

Vectors

- Vectors只能存相同数据类型的数据
- 创建Vector的两种方法：
 - 创建空Vec:

```
let mut v: Vec<i32> = Vec::new();
```

- 利用 `vec!` 这个macro来创建带有初始值的Vec

```
let mut v = vec![1, 2, 3];
```

- 修改某个特定位置的元素:

```
let mut v = vec![100, 32, 57];  
v[2] = 99;
```

- 获取Vec中的值的两种方法:

- 用索引的方法:

```
let third: &i32 = &v[2];  
println!("The third element is {}", third);
```

- 用get:

```
match v.get(2){  
    Some(third) => println!("The third element is {}", third),  
    None => println!("There is no third element."),  
}
```

- 我们也可以看到，上面的两种方法的返回值是有差异的，用索引是返回的数值，而用get()返回的是Option T
- 这也导致在**越界访问**的时候，两种方法会有不同的表现:

```
let does_not_exist = &v[100];    // it will panic  
let does_not_exist = v.get(100);
```

- 第一行会panic，而第二行不会，因为它返回的是None
- 如果想要遍历Vec中的元素，我们可以borrow这个vec:

```
let v = vec![100, 32, 57];  
for i in &v{  
    println!("{}", i);  
}
```

- 如果想遍历并修改vec中的元素的话，我们可以用mutable的borrow

```
let mut v = vec![100, 32, 57];
for i in &mut v{
    *i += 50;    // To change the value that the mutable reference refers
                // to, we have to use the dereference operator *
}
```

- 但要注意的是！Vec也遵循之前所说的mutable和immutable的borrow规则。比如，下面的代码会报错：

```
let mut v = vec![1, 2, 3, 4, 5];
let first = &v[0];
v.push(6);
println!("The first element is {}", first);
```

报错信息是：

```
error[E0502]: cannot borrow `v` as mutable because it is also borrowed as
immutable
--> src/main.rs:30:5
   |
29 |     let first = &v[0];
   |                  - immutable borrow occurs here
30 |     v.push(6);
   |     ^^^^^^^^^ mutable borrow occurs here
31 |     println!("The first element is {}", first);    // it will trigger
compile error
   |                                                    ----- immutable borrow later
used here
```

- 原因是v.push (6) 发生的是一个mutable borrow，而&v[0]则是immutable borrow二者不能同时出现
- 但二者明明不在Vec的同一个位置，为什么会发生这样的错误呢？
 - 因为当push进新的值得时候，可能原先分配给Vec得位置不够用。这就需要给整个Vec找新的位置，这就会导致&v[0]地址也跟着发生变化。所有出于安全，我们也还是严格要求Vec整体也遵循immutable和mutable borrow不能同时出现得规则。
- 如果我们想让Vec装不同的数据类型，我们可以借助enum：

```
enum SpreadsheetCell{
    Int(i32),
    Float(f64),
    Text(String),
}

let mut row = vec![
    SpreadsheetCell::Int(3),
    SpreadsheetCell::Text(String::from("blue")),
    SpreadsheetCell::Float(10.12),
];

row.push(SpreadsheetCell::Int(4));
```

