猜数游戏 --- 一次猜测

知识点：let、match 等方法
相关函数
外部的 crate 第三方包、库

猜数游戏 - 目标

- 生成一个 1 到 100 间的随机数
- 提示玩家输入一个猜测
- 猜完之后，程序会提示猜测是太小了还是太大了
- 如果猜测正确，那么打印出一个庆祝信息，程序退出

let foo = 1;
let bar = foo;

在 rust 中默认所有**变量是不可变的 immutable** 如果想要生成可变的变量 就在前边加 **mut**

let mut guess = String::new();

使用 let 声明了一个变量 guess 将其绑定到一个空的字符串上边 字符串 String **:: 表明 new 是 String 的一个关联函数，具体函数内容如上** new 函数相当于创建了一个空白的字符串 创建类型实例的**惯用函数名称->new**

io 库有一个函数叫 stdin **返回一个 S 的实例** 用作句柄处理终端中的函数

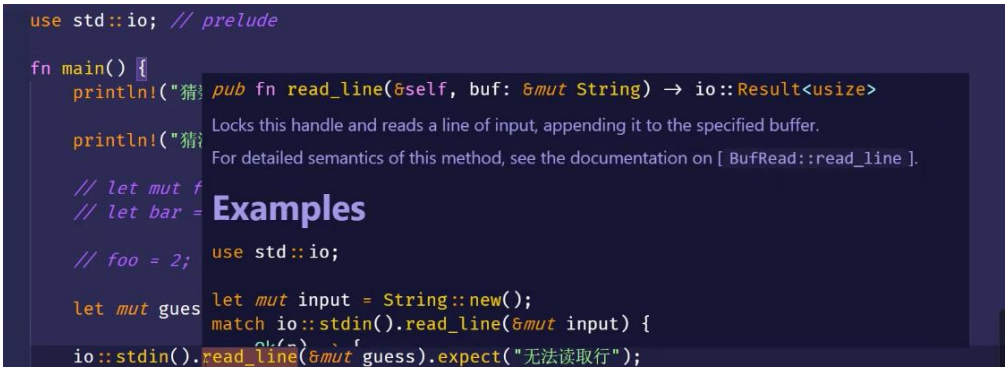
通过 readline 的方法 我们可以获得用户的输入 **readline 的方法就是把输入放进一个字符串中,所以需要一个字符串类型作为 read_line 的参数** 而 guess 是可变的 因为要输入

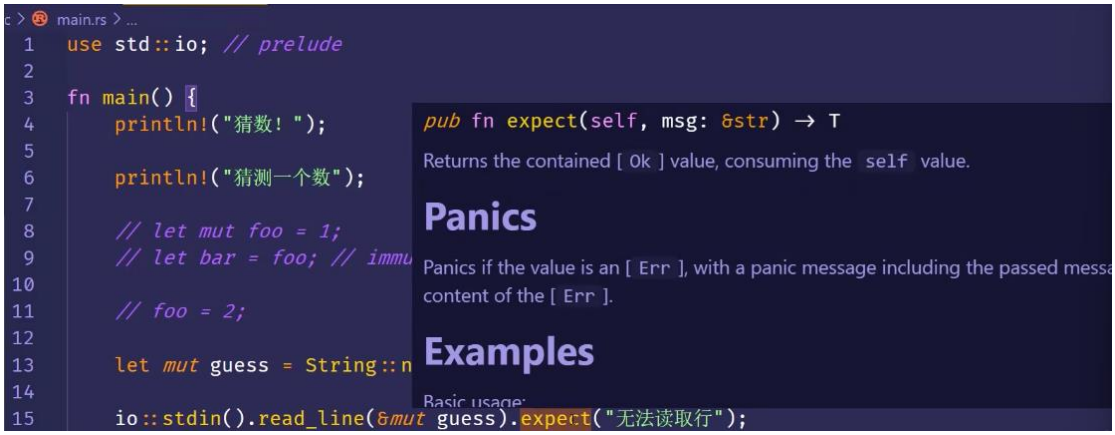
取地址符号表示一个引用 表明这个方法是按照引用传递的,引用在 rust 中是一个很复杂的特性,rust 的核心竞争力之一就是**安全的使用引用功能**

result 是一个枚举类型 有几个固定的值 io::result 就有两个值

io::result Ok,Err 两个变体 Ok 表示操作成功 返回值；Err 代表失败

expect 是定义的方法之一 **对潜在的错误进行处理** 会出波浪线





总体代码预览

main.rs > ...

```rust
// use std::io; // prelude

fn main() {
    println!("猜数! ");

    println!("猜测一个数");

    // let mut foo = 1;
    // let bar = foo; // immutable

    // foo = 2;

    let mut guess = String::new();

    std::io::stdin().read_line(&mut guess).expect("无法读取行");

    println!("你猜测的数是: {}", guess);
}
```

在电脑运行的结果

```
    Finished `dev` profile [unoptimized + debuginfo] target(s) in 0.60
s
     Running `target\debug\guessing_game.exe`
猜数!
猜测一个数
2
你猜测的数是:2
```