芯语教育课程包 · 高校通识版 · 02

教师课程包
12 课时教学方案 · v1.1

每课 90 分钟 · A/B 双轨 · 含任务单、常见错误、差异化与评价证据

课程	课时	结构	平台	版本
中文编程与计算思维	12 次 × 90 分钟	A 轨可解释程序 B 轨原生 9S 源码	芯语创作客户端 3.0	v1.1 · 2026-10

v1.1 双轨说明：第 1—6 课以 A 轨（猜数字、答题、星际收集）为主，使用积木、单步、轨迹；第 7—8 课切到 B 轨，阅读原生贪吃蛇 3D / 俄罗斯方块 3D `.9s` 源码，并由 4D 虚拟机运行 3D 窗口；第 9—12 课双轨并行。教师必须明确告诉学生：B 轨没有积木视图，不能拿“积木核对”当 B 轨证据。

课程总览

课次	主题	核心概念	关键任务	学习证据
1	认识中文编程：从一句话到两类作品	顺序、输出、循环、形理同源；原生 VM 概览	运行星际收集；观察原生贪吃蛇窗口；说出 A/B 差异	运行截图 + A/B 差异说明
2	单步观察：程序是怎样一步步执行的	单步、轨迹、变量、输出	单步猜数字前 5 步并预测	轨迹截图 + 预测表
3	输入与变量：谁记住了我的猜测	选择、变量回写、数字输入	输入 9、2、7 并记录变量	三次输入记录表
4	条件分支：程序为什么会说“大了”	如果 / 否则、比较、布尔、且 / 或	验证三种输入的真 / 假路径	真值表 + 分支解释
5	循环与退出：怎样重复又不会停不下来	当循环、次数上限、安全保护	把 5 次上限改成 3 次并验证	循环次数实验表
6	循环中的状态：分数为什么在变	从…到…循环、加分、状态	改轮次与分值，观察分数	改造前后对比表
7	B 轨：原生 9S 源码阅读——贪吃蛇 3D	3D 场景、立方体蛇身、列表、函数、按键、定时器、顶部配置	定位源码行为并启动 4D 窗口	源码行定位 + 4D 运行记录
8	B 轨：原生 9S 源码改造——俄罗斯方块 3D	形状矩阵、立体棋盘、旋转、消行、落点投影	改一个顶部配置并重新运行	改造前后 + 4D 窗口截图

课次	主题	核心概念	关键任务	学习证据
9	AI 辅助创作与模板路由	提示词、原生模板、内嵌预览、教学原型	生成/路由一个作品并验收来源	提示词 + 路由结果 + 验收单
10	双轨调试：一次只改一处	A：积木/变量；B：源码常量/函数	完成 A 或 B 轨两次改造	改造前后 + 回归记录
11	作品表达：让别人也能运行	导出 `.9s`、作品工程、分享文本	导出作品包并与同伴互测	作品包 + 互评单
12	综合项目答辩：用证据讲清作品	A/B 轨自选；证据与边界说明	3—5 分钟演示与答辩	作品 + 分轨量规 + 反思

01 认识中文编程：从一句话到两类作品

2 学时

课标关联 计算思维；数字化学习与创新；必修 1 程序与算法初步

驱动问题 中文代码怎样变成可解释程序，又怎样变成原生 9S 游戏？

学习目标

- 能说出首页三个入口，并区分 A 轨（积木/单步）与 B 轨（原生源码/VM）。
- 能运行《星际收集》，指出开始、变量、循环、加分四类节点。
- 能观察原生贪吃蛇窗口，说出“它由九S虚拟机运行，不是积木舞台”。

教学流程

环节	时间	活动
导入	5′	教师运行《星际收集》，再启动原生贪吃蛇窗口；问“两个作品都是中文代码，为什么证据不同？”
观察	15′	A 轨：载入《星际收集》，在源码区找“当 开始运行”“变量 分数 = 0”“从 1 到 4 循环 回合”“加分 1”。B 轨：观察原生贪吃蛇窗口标题与操作。
讲解	10′	A 轨：积木与源码共享 AST；B 轨：完整 `.9s` 由原 VM 执行，没有积木/单步。两轨都算中文编程，但证据不同。
实践	30′	任务一：把“加分 1”改成“加分 5”，运行并记录分数。任务二：把循环终点改成 2，预测并验证路线。任务三（观察）：打开原生贪吃蛇源码，找 `创建窗口`。
分享	15′	抽取两组展示，要求用“我改了……所以……”句式说明因果，并注明 A/B 轨。
总结	10′	出口条：写出 A 轨四类节点，以及 B 轨一个原生窗口 API。

常见错误与支架

- 点击运行没有画面：检查是否停留在源码/积木标签，向下滚动到“作品舞台”。
- 只改源码不同步：提醒点击“同步到可视 AST”或切换到积木确认。
- 把“加分”当成变量赋值：强调加分是教学扩展动作，会更新分数状态。

差异化

基础组只改分值；进阶组同时改轮次与分值，先预测分数再运行；挑战组尝试把循环改成 6 轮并解释为什么分数与轮次一致。

证据与作业

提交运行截图 + 一句因果解释。作业：用 5 句自然语言写出自己想做的游戏规则，不写代码。

02 单步观察：程序是怎样一步步执行的

2 学时

课标关联 计算思维；必修 1 变量与程序执行过程

驱动问题 不看结果，能不能提前说出下一步会发生什么？

学习目标

- 能使用“单步 / 下一步”执行程序，并说出当前节点名称。
- 能在“轨迹”中读取变量快照和条件结果。
- 能先预测、再验证，修正自己的错误预测。

教学流程

环节	时间	活动
导入	5′	教师连续运行猜数字，学生只看到结果；提问“中间发生了什么”。
观察	15′	学生载入《猜数字》，点击“单步”，每点一次“下一步”前先在纸上写出预测的变量值。
讲解	10′	解释轨迹三要素：执行了哪条命令、条件是真还是假、变量快照是什么；强调“轨迹是证据，不是装饰”。
实践	35′	单步执行前 5 个节点，记录“神秘数字（隐藏）/猜测次数/玩家猜测/已猜中”的变化；然后复位，改为连续运行，比较两种模式。
分享	15′	展示一份轨迹，要求指出“哪一步让猜测次数变成 1”。
总结	10′	出口条：单步与连续运行各适合什么场景？

常见错误与支架

- “下一步”灰色：程序正在等待输入或已经结束，先看舞台提示。
- 轨迹为空：先点击“轨迹”标签，再执行单步。
- 把变量快照当成最终值：提醒快照是“执行到该节点后的那一刻”。

差异化

基础组跟随教师同步单步；进阶组独立预测 8 步；挑战组用轨迹解释“为什么第 6 步进入如果分支”。

证据与作业

提交轨迹截图 + 预测修正表。作业：用自己的话解释“变量快照”和“最终结果”的区别。

03 输入与变量：谁记住了我的猜测

2 学时

课标关联 计算思维；必修 1 数据输入、处理与输出

驱动问题 我在输入框里输入的数字，程序把它放到了哪里？

学习目标

- 能说明“选择”节点会等待输入，并把值写入指定变量。
- 能完成 9、2、7 三次输入，记录玩家猜测与猜测次数。
- 能区分“输入值”“变量值”“输出文本”。

教学流程

环节	时间	活动
导入	5′	教师输入一个数字，问“这个数字现在存在于程序的哪里？”
观察	15′	运行猜数字，观察输入框下方“输入后写入变量：玩家猜测”。
讲解	10′	变量是命名的存储位置；选择节点负责暂停、读取输入、回写变量；答案类变量在舞台自动隐藏，但调试面板可以看到。
实践	40′	依次输入 9、2、7，每轮填写“输入值 / 玩家猜测 / 猜测次数 / 输出”。最后把神秘数字改成 4，再跑一轮。
分享	10′	对比两组记录，找出“猜测次数为什么每次加 1”。
总结	10′	出口条：用一句话说明输入、变量、输出三者的关系。

常见错误与支架

- 输入字母后无法继续：数字输入框只接受有效数字，提示“请输入有效数字”。
- 把“玩家猜测”与“神秘数字”混淆：舞台只显示公开变量，答案变量被隐藏。
- 忘记按确认：输入后需点击确认或按 Enter。

差异化

基础组按 9→2→7 完成；进阶组自己设计三次能覆盖“大、小、中”的输入；挑战组改变量初始值并解释影响。

证据与作业

提交三次输入记录表。作业：设计 4 个测试用例，说明每个用例想验证什么。

04 条件分支：程序为什么会说“大了”

2 学时

课标关联 计算思维；必修 1 分支结构与逻辑运算

驱动问题 同样的输入，为什么程序会走不同分支？

学习目标

- 能读懂“如果（条件）{…} 否则 {…}”并说出条件的真假。

- 能使用 ==、>、< 比较数字，使用“且 / 或”组合条件。
- 能根据轨迹解释“大了 / 小了 / 猜中”三种结果。

教学流程

环节	时间	活动
导入	5′	展示三条输出“大了 / 小了 / 猜中”，让学生猜它们分别由什么条件触发。
观察	15′	运行输入 9，单步观察“玩家猜测 == 神秘数字 → 假”“玩家猜测 > 神秘数字 → 真”。
讲解	10′	分支是“条件为真走一边，为假走另一边”；比较运算符产生布尔值；嵌套如果用于二次判断。
实践	40′	完成真值表：输入 9、2、7 时两个条件分别真假，实际走哪条分支。改写“大了”“小了”的提示文字并重新运行。
分享	10′	学生用“因为……所以……”解释一条轨迹。
总结	10′	出口条：写出一个包含“且”的条件，并说明两个子条件。

常见错误与支架

- 把“=”当比较：9S 条件中使用“==”判断相等，单个“=”用于赋值。
- 条件写反：先写出自然语言“如果猜测大于答案”，再逐字翻译。
- 嵌套过深：先用两个独立如果，再合并；保持缩进清晰。

差异化

基础组完成三行真值表；进阶组把“大小提示”改成更友好的语言；挑战组增加“输入 0 时提示超出范围”的分支。

证据与作业

提交真值表 + 分支解释。作业：为一个“答题闯关”写两条判断规则。

05 循环与退出：怎样重复又不会停不下来

2 学时

课标关联 计算思维、信息社会责任；必修 1 循环结构与算法安全

驱动问题 为什么程序要限制猜测次数？循环什么时候必须停？

学习目标

- 能读懂“当（条件）{…}”和“从…到…循环”的区别。
- 能修改次数上限并验证循环提前结束。
- 能说明死循环风险和 200 次安全保护的意义。

教学流程

环节	时间	活动
导入	5′	提问：如果没有次数限制，猜数字会怎样？
观察	15′	查看“当（已猜中 == 假 且 猜测次数 < 5）”，指出两个退出条件。
讲解	10′	循环需要“继续条件”和“状态变化”；次数变量必须在循环体内更新，否则可能永不退出。
实践	40′	把上限从 5 改成 3，连续输入错误数字，观察第 4 次是否还允许输入；记录循环次数，解释退出原因。
分享	10′	对比“猜中退出”和“次数用完退出”两种路径。
总结	10′	出口条：写出“最多 3 次”的循环条件，并指出哪个变量保证退出。

常见错误与支架

- 改了上限但没改条件：检查源码中的数字与积木输入框是否同步。
- 循环体没有更新次数：解释“条件永远为真”的后果。
- 认为 200 次保护是游戏规则：说明它是运行环境的安全阀。

差异化

基础组改上限并验证；进阶组把“次数上限”写进输出提示；挑战组设计“3 次内猜中奖励、超过则结束”的规则。

证据与作业

提交循环次数实验表。作业：用自然语言写出一个“重复直到满足条件”的生活例子。

06 循环中的状态：分数为什么在变

2 学时

课标关联 计算思维、数字化学习与创新；必修 1 算法与程序实现

驱动问题 每循环一次，程序怎样记住之前的结果？

学习目标

- 能说明“从…到…循环”的起点、终点和循环变量。
- 能观察加分动作对分数变量的累积影响。
- 能通过改造轮次与分值验证“状态随循环累积”。

教学流程

环节	时间	活动
导入	5′	运行星际收集，问“分数是每轮重新开始，还是一直累加？”
观察	15′	单步执行一轮，查看分数变量与循环变量“回合”。
讲解	10′	循环变量记录第几轮；分数是跨轮保持的状态；两者都在轨迹中可见。

环节	时间	活动
实践	40′	把轮次改成 2、分值改成 5，先预测最终分数再运行；再改成 4 轮、2 分，验证预测。
分享	10′	用表格展示“轮次 × 分值 = 最终分数”的关系。
总结	10′	出口条：解释“状态”和“临时变量”的区别。

常见错误与支架

- 只改分值没改轮次：提醒一次只改一处，先验证单一变量。
- 变量名拼写不一致：源码与积木中的名称必须完全一致。
- 把循环变量当成分数：在轨迹中分别指出两个变量。

差异化

基础组完成 2 轮 × 5 分；进阶组设计 3 组“轮次 / 分值”组合并预测；挑战组增加“达到 10 分就输出胜利”的条件。

证据与作业

提交改造前后对比表。作业：设计一个 6 轮、每轮 3 分的收集规则，写出预期分数。

07 B 轨：原生 9S 源码阅读——贪吃蛇 3D

2 学时

课标关联 计算思维、数字化学习与创新；必修 1 事件与交互

驱动问题 完整中文源码怎样变成 3D 窗口里会动的蛇？

学习目标

- 能在完整 `.9s` 源码中定位顶部配置区、`创建3D场景`、`设置3D相机`、`设置按键函数`、`渲染3D场景` 与游戏逻辑。
- 能启动 4D 虚拟机运行贪吃蛇 3D，并说明“源码—3D 窗口行为”的对应。
- 能说清 B 轨为什么没有积木视图和逐条单步。

教学流程

环节	时间	活动
导入	5′	直接运行贪吃蛇 3D 窗口，问“这个 3D 棋盘和这条蛇在源码哪里？”
观察	15′	在源码中找 `创建3D场景`、`设置3D相机`、`创建蛇块`、`设置按键函数`、`设置定时器`、`渲染3D场景`、`移动一格`。
讲解	10′	原生源码由 4D 虚拟机编译执行；B 轨没有积木 AST 映射，也没有逐条单步，证据是源码行与 3D 窗口。
实践	35′	完成“源码行—功能—窗口表现”对应表；启动 4D 窗口；改蛇身颜色或移动间隔，保存并重启验证。

环节	时间	活动
分享	15′	对比 A 轨轨迹证据与 B 轨源码/4D 证据，强调不能混用。
总结	10′	出口条：写出 3 个 3D 图形 API 和 2 个顶部配置项。

常见错误与支架

- 找不到积木：B 轨不提供积木视图，这是设计边界；不要用 A 轨标准要求。
- 3D 游戏窗口没出现：检查任务栏、4D 虚拟机是否启动、是否被“停止”中断。
- 修改后无法运行：先看源码语法与资源路径，再启动 4D 窗口。

差异化

基础组定位 5 个 API；进阶组解释配置区怎样影响 3D 画面；挑战组改相机高度并说明视角变化。

证据与作业

提交源码行对应表 + 4D 窗口截图。作业：列出 5 个原生 9S 3D/窗口 API 并说明作用。

08 B 轨：原生 9S 源码改造——俄罗斯方块 3D

2 学时

课标关联 计算思维、信息社会责任；必修 1 程序调试

驱动问题 只改一个顶部配置，为什么 3D 方块的行为会变？

学习目标

- 能指出俄罗斯方块 3D 源码中的顶部七色配置、4×4 形状矩阵、立体棋盘、旋转踢墙、消行计分与落点投影。
- 能只改一个顶部配置并重新启动 4D 窗口，记录改造前后差异。
- 能说明完整源码为何超出 A 轨积木表达范围，不能用假积木替代。

教学流程

环节	时间	活动
导入	5′	运行俄罗斯方块 3D，问“方块颜色、下落速度、落点投影分别可能在哪段源码？”
观察	15′	浏览源码结构：顶部配置区、`初始化形状`、`碰撞`、`旋转方块`、`锁定方块`、`消行`、`刷新幽灵`、`刷新预览`。
讲解	10′	原生源码由 4D 虚拟机执行；B 轨的改造单位是源码配置/函数，不提供积木拖拽或单步轨迹。
实践	40′	只改一处：方块一色、基础下落帧或幽灵投影开关；保存源码，重启 4D 窗口，截图/记录改造前后；若语法错误，按提示修复。
分享	10′	展示“改了哪一行、3D 画面哪里变了”，不允许只展示结果。
总结	10′	出口条：写出原生源码改造的三步（定位—改一处—4D 验证）。

常见错误与支架

- 一次改多处：无法判断因果，回到上一版只改一处。
- 把源码改动保存到临时文件：必须在工作台保存后再原生运行。
- 以为有积木：B 轨没有积木视图，直接编辑完整 `.9s` 源码。

差异化

基础组改方块颜色；进阶组改下落帧并比较手感；挑战组关闭落点投影并说明对操作判断的影响。

证据与作业

提交改造前后源码差异 + 4D 窗口截图。作业：写一段 200 字源码阅读笔记，指出 3 个关键函数。

09 AI 辅助创作与模板路由

2 学时

课标关联 信息意识、计算思维、信息社会责任；必修 1 AI 与问题求解

驱动问题 AI 是在写完整引擎，还是在路由到一个可信模板？

学习目标

- 能写出包含目标、规则、输入、反馈和限制的需求描述。
- 能判断系统返回的是原生 9S 模板、内嵌预览还是教学原型，并说明依据。
- 能对 AI 参与部分做声明、验收和伦理判断。

教学流程

环节	时间	活动
导入	5′	对比“做个游戏”与“做一个完整的俄罗斯方块，有旋转、消行、音效”。
观察	15′	生成一个俄罗斯方块需求，观察界面是否显示“原生 9S 源码模板”；再生成五子棋/2048，观察“内嵌预览”提示。
讲解	10′	三种来源：原生模板、内嵌预览、教学原型；白名单只保证协议安全，不保证语义完全正确。
实践	40′	写提示词，生成/路由一个作品；验收：来源标签？能否运行？源码是否中文？AI 声明是否完整？
分享	10′	展示一条“把预览说成原生源码”的错误表述，并纠正。
总结	10′	出口条：写出原生模板、内嵌预览、教学原型三者的区别。

常见错误与支架

- 把“AI 生成”理解为模型从零写完整引擎：说明完整游戏来自已审核原生模板。
- 把内嵌预览说成原生 9S：查看界面标签与“完整 9S 源码”入口。
- 不做 AI 声明：在作品文档中写明模板路由、修改与验证。

差异化

基础组完成一次原生模板路由；进阶组比较原生/预览两种结果；挑战组写一份“AI 使用与来源说明”。

证据与作业

提交提示词、路由结果、验收单。作业：把一条模糊需求改成可验收需求，并标注期望来源（原生/预览/原型）。

10 双轨调试：一次只改一处

2 学时

课标关联 计算思维；必修 1 程序调试与优化

驱动问题 不同轨道的“证据”不同，怎样证明结果真的变了？

学习目标

- A 轨：能根据错误行号、单步和轨迹定位问题，只改一个变量/条件。
- B 轨：能在完整源码中定位常量/函数，只改一处并用原 VM 验证。
- 能说明两条轨道的证据不可互换。

教学流程

环节	时间	活动
导入	5′	展示 A 轨错误行与 B 轨源码行，问“各自的证据在哪里？”
观察	10′	教师分别演示 A 轨“单步—轨迹—改条件”和 B 轨“源码定位—保存—VM 运行”。
讲解	10′	A 轨调试顺序：错误行→参数→单步→改一处；B 轨：源码行→常量/函数→保存→VM 窗口验证。
实践	45′	A 站：改神秘数字/次数上限；B 站：改俄罗斯方块 3D 的方块颜色或基础下落帧。两站都记录改造前后。
分享	10′	展示一条证据，说明它为什么只适用于所在轨道。
总结	10′	出口条：写出 A/B 两轨各自的调试四步。

常见错误与支架

- 用 B 轨源码去对照 A 轨积木：轨道混用，证据无效。
- 一次改多个参数：无法判断因果，回到上一版重来。
- 改完不保存：A 轨先同步，B 轨先保存源码，再运行。

差异化

基础组完成本轨两个实验；进阶组做三次对照；挑战组写一份“分轨改造报告”供同伴复现。

证据与作业

提交 A 轨轨迹/对比或 B 轨源码差异/VM 截图 + 回归记录。作业：在自选轨道完成一次完整改造。

11 作品表达：让别人也能运行

2 学时

课标关联 数字化学习与创新；必修 2 信息系统的表达与交流

驱动问题 离开你的电脑，别人还能运行并看懂你的作品吗？

学习目标

- 能导出 A 轨源码/工程或 B 轨完整 `.9s`，并说明运行方式。
- 能写一份包含玩法、来源（原生模板/预览/教学子集）、AI 使用说明与限制的作品文档。
- 能对同伴作品给出基于证据的反馈。

教学流程

环节	时间	活动
导入	5′	提问：如果你的作品只能在本机运行，它算完成了吗？
观察	10′	展示分享弹窗：A 轨源码/工程/分享文本；B 轨完整 `.9s` 与原 VM 运行说明。
讲解	10′	作品文档五要素：目标用户、玩法规则、关键逻辑、来源声明（原生/预览/教学）、AI 参与与限制。
实践	45′	导出作品，和同伴交换；同伴按说明运行；填写互评单；明确指出是原生 9S 还是内嵌预览。
分享	10′	展示一条最有价值的同伴反馈。
总结	10′	出口条：写出“可复现作品”的三个条件，并注明轨道。

常见错误与支架

- 只分享截图：截图不能运行，必须导出源码/工程或 `.9s`。
- 把内嵌预览写成原生源码：按界面标签与源码入口核对。
- 文档只写“很好玩”：必须写清规则、来源、输入和反馈。

差异化

基础组完成一次导出与互测；进阶组根据反馈改一处并复测；挑战组制作 1 页分轨作品说明海报。

证据与作业

提交作品包 + 互评单。作业：根据反馈完成一次迭代，并记录改动与轨道。

12 综合项目答辩：用证据讲清作品

2 学时

课标关联 综合素养；必修 1 / 2 综合应用

驱动问题 你能不能用 3—5 分钟，让别人相信这个作品是你理解并改造过的？

学习目标

- 完成一个可运行、可解释、可复现的综合作品（A 轨教学子集或 B 轨原生 9S）。

- A 轨能用轨迹/变量证据，B 轨能用源码行/VM 窗口证据解释至少 3 个关键节点。
- 能说明 AI 参与程度、作品来源（原生/预览/教学）与限制。

教学流程

环节	时间	活动
准备	10′	检查作品、源码、导出包、答辩提纲；B 轨确认 VM 能启动。
展示	35′	每人/每组 3—5 分钟：轨道—目标—演示—关键证据—来源/AI 声明—改进。
提问	20′	教师与同伴按轨道提问：A 轨问轨迹/变量；B 轨问源码行/VM 行为。
评分	10′	按分轨四维量规独立评分，取教师与同伴平均。
反思	15′	填写反思表：最有价值的失败、下一次会怎么改。

答辩必答问题

1. 你的作品属于 A 轨还是 B 轨？为什么？
2. A 轨：请指出一个条件分支并展示轨迹；B 轨：请指出源码中的窗口/绘制/按键函数并展示 VM 窗口。
3. 你的作品是原生 9S、内嵌预览还是教学原型？依据是什么？
4. AI 参与了哪一部分？你做了哪些验证和改造？
5. 如果只给你 10 分钟再改一处，你会改哪里？为什么？

评分与证据

使用《04 评价量规与课堂实施》中的分轨项目量规。答辩不是背稿，A 轨必须现场运行，B 轨必须现场启动原 VM 并接受追问。

课堂通用任务单

字段	填写要求
本课驱动问题	用一句话写出今天要解决的问题。
我预测	运行前写出预期结果或变量值。
我改了	写清改了哪个节点、哪个参数，只写一处。
实际结果	记录输出、变量、轨迹或截图。
为什么	用“因为……所以……”解释因果。
还没懂	写一个仍然不清楚的问题，交给下一次课。

8 课时压缩建议

次序	合并课次	保留任务	删减建议
1	A 轨：第 1—2 课	运行 + 单步 + 变量	减少语法速览，保留一次改造
2	A 轨：第 3—4 课	输入 + 条件 + 真值表	删减嵌套分支拓展
3	B 轨：第 7 课	原生贪吃蛇源码阅读 + VM 运行	不要求读俄罗斯方块完整源码
4	第 9 + 11/12 课	AI 路由验收 + 导出 + 3 分钟展示	项目改为小组快速原型，互评组内完成