行缺少分号，补上 ;。
- 第二行用了中文弯引号，改成英文双引号 "。
- 第三行的 Println 大小写不对，应为 println。

编译器指出的行号，有时是它“终于发现不对劲”的位置。检查那一行，也看看前一行是否少了引号、分号或括号。

06 注释：留给未来自己的说明



排错时，比到处改动，一条清楚的线索更有用。

好注释解释的是选择背后的原因。半年后的你，往往和第一次读这段代码的朋友一样，需要一点背景。注释不会成为控制台输出，也不会替程序执行任务。

片段：放在 main 中，单独运行

```
// 先显示安全提示，再介绍今天的任务。
System.out.println("带上火把，避免迷路。");

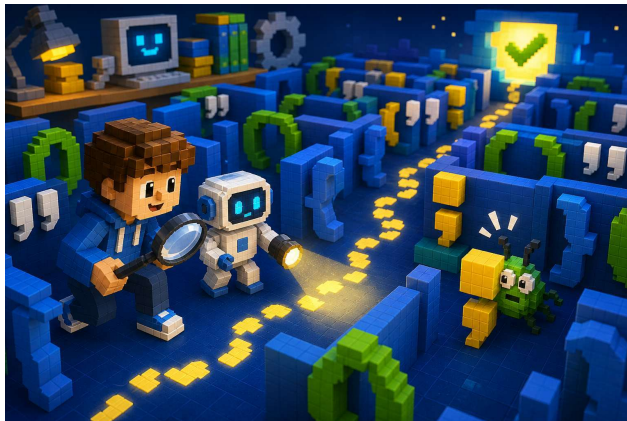
/* 欢迎屏只负责展示文字。
   输入和指令处理会在后面的课程学习。 */
System.out.println("今日任务：收集原木。");
```

运行核对

带上火把，避免迷路。
今日任务：收集原木。

不要给“输出欢迎”再写一条“这里输出欢迎”。能说明为什么这样安排、有什么限制的注释，才是在帮助读者。

07 第一次调试：让差异变成线索



调试不是和电脑吵架，而是寻找“预期”与“实际”分开的地方。

看到红线，先停一下。找到第一条与你的文件有关的报错，读清文件名和行号，再检查附近的符号。一次只改一个原因，保存后重新编译。这样，你才知道是哪一步起了作用。

找错实验：这份程序故意不能编译

```
public class BugHunt {  
    public static void main(String[] args) {  
        System.out.println("我会找到它! ")  
    }  
}
```

先回答，再动手

- 输出语句的末尾少了什么？
- 它属于编译错误，还是运行之后的错误？
- 修好后应该只输出哪一行？

答案：补上分号；这是编译错误；修复后输出“我会找到它！”。如果程序能运行却说错了话，就比较内容和顺序，而不是继续寻找分号。

08 小作品：服务器欢迎屏

一张好的欢迎屏会告诉读者：这里叫什么、谁来了、今天做什么、应该注意什么。先在纸上排好行，再让每一行对应一条输出语句。

把欢迎屏拆成四个任务



完整程序 • ServerWelcome.java

```

public class ServerWelcome {
    public static void main(String[] args) {
        System.out.println("=====");
        System.out.println("欢迎来到：方块探险营");
        System.out.print("玩家：");
        System.out.println("Alex");
        System.out.println("今日任务：收集 10 个原木");
        // 安全提示放在趣味提示前面。
        System.out.println("安全提示：不要直直地向下挖");
        System.out.println("趣味提示：苦力怕也想听课，请保持距离");
        System.out.println("=====");
    }
}
  
```

这是控制台中的欢迎屏。它不会把文字发送到游戏服务器；不过，排版、任务分解和排错的方法，已经可以带到下一份程序里。

09 验收：你的欢迎屏读起来顺畅吗？

上一页参考程序的完整输出

```
=====
欢迎来到：方块探险营
玩家：Alex
今日任务：收集 10 个原木
安全提示：不要直直地向下挖
趣味提示：苦力怕也想听课，请保持距离
=====
```

第一课的能力地图



三个小实验

- 只改服务器名字。预期：只有对应的一行发生变化。
- 把“玩家：”后的 print 改成 println。预期：Alex 会到下一行。
- 在边框后增加空行。预期：内容不变，视觉间距变化。

完成后，用自己的话解释每条语句。能改动代码并预测结果，比一字不差地抄下来更有用。

10 再往前走一小步

任选一个作品，写出预期输出

- 基础：写五行“我的生存第一天”，至少使用三次 `println`。
- 进阶：先用 `print` 拼出一行合成式，再用两行说明解释配方。
- 创造：用文字字符画一座小屋，至少七行高，检查每行的空格。

30秒自测

① 引号会出现在输出里吗？② `println` 的换行发生在内容之前还是之后？③ `HelloWorld.java` 中的 `public class` 可以叫 `helloWorld` 吗？

答案：① 不会，引号界定文字。② 在内容之后。③ 不能直接这样改；`public` 类名和文件名必须一致。

下一课：让程序记住变化

今天的文字是固定的。下一课，我们会给生命值、物品数量和玩家名贴上标签，让同一个名字在不同时间保存不同的值。

技术参考：Java 的输出 API 与语法规则见 docs.oracle.com。示例均以 Java 11 为目标验证；故事和插画用于解释概念。