



Rust 编程语言入门



Microsoft®
Most Valuable
Professional

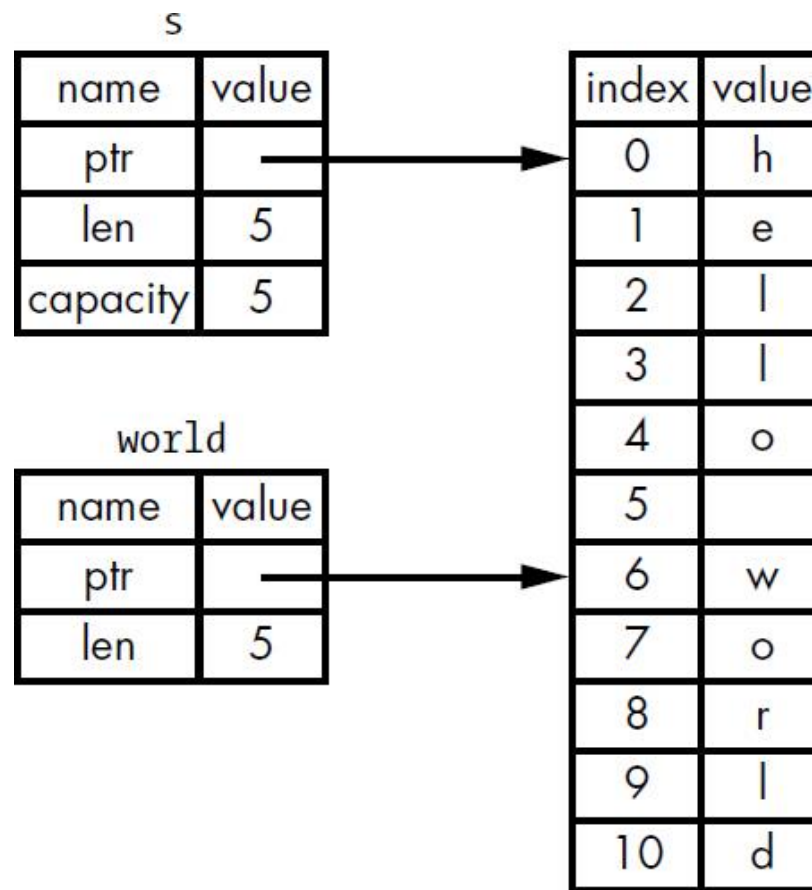
杨旭，微软MVP
Rust、Go、C#开发者

4.3 切片

- Rust 的另外一种不持有所有权的数据类型：切片（slice）
- 一道题，编写一个函数：
 - 它接收字符串作为参数
 - 返回它在这个字符串里找到的第一个单词
 - 如果函数没找到任何空格，那么整个字符串就被返回
- （尝试解答）

字符串切片

- 字符串切片是指向字符串中一部分内容的引用
- （例子）
- 形式：[开始索引..结束索引]
 - 开始索引就是切片起始位置的索引值
 - 结束索引是切片终止位置的下一个索引值



注意

- 字符串切片的范围索引必须发生在有效的 UTF-8 字符边界内。
- 如果尝试从一个多字节的字符中创建字符串切片，程序会报错并退出

使用字符串切片重写例子

- （例子）

字符串字面值是切片

- 字符串字面值被直接存储在二进制程序中。
- `let s = "Hello, World!";`
- 变量 `s` 的类型是 `&str`，它是一个指向二进制程序特定位置的切片
 - `&str` 是不可变引用，所以字符串字面值也是不可变的

将字符串切片作为参数传递

- `fn first_word(s: &String) -> &str {`
- 有经验的 Rust 开发者会采用 `&str` 作为参数类型，因为这样就可以同时接收 `String` 和 `&str` 类型的参数了：
- `fn first_word(s: &str) -> &str {`
 - 使用字符串切片，直接调用该函数
 - 使用 `String`，可以创建一个完整的 `String` 切片来调用该函数
- 定义函数时使用字符串切片来代替字符串引用会使我们的 API 更加通用，且不会损失任何功能。
- （例子）

其它类型的切片

- （例子）

再见

