芯语教育课程包 · 教材 · 09

芯语中文编程教材
从零基础到做出作品

17 章 · 每章有目标、例子、逐行解释、动手任务、常见错误与小结

对象	语言	结构	版本
零基础学生	中文 9S	17 章 · 5 单元	v1.1 · 2026-10
教师与家长可陪读	原生窗口 / 3D 场景	章章有练习	

前言

零基础课堂上，学生最先想知道的常常是“我点了以后会发生什么”。如果老师从关键字和符号讲起，两边一错位，课就冷了。本书把这层顺序调过来：先运行，再拆开看，再只改一处，最后自己写。

书里的例子全部用中文 9S。变量、如果、当、函数这些词，读到就能猜到大概意思，学生不必先背一整套英文。需要说明的是，能读懂中文不等于能写出程序；每章后面的动手任务，才是真正要花时间的地方。

建议每次课保留三个动作：先运行一次，再只改一处，最后用自己的话讲清改了什么。看起来慢，实际比一次性抄完更稳。本书按由易到难的顺序编排，但学习进度仍取决于课时、练习量和教师支持，不承诺所有学生达到同一结果。

> **本书怎么用：**教师按章备课，学生按章练习，家长可以陪读“先做再看”和“本章小结”两部分。前 16 章每章的新关键字控制在两个以内；第 17 章集中认识项目组织词，只要求读懂，不要求一次记住。

学习路线：每章只上一个台阶

阶段	章节	新关键字 / 语法	学生要能做到	练习强度
会运行	1—4	输出、变量、参数	打开、运行、停止、只改一处、单步看轨迹	每课 1 次演示 + 2 次自己运行
会读	5—8	选择、如果 / 否则、当、从…到…循环、列表	能预测结果，能指出条件真假和循环退出原因	每课 1 张记录表
会写	9—12	函数、返回、按键、定时器、画布、3D 场景	能写小函数，能画 2D 界面，能改 3D 配置	每课 2 次改造
会做	13—16	不引入新语法，组合前面内容	能读软件/游戏结构，能调试、测试、写出作品说明	每课 1 个完整任务
会组织	17	引入、枚举、结构体、阳层、阴层、五行板块、公共、私有	能读懂完整模板的目录和分层，能找到公共接口	1 次源码走读 + 1 次改造

目录

单元	章	标题	你会做出什么
一 会运行	1	第一次打开与运行	跑通一个示例并截图
	2	顺序与输出	三句话按顺序出现
	3	只改一处	改造前后对比截图
	4	单步与轨迹	一张预测修正表
二 会读	5	变量与输入	三次输入记录
	6	条件分支	两条路线图
	7	循环与退出	循环次数实验表
	8	列表与状态	分数预测表
三 会写	9	函数与复用	两个自己写的函数
	10	事件、按键与定时器	按键控制的小方块
	11	2D 画布	自己画的文字和方块
	12	3D 场景	能跑起来的 3D 画面
四 会做	13	软件解剖：待办清单	能用的软件改造版
	14	游戏解剖：打砖块	能玩的 2D 游戏
	15	3D 游戏解剖：躲避陨石	能玩的 3D 游戏
	16	调试、测试与发布	可复现的作品包
五 会组织	17	项目组织：引入、枚举、结构体与阴阳五行分层	能读懂完整模板的目录与分层

第一单元 · 会运行

第 1 章 第一次打开与运行

学习目标

- 能独立打开芯语创作客户端。
- 能从示例库载入一个作品并运行。
- 能说出作品舞台在哪里、停止按钮在哪里。

先做再看

打开客户端，点“示例库”，选《星际寻宝》，点“运行作品”。你会看到舞台上一句一句出现文字。看完点“停止”。

知识讲解

客户端分四个区域：左边导航、中间源码、下方作品舞台、顶部运行按钮。源码是“写给电脑看的”，舞台是“电脑做给你看的”。运行就是把源码交给运行环境执行一遍。

动手任务

保底：独立完成一次载入—运行—停止，截一张图。**进阶：**换一个示例再跑一遍，说出两个示例的相同点。

常见错误

点了运行没反应：先看是否停在“源码”标签，再向下滚动到舞台。舞台空白：先点运行，再等一秒。

本章小结

会打开、会运行、会停止，是学编程的第一步。别急着改代码，先让结果出现。

第 2 章 顺序与输出

学习目标

- 能说出“顺序执行”是什么意思。
- 能改写一句输出并成功运行。
- 能按“先……再……最后……”复述自己的程序。

代码示例

```
当 开始运行
  输出("大家好，我是小芯")
  输出("今天学习中文编程")
  输出("先看结果，再读代码")
```

逐行解释

第一行是程序入口，意思是“当作品开始运行时”。后面三行依次执行，所以舞台上三句话按这个顺序出现。把第二、三行对调，出现的顺序也会对调。

动手任务

保底：把第一句改成自己的名字并运行。**进阶：**写三句有先后关系的输出，说明为什么不能颠倒。

常见错误

改了没变化：没保存或没重新运行。引号只剩一个：输出文字必须成对使用引号。

本章小结

程序像排队：一句做完才做下一句。顺序是编程里最基础、也最容易被忽视的规则。

第 3 章 只改一处

学习目标

- 能找到源码顶部的“可改配置区”。
- 能只改一个数字或颜色并重新运行。
- 能说出改造前后的差异。

例题

打开《俄罗斯方块 3D》，顶部有一块“可改配置”：方块颜色、下落帧、相机距离。把“方块一色”改成紫色，保存，点“原生运行”，3D 窗口里的方块就变紫了。

为什么只改一处

同时改颜色和速度，画面变了，你无法判断是哪一个参数起作用。只改一处，结果和原因才能一一对应，这叫“控制变量”。

动手任务

保底：只改一个颜色，交改造前后截图。**进阶：**只改一个数字参数，用一句话说明变化。

常见错误

改完窗口没出现：看任务栏，或先停止再运行。改坏无法运行：恢复原值，再只改一个字符重试。

本章小结

改造不是重写，是“找到开关—拨一下—再跑一遍—记录差异”。

第 4 章 单步与轨迹

学习目标

- 能进入单步模式并连点“下一步”。
- 能在轨迹里找到当前执行位置和变量值。
- 能先预测、再验证，记录一次预测错误。

知识讲解

连续运行是一口气跑完，单步是一步一停。每停一次，轨迹就记一笔：执行到哪、变量是多少、条件是真还是假。轨迹是证据，不是装饰。

动手任务

载入《猜数字》，单步 5 次，每次先在纸上写下“猜测次数”和“玩家猜测”的预测值，再点下一步验证。

常见错误

下一步是灰的：程序在等输入或已经结束。轨迹为空：先点“轨迹”标签再单步。

本章小结

会单步，你就能回答“程序为什么走了这一步”，而不是只会看最后结果。

第二单元 • 会读

第 5 章 变量与输入

学习目标

- 能说出变量是“有名字的盒子”。
- 能指出输入写进了哪个变量。
- 能解释“新值会盖掉旧值”。

代码示例

```
变量 玩家猜测 = 0
变量 猜测次数 = 0
玩家猜测 = 选择("请输入你猜的数字")
猜测次数 = 猜测次数 + 1
```

逐行解释

第一、二行创建两个盒子，先放 0。第三行把用户输入的数字放进“玩家猜测”。第四行读出旧值，加 1，再放回同一个盒子。

动手任务

运行猜数字，依次输入 9、2、7，记录每次的“输入值 / 玩家猜测 / 猜测次数 / 输出”。

常见错误

输入字母没反应：数字输入框只收有效数字。变量看不到：答案类变量被隐藏，看轨迹。

本章小结

变量负责“记住”。程序没有变量，就记不住分数、次数和用户的选择。

第 6 章 条件分支

学习目标

- 能判断一个条件是真还是假。
- 能画出“如果 / 否则”两条路线。
- 知道“=”是赋值，“==”才是比较。

代码示例

```
如果（玩家猜测 == 神秘数字）{
  输出("猜中了")
} 否则 {
  如果（玩家猜测 > 神秘数字）{
    输出("大了")
  } 否则 {
    输出("小了")
  }
}
```

逐行解释

先判断“相等”，为真就走“猜中了”；为假再判断“大于”，为真走“大了”，否则走“小了”。两条路只能走一条。

动手任务

保底：画出两条路线图，标出 9、2、7 各走哪边。**进阶：**加一条“输入 0 提示超出范围”的分支。

常见错误

提示反了：比较符号写反。条件永远为真：把“=”当成了比较。

本章小结

条件让程序会“看情况”。写条件前，先用中文说清楚“如果……就……”。

第 7 章 循环与退出

学习目标

- 能读出循环的“继续条件”和“退出条件”。
- 能改次数上限并验证提前结束。
- 能说出死循环为什么会发生。

代码示例

```
变量 猜测次数 = 0
当 (已猜中 == 假 且 猜测次数 < 5) {
  玩家猜测 = 选择("请再猜一次")
  猜测次数 = 猜测次数 + 1
}
```

逐行解释

只要“没猜中”且“次数还不到 5”，循环就继续。循环体里必须更新“猜测次数”，否则条件永远为真。

动手任务

把上限 5 改成 3，连续输入错误数字，观察第 4 次是否还被允许。记录退出原因。

常见错误

循环停不下来：先点停止，再查循环体有没有更新次数。改了上限没生效：源码和积木输入框没同步。

本章小结

循环必须回答两个问题：什么时候继续？什么时候停？

第 8 章 列表与状态

学习目标

- 能说出列表是“装一串东西的抽屉”。
- 能用追加、查询、移除、长度四个动作。
- 能预测多轮循环后的总分。

代码示例

```
变量 敌机X = []
列表追加(敌机X, 100)
列表追加(敌机X, 220)
变量 数量 = 列表长度(敌机X)
变量 第一架 = 列表查询(敌机X, 0)
列表移除(敌机X, 0)
```

逐行解释

空方括号表示空列表。追加把新值放到末尾；长度是数一数；查询按序号取；移除按序号删。序号从 0 开始。

动手任务

打开《打砖块》源码，找到砖块列表和子弹列表，说出各装什么、谁在追加、谁在移除。

常见错误

序号越界：列表长度是 3 时，最大序号是 2。删着删着漏了：从后往前移除更安全。

本章小结

列表让程序能同时管理很多对象：多架敌机、多颗子弹、多条待办。

第三单元・会写

第 9 章 函数与复用

学习目标

- 能区分“定义函数”和“调用函数”。
- 能写一个只含两句输出的函数并调用两次。
- 能说出参数的作用。

代码示例

```
函数 打招呼(名字) {
  输出("你好，" + 名字)
  输出("欢迎来到芯语")
}
打招呼("小芯")
打招呼("同学")
```

逐行解释

“函数 打招呼(名字)”是定义：把两句话包起来，起个名字，括号里的“名字”是参数。下面两次是调用，每次换一个参数。

动手任务

保底：调用现成函数让画面变化。**进阶：**写一个带参数的函数，用两个不同值调用。

常见错误

调用没反应：函数名拼错或没定义。参数对不上：个数和顺序要一致。

本章小结

函数的作用是给一段逻辑一个修改入口，少打几个字只是顺带的好处。

第 10 章 事件、按键与定时器

学习目标

- 能说出“按键触发函数”。
- 能指出定时器在源码哪一段。
- 知道“帧”是画面刷新的单位。

代码示例

```
函数 按键处理(键码) {
  如果 (键码 == 键_左) {
    玩家X = 玩家X - 20
  }
  如果 (键码 == 键_右) {
    玩家X = 玩家X + 20
  }
  请求重绘()
}

设置按键函数(按键处理)
设置定时器(16, 帧刷新)
```

逐行解释

“按键处理”是事件函数：用户按键时才执行。定时器每隔 16 毫秒调用一次“帧刷新”，游戏世界因此动起来。

动手任务

打开《打地鼠》源码，找到方向键选洞、空格敲击、定时器刷新三处，画出“按键→函数→画面”的箭头图。

常见错误

按键没反应：3D/2D 窗口没获得焦点，先点一下窗口。改了速度没变：保存后要重新运行。

本章小结

一个交互作品 = 按键管用户 + 定时器管世界 + 绘制管画面。

第 11 章 2D 画布

学习目标

- 能说出坐标原点在左上角。
- 能画文字、矩形、方块。
- 知道绘制函数第一个参数是画布。

代码示例

```
画布.清屏()
画布.画矩形(0, 0, 480, 640, 深灰, 深灰)
画布.画矩形(100, 200, 80, 80, 青色, 青色)
画布.画文本(20, 20, "分数: 0", 白色, 22)
```

逐行解释

清屏把上一帧擦掉；画矩形前两个数是左上角坐标，中间两个是宽高，后面是填充色和边框色；画文本先坐标、再内容、再颜色、再字号。

动手任务

保底：画一行文字和一个方块。**进阶：**把方块移到屏幕右下角，再画三个不同颜色的方块。

常见错误

什么都不显示：检查是否真的调用了绘制函数、是否在刷新帧里。位置不对：坐标顺序是先 X 再 Y。

本章小结

2D 画布是所有 2D 游戏的地基。会画方块和文字，就能画出分数、生命和按钮。

第 12 章 3D 场景

学习目标

- 能定位创建场景、设置相机、渲染场景三处。
- 能说出立方体和球体怎么加。
- 能改一个 3D 配置并重新运行。

代码示例

```
场景 = 创建3D场景(1400)
设置3D相机(场景, 0, 70, -900)
设置相机注视(场景, 0, 20, 300)
设置3D光源(场景, 0.4, -0.8, -0.3, 0.75)
变量 方块 = 添加立方体(场景, 0, 0, 0, 40, 40, 40, 颜色_青)
渲染3D场景(场景)
```

逐行解释

场景是 3D 世界；相机决定从哪看；注视决定看向哪；光源决定明暗；添加立方体把物体放进去；渲染把 3D 世界画到窗口里。

动手任务

保底：在《躲避陨石 3D》里找到创建场景、相机、渲染三行。**进阶：**改相机距离并说明画面变化。

常见错误

3D 窗口不见：看任务栏，或先停止再运行。画面全黑：相机位置可能在物体背后，先把相机拉远。

本章小结

3D 不是魔法：场景、相机、光源、物体、渲染，五件事说清楚，画面就出来了。

第四单元 · 会做

第 13 章 软件解剖：待办清单

学习目标

- 能说出一个软件的四个组成部分：数据、界面、操作、刷新。
- 能定位待办清单的列表、选中项、完成状态。
- 能改一条待办并重新运行。

软件是怎么搭起来的

待办清单只有四件事：一个装事项的列表、一个装完成状态的列表、一个“当前选中第几项”的变量、一个把界面画出来的函数。按键改变数据，数据改变后请求重绘，界面就变了。

代码片段

```
变量 事项 = []
变量 已完成 = []
变量 选中 = 0

函数 按键处理(键码) {
  如果 (键码 == 键_空格) {
    已完成[选中] = 真
  }
  请求重绘()
}
```

动手任务

保底：把一条待办文字改成自己的作业。**进阶：**把“空格完成”改成“空格在真/假之间切换”。

本章小结

软件 = 数据 + 界面 + 操作 + 刷新。会这四件事，就能做工具，而不是只会做游戏。

第 14 章 游戏解剖：打砖块

学习目标

- 能说出一个游戏的四要素：玩家、世界、碰撞、反馈。
- 能找到挡板、球、砖块三组数据。
- 能改一个游戏参数并验证。

游戏主循环

每一帧做四件事：球移动、球撞墙/挡板反弹、球撞砖块消除并加分、球出界扣命。全部写在一个“帧刷新”函数里，由定时器反复调用。

代码片段

```
函数 帧刷新() {
  如果 (游戏结束 == 假) {
    更新球()
    检测砖块()
    球出界()
    更新粒子()
  }
```

```
  请求重绘()
}
```

动手任务

保底：把初始生命从 3 改成 5。**进阶**：把砖块列数从 7 改成 6，观察布局变化。

本章小结

游戏的核心是一个每帧重复的循环，画面效果都挂在这个循环上。先写循环，再往里加规则。

第 15 章 3D 游戏解剖：躲避陨石

学习目标

- 能说出 3D 游戏的“世界坐标”和“玩家坐标”。
- 能找到陨石生成、移动、碰撞、计分四处。
- 能改陨石速度并体验难度变化。

世界怎么动

玩家战机停在靠近相机的位置，陨石从远处 z=1050 向相机方向飞来。每帧减少 z 值，靠近玩家时判断 x、y、z 三个方向的差值，进入范围就判定撞击。

代码片段

```
如果 (石Z > 玩家Z - 90 且 石Z < 玩家Z + 90) {
  变量 差X = 玩家X - 石X
  变量 差Y = 玩家Y - 石Y
  如果 (差X > -60 且 差X < 60 且 差Y > -50 且 差Y < 50) {
    爆炸(石X, 石Y, 石Z)
    生命 = 生命 - 1
  }
}
```

动手任务

保底：把初始生命改成 8，玩一局。**进阶**：把陨石生成间隔从 18 改成 30，比较难度。

本章小结

3D 游戏仍然是循环 + 数据 + 碰撞 + 反馈，只是坐标多了一个 z。

第 16 章 调试、测试与发布

学习目标

- 能用“一次只改一处”定位问题。
- 能写三条测试用例并记录结果。
- 能导出 `.9s` 并写清运行说明。

调试四步

第一步看提示，第二步回到上一版，第三步只改一处，第四步重新运行验证。四步走完还不行，再求助，不要连续乱改。

测试三条

正常路径：按说明能玩/能用。边界路径：0 分、0 生命、空列表会怎样。错误路径：乱按键、输入超范围会怎样。

发布清单

1. 源码是中文 9S，文件名中文，能由对应虚拟机运行。
2. 作品文档写清玩法、操作、配置项和运行方式。
3. 至少3 张截图：开始、进行中、结束/胜利。
4. 说明哪些是模板、哪些是自己改的，AI 参与了哪部分。

本章小结

能运行不算完成，能解释、能复现、能交给别人运行，才算作品。

第五单元 • 会组织

第 17 章 项目组织：引入、枚举、结构体与阴阳五行分层

这一章是选读，放在教材后段。客户端首页和创作页不展示这套术语，先把运行、改造、调试学会，再回来读这一章。

学习目标

- 能说出“引入、枚举、结构体”各自解决什么问题。
- 能区分阳层的公开接口和阴层的私有实现。
- 能按“金木水火土”找到完整模板里的五类职责。

先做再看

打开示例库里的《俄罗斯方块》2D 完整源码。先不要从头读到尾，只看文件开头的注释和分区标题：引入、枚举、结构体、阳层、阴层、金板块、木板块、水板块、火板块、土板块。你会发现问题被分进了不同的抽屉。需要说明的是，3D 模板目前按“场景、相机、游戏逻辑”分节，还没有采用这套五行板块；读 3D 模板时按它自己的分区读，不要硬套。

17.1 引入：先借工具

完整项目不会把所有东西从零写一遍。开头用“引入”把标准库借进来，后面才能用列表、画布、窗口这些能力。

```
引入 标准.基础
引入 标准.集合
引入 标准.图形
```

可以把它理解成上课前把教材、练习本、画笔放到桌上。没有引入，后面的函数就找不到工具。

17.2 枚举：给状态起名字

游戏有“未开始、运行中、暂停、结束”几种状态。直接写 0、1、2、3 过几天就忘了含义，用枚举就能让代码自己说明自己。

```
枚举 游戏状态 {
    未开始
```

```
    运行中
    已暂停
    游戏结束
}
```

使用时要带上枚举名，例如“游戏状态.运行中”。这样读代码的人不用猜数字。

17.3 结构体：把一组数据装进一个盒子

一个游戏不只有一个分数，还有生命、关卡、当前方块、下一块。把它们散落在十几个变量里，函数参数会越来越长。结构体把这些字段收在一起。

```
阳层 {
    结构体 游戏数据 {
        分数 = 0
        生命 = 3
        状态 = 空
    }
}
```

用的时候写“数据.分数”“数据.生命”。结构体不产生新的执行动作，它的作用是把相关字段组织在一起，让函数参数不必越写越长。

17.4 阳层与阴层：公开的与内部的

阳层放对外公开的东西：常量、结构体、公共函数。别人要调用、要改配置，就看阳层。阴层放内部实现：私有函数、临时计算、细节处理。外部不需要知道它们怎么做的。

```
阳层 {
    常量 窗口宽度 = 480
    公共 函数 取分数(数据 -> 游戏数据) -> 整数型 {
        返回 数据.分数
    }
}
阴层 {
    私有 函数 _加分(数据 -> 游戏数据，分值 -> 整数型) {
        数据.分数 = 数据.分数 + 分值
    }
}
```

“阳阴合一”说的是一个完整模块两层都有：外面能看见接口，里面有自己的实现。只写阳层，程序跑不起来；只写阴层，别人不知道怎么用。两层都在同一份源码里，一层对外，一层对内。

17.5 金木水火土：五类职责，不是五步流程

采用这套结构的完整模板，会把公共接口按职责分成五个板块。以《俄罗斯方块》2D 完整版为例，源码里的分工是：

板块	负责什么	典型公共函数
金	初始化与校验	初始化游戏、校验游戏状态、重启游戏
木	方块生成与变换	生成新方块、旋转方块

板块	负责什么	典型公共函数
水	移动与下落	左移、右移、下落、硬下落
火	消行、计分与等级	检查消行、计算分数、计算等级
土	渲染与界面	启动图形游戏、绘制画面、显示结束

这五个名字借用了五行，在代码里只表示五类职责，不规定执行顺序。真正运行时，还是由主循环按需要调用。教学时先把职责讲清，再提“金木水火土”这个名字，学生就不会把标签当魔法。

17.6 形与理：积木和源码是同一份东西

A 轨的积木和源码共享同一棵语法树（AST）。改积木就是改源码，改源码也会同步到积木。完整原生 `.9s` 没有积木视图，但“先看结构、再找接口、最后改一处”的读法是一样的。

17.7 读完整模板的顺序

1. 先看顶部配置区：哪些常量可以改。
2. 再看阳层：公开了哪些结构体和公共函数。
3. 再看阴层：每个公共函数背后调用了哪些私有函数。
4. 最后看金木水火土板块：五类职责各在哪些行。
5. 只改一个公共常量，重新运行，确认变化。

动手任务

保底：在《俄罗斯方块》完整源码里找到阳层、阴层、金板块、土板块四处，并截图标注。**标准：**列出金、木、水、火、土各一个公共函数。**进阶：**只改阳层里的一个配置常量，重新运行并记录变化。

常见错误

把五行当成运行顺序：它只是职责分类，程序不会按金木水火土依次执行。把私有函数写进阳层：外部会看到不该看到的细节。结构体字段乱放：先问“这个字段是不是这份数据的一部分”，再决定放哪。以为阳层不能改：阳层的常量本来就是给改造用的，改完要重新运行验证。

本章小结

会写单个文件之后，下一步是学会把代码放进合适的抽屉。引入借工具，枚举起名字，结构体装数据，阳层公开接口，阴层藏实现，五行板块分职责。这样做的用处很实际：完整项目能被别人读懂，也方便自己下次修改。

附录 A • 9S 语法速查

写法	作用
`变量 分数 = 0`	创建变量并赋初值
`输出("文字")`	向舞台输出一句话
`如果（条件）{ … } 否则 { … }`	条件为真/假走不同分支
`当（条件）{ … }`	条件为真时重复执行

写法	作用
`从 1 到 5 循环 序号 { … }`	固定次数循环
`函数 名字(参数) { … }`	定义可复用的代码块
`列表追加/查询/移除/长度`	管理一串数据
`创建窗口 / 九S画布 / 设置定时器`	原生窗口程序入口
`创建3D场景 / 添加立方体 / 渲染3D场景`	3D 场景与物体

附录 B · 调试检查表

1. 程序停在哪个标签？源码、积木还是舞台？
2. 改的是当前项目吗？保存了吗？重新运行了吗？
3. 引号、括号、花括号成对吗？
4. 变量名前后完全一致吗？
5. 上一次能运行是什么版本？先回去，再一次只改一处。

附录 C · 作品文档模板

```
作品名：_____
一句话说明：_____
给谁玩 / 给谁用：_____
操作方法：_____
关键规则：_____
我改过的源码位置：_____
AI 参与部分与我的验证：_____
已知限制：_____
运行方式（九S虚拟机 / 4D 虚拟机 / 客户端）：_____
```

附录 D · 关键字与语义索引

关键字 / 语法	首次出现	语义	常见误用
`输出`	第 2 章	把一句话显示到作品舞台	忘记引号或引号不成对

关键字 / 语法	首次出现	语义	常见误用
变量	第 5 章	创建一个有名字的存储位置	把变量名写成两个不同的词
空 / 真 / 假	第 5—6 章	空表示还没有值；真/假表示条件成立或不成立	把“假”写成字符串“假”
文本 / 取整 / 随机整数	第 8—10 章	把数字转成文字、取整数、生成指定范围的随机数	随机范围写反；把文本当数字比较
常量	第 17 章	运行前固定好的值，适合做窗口尺寸、配置项	把需要变化的量写成常量
选择	第 5 章	暂停程序，等待用户输入并写入变量	把输入值当成最终结果
如果 / 否则	第 6 章	条件为真走一条路，为假走另一条	把“=”当成比较
当	第 7 章	条件为真时重复执行循环体	循环体里不更新状态，导致停不下来
从…到…循环	第 7 章	按固定次数重复	把终点当成“多执行一次”
列表追加/查询/移除/长度	第 8 章	向列表放、取、删、数元素	序号越界；边循环边删除导致漏项
函数 / 返回	第 9 章	把一段逻辑包起来，可反复调用；返回把结果交给调用处	函数名拼错；有返回值却忘了接收
按键函数 / 定时器	第 10 章	按键时执行；定时器按固定间隔反复执行	窗口没有焦点就按；改了间隔没重新运行
创建窗口 / 九S画布	第 11 章	建立真实窗口和绘图画布	绘制函数没设置，窗口空白
创建3D场景 / 渲染3D场景	第 12 章	建立 3D 世界并把它画到窗口	相机在物体背后，画面全黑
引入	第 17 章	把标准库或模块借进来	没引入就直接用列表/画布能力
枚举	第 17 章	给一组状态起名字	使用时忘记写“枚举名.成员”
结构体	第 17 章	把相关字段收进一个数据盒子	字段散落在十几个变量里
阳层 / 阴层	第 17 章	阳层放公开接口，阴层放私有实现	把私有细节写进阳层；以为阳层不能改
公共 / 私有	第 17 章	标记函数对外可用或仅内部可用	公共函数没有明确参数与返回
金 / 木 / 水 / 火 / 土板块	第 17 章	按职责划分的五类模块	当成五个执行步骤或玄学

关键字 / 语法	首次出现	语义	常见误用
`-> 类型标注`	第 17 章	说明参数和返回值的数据类型	参数顺序与类型写反

教材使用建议：每章 1 次课（90 分钟）或拆成 2 次 40 分钟课。第 1—4 章建议全员通过；第 5—8 章可安排 1—2 次补课；第 9—12 章按兴趣分组；第 13—16 章进入项目制；第 17 章作为完整模板的选读课。不要跳章，也不要用“看老师做”替代“自己运行一次”。

版权与合规：本书是教学材料，不表示已获得教育行政部门审定或完成教育应用备案。教材中的模板、截图与示例代码版权归绵阳九艾斯科技有限公司所有；课堂使用时请遵守学校规定与未成年人保护要求，不收集学生个人信息。本书按教学顺序编排，不承诺所有学生达到同一学习结果。