鸣众数研 / 科研教学

B26 / 基础

课堂讲义 · Python 零基础编程

1

Python入门：从第一段代码到可核对的金额表

读入一张公告表后，怎样可靠地转换金额、筛选记录，并把缺失和零值分开？

学习任务 完成20条公告的金额表，再独立处理7条含混合单位与缺失的模拟记录，提交代码及异常清单。

资料与样本 跟做和改做使用2026-11-04已留存的20条政府采购合同公告，仅限这次列表窗口；每条保留公告详情链接。

独立做为原创7条模拟数据。所有运行离线，不重新采集，也不代表全国统计。

把学习任务缩到一张可以核对的表

本课从一张已有的公告表开始，把金额从万元换成元，再按条件筛选。先不要记住所有Python语法。打开公告金额练习.csv，观察合同编号、合同名称、合同金额（万元）和公告详情链接四列。每一行是一条合同公告记录；不同合同可以有同样的名称，不能只按名称删重。脚本将保留合同编号和出处，让转换后的金额还能回到原记录。

这20条是2026年11月4日留存的一次政府采购合同公告列表样本。我们沿用已整理材料，不在运行时访问网站，也不声称本轮重新采集。首页记录会变动，样本规模和来源选择都限制了外推。12405922.20元是这20条记录的课堂合计，不能写成全国采购额或某行业的全年规模。

先给自己设一个可检查的终点：生成五张成果CSV，解释首条金额为什么是9992.47元，保留两份不同编号的印刷合同，再在新样本里区分缺失和零值。网页能核对你填入的数字，但文件是否保留来源、处理规则是否合理仍需要按人工标准检查。

第一次启动：分清编辑器、终端和运行入口

完整解压课堂包后，先在文件夹中确认课堂练习.py、公告金额练习.csv和运行模式.json都在同一级。不要只把.py拖到桌面，也不要在压缩包窗口里直接启动。首次配置按科研教学页面下载对应软件环境并运行安装入

安装与日常启动按随包《五步学习说明.txt》执行；非Python主课的Python复核为选做。

课计算本身离线。

编辑器用于写和保存代码；终端用于发出启动命令；带有>>>提示符的Python交互窗口接受一条条Python语句。这三处不能混着使用。例如python 课堂练习.py属于终端命令，把它输进>>>窗口会得到语法错误。课堂主入口已经替你选择固定解释器，起步阶段按它运行即可。查看源码可以用已有的纯文本代码编辑器，保存时保持.py扩展名。

本课执行的是整份脚本，先定义函数再使用它。只选中中间几行运行，可能缺少前面定义的目录或函数。若曾经在交互窗口定义过同名变量，旧状态还会掩盖缺行问题。检查是否真的可复跑，应关闭旧会话并重新运行完整入口。Windows固定Python已实跑；macOS入口保留使用说明，本课未做Mac实机验证。

变量、字符串与数字：先判断它代表什么

代码中的“金额文本”是一个变量名，里面保存从CSV读来的内容。CSV读取器默认给我们字符串，所以"0.999247"是一段文字。两个字符串用加号会拼接，如"10"+"20"得到"1020"；要做数值计算须先转换。合

同编号即使全是数字，也通常应当保留字符串，例如独立作业的001不应变成1，否则可能与另一张表匹配失败。

等号表示赋值，把右边的结果存到左边；双等号判断是否相等。先手写一个小实验：金额文本 = "0.999247"，再输出print(金额文本)，应看到原文字。字符串外面的引号不会作为内容打印。中文或英文变量名都可以使用，但同一个名字要前后一致；“总额”和“金额合计”是两个不同变量，不会自动互相替代。

本课特意保留原金额列和单位说明。数据类型不是研究含义：字符串可以代表金额，也可以代表编号。更不能看到字段里有0就决定它是未采购、未观测还是缺失。类型转换之前先读原表含义，类型转换之后再核对观察单位和数值范围。

从万元到元：把单位写进函数

第一条公告为0.999247万元，手算0.999247乘以10000得到9992.47元。函数转成元把同一规则集中在一个位置：输入原金额文字和单位，返回以元计的Decimal。定义def只登记规则，不会自动处理整张表；只有调用转成元("0.999247", "万元")时才执行函数内部语句。return将计算结果交回调用处，print只是显示信息。

金额使用Decimal从字符串直接构造。许多十进制小数不能在二进制浮点中精确表示；对账时不宜先转成float，再转Decimal补救。本课不将“显示两位小数”等同于“原数据精确到分”：先乘倍率，再将结果与量化到0.01的值比较。如果多出了分以下的金额，就要求核查，而不是悄悄四舍五入掩盖原始精度问题。函数只接受元和万元两种已明确的单位。独立样本中的美元会进入待核查表，因为货币兑换还需要明确汇率日期和来源；本课没有这些依据。空白金额、待确认文字、负值和非有限值也不能作为普通非负合同金额直接累计。确认含义为零的"0"则允许保留，这一差别会影响有效记录数。

相对路径与CSV：读到的是哪一个文件

Path(__file__).resolve().parent取得正在执行的脚本所在文件夹。目录 / "公告金额练习.csv"在这个位置寻找输入，因此从别的工作目录启动也能找到同级文件。它不会在整台电脑中自动搜索，也不能从只复制过来的脚本旁边变出数据。文件找不到时，先核对解压是否完整、文件名和扩展名是否一致，以及脚本旁是否有输入CSV。

with语句管理文件的打开和关闭；csv.DictReader把表头变成每行字典的键。例如行["合同金额（万元）"]读取这一列，列名里的中文括号也是名称的一部分。若把表头改为“金额”，旧代码不会猜测你的意思，而会报KeyError。打开文件时使用utf-8-sig兼容本课CSV的UTF-8标记，newline=""交给CSV模块处理换行。

list把读到的各行装进列表，len可以检查行数，列表的首个位置是0。本课应读到20行数据，表头不计入数据行。先打印第一条合同编号，再看第一条金额，不要只看程序有没有结束。读取错误的同名旧文件时，程序也可能顺利完成，输入名称、数量与首行核对能帮助发现这种情况。

用循环和条件逐笔计算，不按合同名称删重

for 行 in 公告依次取出每条记录。冒号后缩进的代码属于循环体，会重复执行；缩进结束后才进入汇总阶段。每轮先取合同编号，再换算本条金额。编号集合使用set记录已经出现过的编号；编号为空或重复时停止参考样本处理，要求回查。集合适合做存在性判断，金额表列表则保留需要导出的每一行。

总额从Decimal("0")开始累加。把总额初始化放进循环体，会导致每轮清零，最终往往只留下最后一笔金额。原脚本用sum对已统一单位的金额列表求和，作用同样是逐项累计。新手可以把前两条手算：9992.47元加375.60元等于10368.07元；再用20条总额检查循环是否覆盖全部记录。

if 行["合同名称"] == "印刷服务合同"只选择名称完全相同的记录，并不把它们合为同一笔交易。这里有两个不同合同编号，对应375.60元和2404.50元，合计2780.10元。按名称删掉其中一条会减少真实记录。是否重复需要更完整的业务键和来源核对，本课只检查这份小样本中合同编号的唯一性。

鸣众数研 / 科研教学

B26 / 基础

课堂讲义 · Python 零基础编程

3

改做练习：把阈值变成可以检查的参数

打开筛选参数.json，默认基准阈值元为100000、改做阈值元为10000。它们是以元计的数字字符串，代码读取后转为Decimal。条件>=表示“大于或等于”，因此恰好100000元的合同也进入10万元及以上组。不要把参数10000误写成1，再把标签仍写作1万元。参数和结果表说明必须使用同一种单位。

原样本中不低于10万元有11条，门槛降为1万元后有13条，筛后合计12375809.63元，新增的两笔为80000元和31700元。手算检查：新增金额为111700元，应该等于两种阈值下合计的差。数量增加而金额减少，通常提示累计对象、单位或筛选逻辑出了问题。

完成默认题后可把改做阈值改为50000，再运行并观察哪些记录离开集合。此时网页默认题目的13条答案不再适用，作业提交前应恢复10000。主程序在默认参数下执行精确答案断言，改参数时仍检查降低阈值不应减少记录的基本关系。研究中的筛选规则要有理由和版本记录，不能仅因结果好看就换阈值。

读报错与隔离异常：从最后一行找到具体对象

报错不等于数据应被删掉。先读最后一行的异常类型与说明，再往上看属于你脚本的文件名和行号。FileNotFoundError通常与路径或文件有关；KeyError通常与表头名称有关；NameError说明使用了尚未定义的名字。SyntaxError或IndentationError可能发生在真正计算之前，检查对应位置附近的冒号、引号、括号和缩进。

转成元遇到"待确认"会抛出ValueError。跟做的真实样本采用严格策略，发现异常应停下来核对；独立作业则演示如何逐条保留待核查记录，再继续处理其他可确认的行。try包住预期可能失败的转换，except只接收ValueError，else处理转换成功的记录。不要用不区分类型的except把所有程序错误吞掉，再一律写零。

独立练习_模拟金额.csv是原创的7条教学记录。001为1.20万元，002为850元，005为已确认0元，006为0.25万元；其余三条分别为空白、待确认和美元。正常结果应有4条有效记录、3条待核查，有效合计15350元。这个总额只是已确认记录的合计，不包含待核查项的未知金额。

写出文件后再读回，保留能解释的成果

写文件时先写中文表头，再用writerows写入各行。不要用字符串拼接逗号来造CSV；合同名称本身可能含逗号或换行，CSV模块会处理必要的引用。本课导出跟做_逐笔金额.csv、改做_阈值筛选.csv、独立做_有效

记录.csv、独立做_待核查.csv与课堂成果.csv。金额元列用两位小数字符串，编号和来源链接保持。

保存成功后再读跟做_逐笔金额.csv，检查20行和12405922.20元是否保持。屏幕中打印的总额正确，并不能证明写入文件用了同一批记录或同一个变量。若表格软件把001自动显示成1，先以文本编辑器或Python读回原CSV判断文件内容；单纯修改表格中的显示格式不一定能恢复已经丢失的原编号。

每次运行参考入口会重建参考成果，不会改写输入CSV。自己的作业请使用学生有效记录.csv和学生待核查.csv，避免覆盖参考文件。交作业时同时写明使用哪个输入、以什么单位计算、哪些记录被保留待核查。仅交一个正确的总额，无法说明程序的处理过程是否可信。

从参考程序转到自己的作业

先保持运行模式.json中的模式为参考，运行完整脚本得到基准结果。跟做结束后用纯文本编辑器把这个值改成学生，再打开作业起步.py。相同的可选Python独立复核入口会执行学生文件，方便沿用同一套固定环境。起步文

件第一次运行会明确提示请填写函数，这个NotImplementedError是留给你的学习位置；恢复模式为参考即可回到完整示范。

起步文件已负责读入新CSV和遍历记录，留下金额函数、输出和汇总几个任务。按TODO顺序完成：先支持元/万元，再判断空白与无效数字，保留确认零，最后输出有效与待核查两张表。不能直接把15350写成总

额，也不能删掉三条异常后声称没有问题。完成后用网页数值检查，再对照人工标准查看编号、单位和异常原因。

实在卡住时先看分层作业.txt中的提示，再看参考答案.txt。作业参考代码.py是一种完整实现，可以读来比较；如果要通过同一学生入口实际运行它，请先另存自己的起步文件，再将参考代码内容复制到作业起步.py并保存。数值一致只是第一关，还要解释为什么本样本允许零值、未知单位为何等待核查，以及换到新业务时哪些规则必须重审。

分层作业：跟做：完成20条公告金额换算

使用公告金额练习.csv，在参考模式运行课堂练习.py。手算首两笔，再检查输出记录与两份同名印刷合同。

提交内容：跟做_逐笔金额.csv，加一段说明观察单位、金额单位和同名记录为何保留。

提示：先确认表头原单位为万元；首条应换算成9992.47元。

提示：合同编号不同的两份印刷合同都应保留，不能按名称删重。

自查：输出20条且保留合同编号、合同名称和原公告链接。

自查：金额列明确为元，说明真实样本只是一次列表留存窗口。

自查：能用前两条10368.07元的手算检查程序。

参考解释：真实样本20条，合计12405922.20元。两份印刷合同375.60元与2404.50元合计2780.10元。

数值核对：跟做记录数 = 20 条；应保留全部20条；表头不计入数据行，同名不同编号不删重。

数值核对：跟做合计 = 12405922 元；核对每条先由万元乘10000，再累计，不要只乘部分记录。

数值核对：同名印刷合同合计 = 2780.1 元；375.60加2404.50；先核对两个不同合同编号。

分层作业：改做：将筛选门槛从10万元降到1万元

使用筛选参数.json，基准阈值100000元，改做阈值10000元，条件为大于或等于。交筛选结果并比较两种门槛。

提交内容：改做_阈值筛选.csv，加一张新增合同编号与金额的手工核对记录。

提示：>=包含等于阈值。先将输入金额都变成元再比较。

提示：新增两笔为80000元和31700元，可以核对集合差与总额差。

自查：写明参数以元计、使用包含等号的筛选条件。

自查：能列出新增的两份合同编号，验证差额111700元。

自查：修改其他阈值后恢复10000再填写本题，不能把不同口径的结果混在一起。

参考解释：10万元门槛11条，1万元门槛13条；新门槛合计12375809.63元，新增2条合计111700元。

数值核对：1万元门槛记录数 = 13 条；阈值应为10000元，记录数由11增至13。

数值核对：1万元门槛合计 = 12375810 元；只累计进入筛选结果的合同，检查是否误用全样本总额。

数值核对：新增记录数 = 2 条；两个筛选集合按合同编号作差，应得到80000元和31700元两笔。

分层作业：独立做：处理一份混合单位与缺失的模拟表

读取独立练习_模拟金额.csv，切换学生模式完成作业起步.py。只允许元和万元，空白、非数字与美元保留待核查；确认的零应留在有效记录中。

提交内容：学生有效记录.csv、学生待核查.csv和修改后的作业起步.py，附三种待核查原因。

提示：先把换算写成函数；单位检查、空白检查、Decimal转换分别处理。

鸣众数研 / 科研教学

B26 / 基础

课堂讲义 · Python 零基础编程

5

提示：001=12000元，002=850元，005=0元，006=2500元；这些是核对线索，不能直接硬编码成总额。

自查：从输入逐行计算，编号001等仍保留前导零。

自查：记录003/004/007分别记录缺失、非数字与未知单位原因，不把它们改成零。

自查：输出中保留005的合法零值，说明15350只是有效记录合计。

参考解释：原创模拟7条；有效4条、待核查3条、有效合计15350元。网页数值检查不能自动判断代码是否硬编码，也不能替代异常处理说明。

数值核对：有效记录数 = 4 条；005的0元是合法有效记录，不能因为数值为零就删除。

数值核对：待核查记录数 = 3 条；003空白、004待确认、007美元均需保留原因，不应混入有效合计。

数值核对：有效金额合计 = 15350 元；12000+850+0+2500=15350，未知金额不应被视为已经确认的零。

常见问题与处理

双击后提示找不到公告金额练习.csv。原因：只复制了脚本，或仍在压缩包内运行，或输入文件改了名。操作：完整解压到新文件夹，确认课堂练习.py和输入CSV同级；核对扩展名，不要把.csv改成.xlsx。复查：入口应读到20条，首条合同编号以KC-20261103开头。

出现SyntaxError，错误行写着python 课堂练习.py。原因：把终端命令输入到了Python的>>>交互窗口。

安装与日常启动按随包《五步学习说明.txt》执行；非Python主课的Python复核为选做。

令。复查：看到本课目录和分层任务的运行结果。

KeyError: 合同金额（万元）。原因：代码要求的表头与CSV第一行不同，或读取了另一份文件。操作：用文本编辑器查看第一行，核对中文括号与列名；恢复本课输入，不凭猜测替换字段含义。复查：转成元获得字符串数值，首笔为9992.47元。

NameError或IndentationError。原因：只执行中间几行、变量拼写不一致，或循环/函数的缩进被破坏。操作：先运行完整参考脚本；定位报错行的名字和定义。代码块统一四个空格，检查前行冒号，不混用制表符。复查：重新启动后从头执行，不依赖旧交互窗口的变量。

金额文字不能转换，或待核查记录突然变少。原因：存在空白、待确认或未知单位；也可能被错误地替换成0。操作：保留原始文字和编号，先核对原因。独立作业将预期的ValueError写入待核查表，其他异常继续报出。复查：有效4条、待核查3条；合法0元的005仍在有效表。

学生模式出现NotImplementedError。原因：作业起步.py中的金额函数仍是待完成位置。操作：按TODO完成函数与导出；查看渐进提示。需要回看完整演示时，把运行模式.json的学生改回参考。复查：学生文件完成后输出两张学生CSV；参考模式应仍能生成原五张成果表。

改阈值后网页数值检查不通过。原因：网页题目的口径固定为10000元，而本地参数可能已经改为别的值。操作：先记录新阈值实验的结果，再把筛选参数.json恢复默认100000和10000，重新运行。复查：默认改做结果13条、12375809.63元。

计算结果与口径

真实样本记录数 20

真实样本合计元 12405922.20

独立练习待核查数 3

任务

记录数

合计元

口径

跟做：公告样本

20

12405922.20

全部金额已统一为元

鸣众数研 / 科研教学

B26 / 基础

课堂讲义 · Python 零基础编程

6

任务

记录数

合计元

口径

跟做：同名印刷合同

2

2780.10

按不同合同编号保留

改做：不低于10000元

13

12375809.63

以元比较；包含等于阈值

独立做：有效记录

4

15350.00

自建模拟样本；包括已确认

的零

独立做：待核查

3

缺失、非数字和未知单位不

计入有效合计

前两项使用2026-11-04已留存的20条政府采购合同公告；独立练习为自建7条模拟记录。均为离线计算，不作全国统计推

断。

练习与参考答案

1. 使用公告金额练习.csv，在参考模式运行课堂练习.py。手算首两笔，再检查输出记录与两份同名印刷合同。

参考答案 真实样本20条，合计12405922.20元。两份印刷合同375.60元与2404.50元合计2780.10元。

2. 使用筛选参数.json，基准阈值100000元，改做阈值10000元，条件为大于或等于。交筛选结果并比较两种门槛。

参考答案 10万元门槛11条，1万元门槛13条；新门槛合计12375809.63元，新增2条合计111700元。

3. 读取独立练习_模拟金额.csv，切换学生模式完成作业起步.py。只允许元和万元，空白、非数字与美元保留待核查；确认的零应留在有效记录中。

参考答案 原创模拟7条；有效4条、待核查3条、有效合计15350元。网页数值检查不能自动判断代码是否硬编码，也不能

替代异常处理说明。

阅读材料与运行说明

Python 3.12 官方教程：数字、字符串与交互窗口

Python 3.12 官方教程：条件、循环与函数

Python 3.12 官方文档：Decimal 十进制计算

Python 3.12 官方文档：CSV 字典读取与写入

Python 3.12 官方教程：错误与异常

Python 3.12 官方文档：Path 文件路径

中国政府采购网合同公告检索（2026-11-04已留存20条，逐条链接见输入表）

完整注释代码、输入资料和成果文件见随课实操包。Windows固定Python 3.12.11已用随包输入实跑。macOS入口未实

机测试。本课以脚本为主，不需要Notebook编辑器。学生模式须先完成起步代码中的TODO，数值检查不能替代人工代

码审阅。

Python入门：从第一段代码到可核对的金额表

本课软件与五步学习说明

主软件：Python

软件准备入口：/learning-setup#python

1. 选择本课软件路线

本课主路线为Python。软件准备指南：/learning-setup#python。只准备本路线需要的工具。

2. 完成一次学习准备

先检查已有软件。Python主课或选做Python独立复核时，首次/修复使用prepare-python.cmd（Windows）或prepare-python.command（Mac）；日常打开使用open-python对应入口。其他软件按各自指南准备，不先装Python。

3. 第一次跑通并保存

首次准备后双击open-python.cmd（Windows）或open-python.command（Mac），在JupyterLab打开课堂学习.ipynb并选择本课内核。打开示例数据，运行一次分析或操作，看到结果，保存文件并找到它。仅安装成功还不能算跑通。

4. 打开本课项目

完整解压课程包。打开课堂学习.ipynb。课程输入和作业分别保存；兼容课程复用已准备环境，不在每节课重复安装。

5. 按步骤学习并改做

按本课分步实操运行，逐项核对预期结果。再复制课程目录，修改参数或换用独立作业数据；保留基准版本，不覆盖原始资料。

本项目额外要求与平台范围

Python固定依赖组：base

原脚本：课堂练习.py

分步笔记本：课堂学习.ipynb