



6

4

```
let mut numbers = vec![1, 2, 3];  
let mut iterator = numbers.iter();  
numbers = vec![6, 7, 8];  
  
while let Some(num) = iterator.next() {  
    println!("{}", num);  
}
```

**error[E0506]: cannot assign to `numbers` because it is borrowed**

**-->** <source>:29:5

```
|  
28 |     let mut iterator = numbers.iter();  
|                                     ----- `numbers` is borrowed here  
29 |     numbers = vec![6, 7, 8];  
|     ^^^^^^^ `numbers` is assigned to here but it was already borrowed  
30 |  
31 |     while let Some(num) = iterator.next() {  
|                                     ----- borrow later used here  
|  
= note: borrow occurs due to deref coercion to `[i32]`
```



Rust (safe)



# Rust (safe)

```
let mut numbers = vec![1, 2, 3];
let mut iterator = numbers.iter();
numbers = vec![6, 7, 8];

while let Some(num) = iterator.next() {
    println!("{}", num);
}
```

**error[E0506]: cannot assign to `numbers` because it is borrowed**

**-->** <source>:29:5

```
28 |         let mut iterator = numbers.iter();
    |                               ----- `numbers` is borrowed here
29 |         numbers = vec![6, 7, 8];
    |         ^^^^^^^ `numbers` is assigned to here but it was already borrowed
30 |
31 |         while let Some(num) = iterator.next() {
    |                               ----- borrow later used here
```

**= note:** borrow occurs due to deref coercion to `[i32]`

# A C++ Borrow-Checker?