2025.4.1

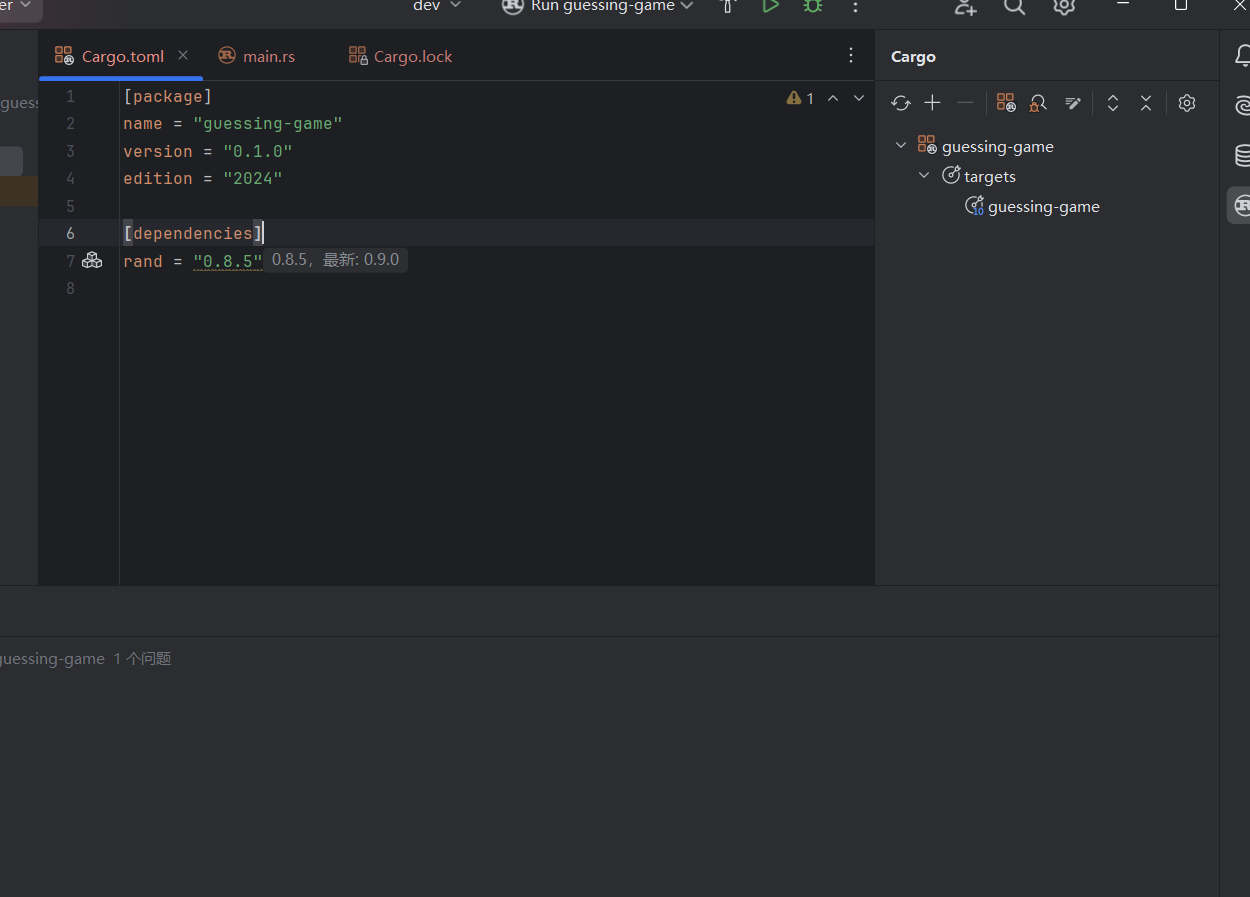
根据视频教程安装了rust，rustrover，并在cmd命令行进行了简单的文件操作，但有一个ls的指令我无法像视频中那样正常运行，看评论中博主说是在安装git时勾选某个特定选项就可以了，但我这时没有安装git故不能这样执行。

2005.4.2

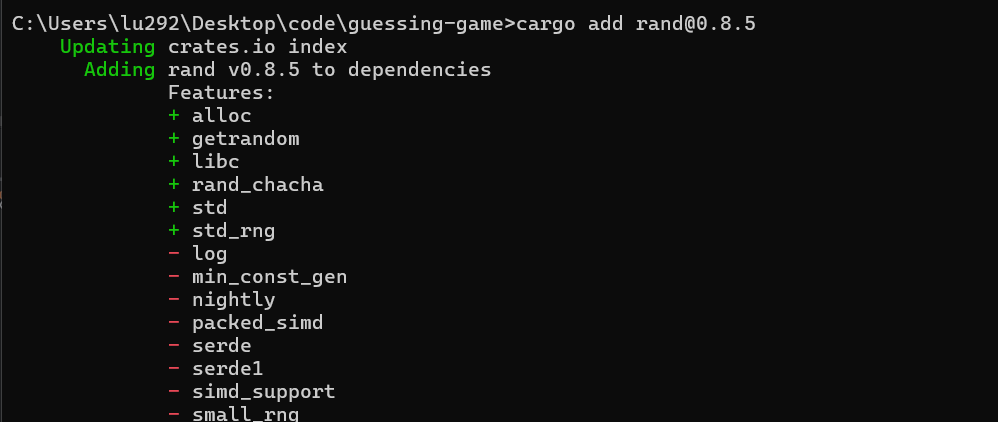
今天完善了一下rustrover相关配置，添加了其bin目录进入环境变量，可以在命令行执行XX\*的命令跳转到文件。看视频了解了一下rust程序的基本结构，使用了基本命令cargo build，cargo run。

添加库的方法

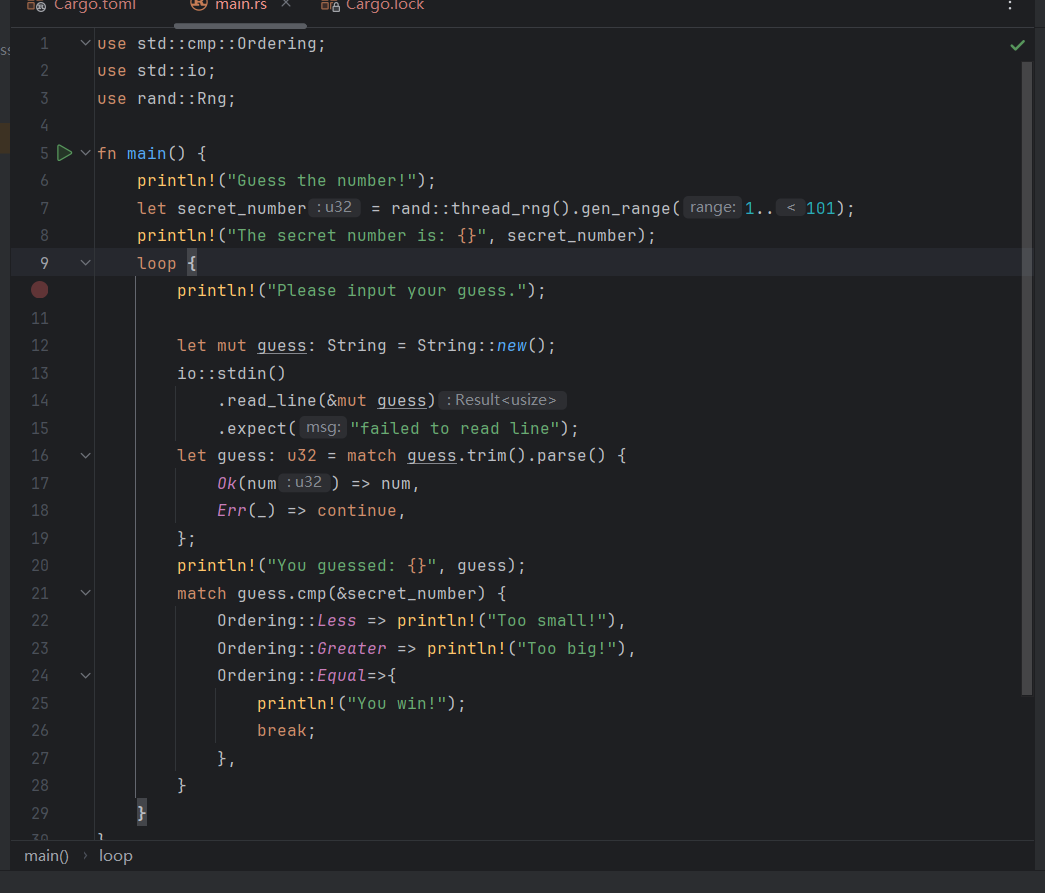
1. 在rustcover中添加



1. 在命令提示符中使用cargo add 库名@版本添加（若不@版本则自动添加最新版本）



并且我学习了rust的简单教程，猜数字游戏，通过该案例，我对rus的语法有了简单的了解，其包含声明变量，输入，输出，重覆盖变量类型，以及多种库与方法，了解了rust的一个loop循环结构，其结束语句依然与其他语言相似为break；同时rust还有方法可以对用户输入数据进行处理，如果输入类型错误还可以丢弃并再次进入寻欢。在随机数比较中，还使用了枚举类型，这是之前我很少接触的类型。总之，通过该教程，我对rust语言有了基本了解，剩下的就是区具体学习rust语言。

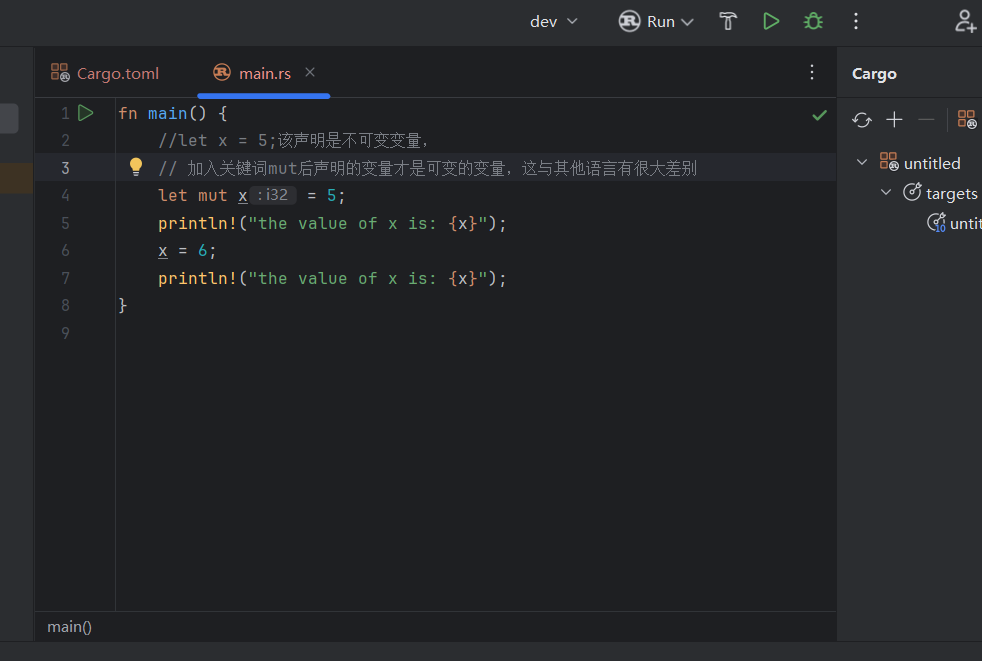


2025.4.4

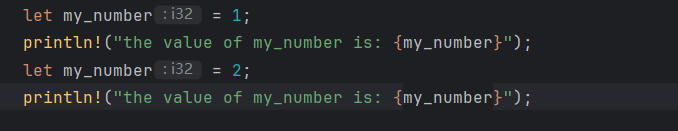
今天看了几个相关语言的介绍视频，主要是区别各个语言类型，语言有强弱之分，也有编译型语言和解释型语言之分，各个类型的语言有各个的特点，如编译型语言的运行速度更快，但是在不同的系统上就要编写不同的程序，而解释型语言只需要下载相同的解释器就可以运行，强语言在编写程序的时候就会对每个变量进行检测，每个语言在声明的时候就应该指定其类型，而弱语言却不需要在声明时指定，程序编译时系统会自动为其分配类型。

2025.4.5

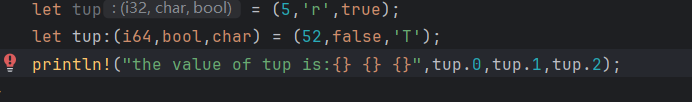
今天学习rust的基本数据类型和语法，首先最让我耳目一新的时，rust的声明语句默认是声明一个不可改变的变量，简直是一个常量，而常量也有其关键字使用const声明。



Rust还有一个变量遮蔽的特性，既重复声明变量是可行的，后声明的变量会将之前的声明覆盖。

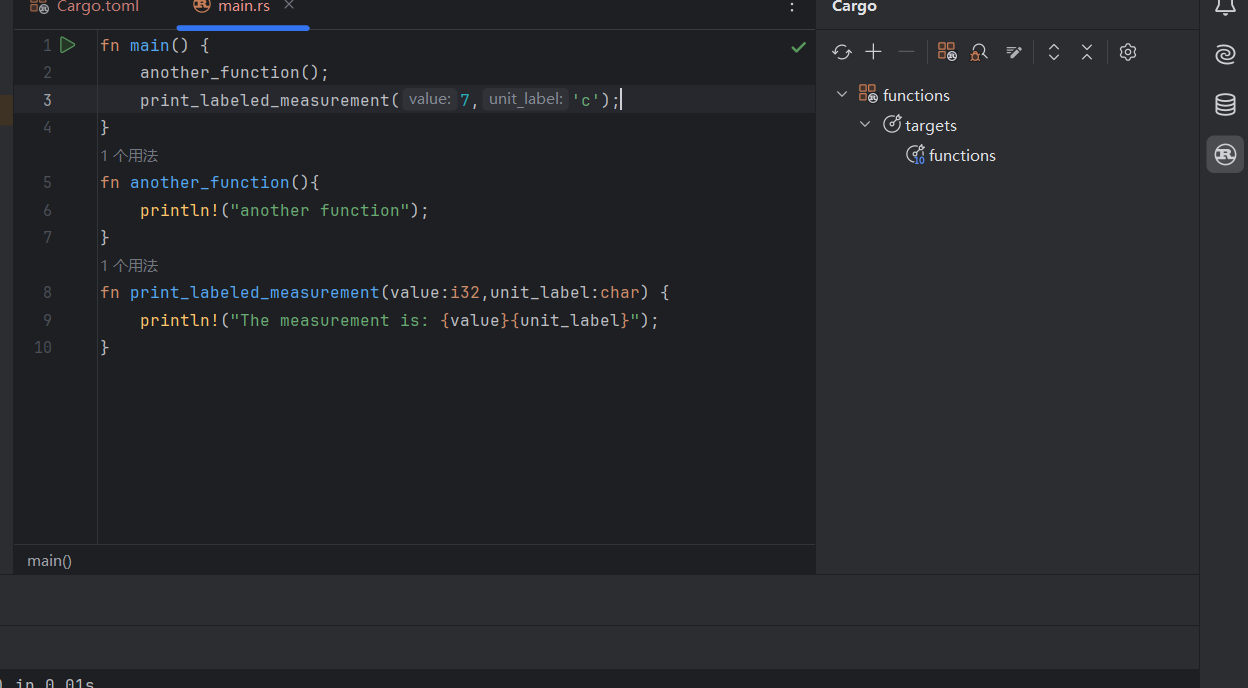


Rust可出现元组，像一个数组，但是可以出现不同的数据类型。让我感觉类似于结构体。

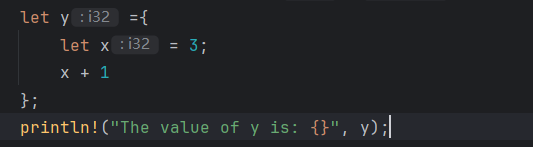


2025.4.6

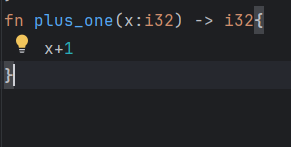
Rust中也有函数，其定义使用fn，命名规则为snake case 所有字母小写，单词之间用下划线隔开。



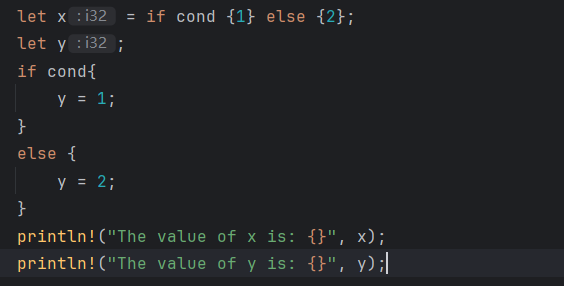
{}里面可以构成一个表达式来给变量赋值，语句不可以给变量赋值。



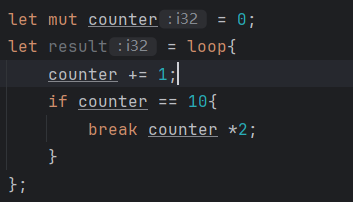
可以使用->指定返回值类型，函数体的最后一个表达式的值可以作为返回值。



其选择结构与其他语言的选择结构基本相同



Rust的循环结构跳出是可以返回一个表达式



嵌套循环可以用标签法直接结束外层循环

2025.4.7

Rust的安全性，不允许有未定义行为，这是其一个基本点。

Rust不允许手动管理内存，这与之前我接触的一些语言不一样。Rust是自动管理内存的。

如果一个变量拥有一个box，但rust释放该变量的frame时，rust会释放该box的堆内存（heep）

当前对于未定义行为理解还不是很好，在所有权视频中还出现了指针类型也不是很理解。

为了防止所有权转移后再使用时引发的未定义问题，我们可以让函数返回其所有权。

