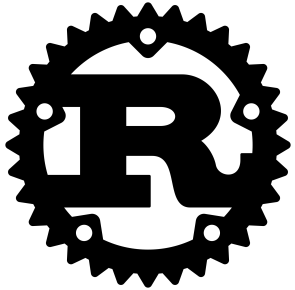

TinTin

第一课：初识Rust

Mike Tang

daogangtang@gmail.com

2023-5-1

Rust语言的Slogan

TinTin

A language empowering everyone to build reliable and efficient software.

一门赋予每个人构建可靠的高效软件能力的语言。

Rust语言的主要特点

- 高性能
- 内存安全
- 无忧并发(程序的开发)

高性能

TinTin

- 与C/C++一个级别的运行速度
- 方法抉择：
  - Zero Cost Abstract 零开销抽象
  - 无GC的自动内存管理 RAII
  - 可做到与C ABI一致的内存模型

内存安全

TinTin

- 使用Rust（非unsafe部分）写出来的代码，保证内存安全
- 方法抉择：
  - Ownership, move语义
  - Borrowchecker
  - 强类型系统
  - 无空值（Null, nil等）设计

无忧并发

TinTin

- 使用Rust进行多线程以及多任务并发代码开发，不会出现数据竞争和临界值破坏
- 方法抉择：
  - 对并发进行了抽象 Sync, Send
  - 融入类型系统
  - 基于Ownership, Borrowchecker实现，完美的融合性

打开几个辅助网站

https://play.rust-lang.org/?version=stable&mode=debug&edition=2021

https://doc.rust-lang.org/std/index.html

https://doc.rust-lang.org/book/title-page.html

https://doc.rust-lang.org/rust-by-example/index.html

创建一个工程

cargo new --bin helloworld

cargo build

cargo build —release

cargo run

cargo run —release

Hello, World

```
fn main() {
    println!("Hello, world!");
}
```

TinTin

赋值

```
fn main() {
    let a: u32 = 1;
}
```

数字类型

长度	有符号	无符号
8-bit	i8	u8
16-bit	i16	u16
32-bit	i32	u32
64-bit	i64	u64
128-bit	i128	u128
arch	isize	usize

整数字面量

Number literals	Example
Decimal	98_222
Hex	0xff
Octal	0o77
Binary	0b1111_0000
Byte (u8 only)	b'A'

浮点数

f32

f64

思考：

有没有类似于JavaScript中的那中统一的数字类型？

布尔类型

`bool`: `true` or `false`

字符 char

```
fn main() {
    let c = 'z';
    let z: char = 'ℤ';
    let heart_eyed_cat = '😻';
}
```

Unicode Scalar Value. Any Unicode *code point* except high-surrogate and low-surrogate code points. In other words, the ranges of integers 0 to $D7FF_{16}$ and $E000_{16}$ to $10FFFF_{16}$ inclusive. (See definition D76 in Section 3.9, Unicode Encoding Forms.)

https://unicode.org/glossary/#unicode_scalar_value

https://doc.rust-lang.org/std/primitive.char.html

字符串 String

String内部存储的Unicode字符串的UTF8编码。

```
let s = String::from("initial contents");
let hello = String::from("السلام عليكم");
let hello = String::from("Dobrý den");
let hello = String::from("Hello");
let hello = String::from("こんにちは");
let hello = String::from("안녕하세요");
let hello = String::from("你好");
```

字符串不能使用索引操作

注意:Rust中的String不能通过下标去访问:

```
let hello = String::from("你好");

let a = hello[0];  // 你可能想把"你"字取出来，但实际上这样是错误的
```

因为String存储的UTF8编码，这样访问即使能成功，也只能取出一个字符的UTF8编码的一部分，其是没有意义的。因此Rust直接对String禁止了这个索引操作。

```
error[E0277]: the type `String` cannot be indexed by `{integer}`
```

思考：如何将字符串转换为字符数组？

`String -> [char]` 或 `Vec<char>`

原始字符串字面量

```
fn main() {
    let raw_str = r"Escapes don't work here: \x3F \u{211D}";
    println!("{}", raw_str);

    let quotes = r#"And then I said: "There is no escape!""#;
    println!("{}", quotes);

    let longer_delimiter = r###"A string with "# in it. And even "##!"###;
    println!("{}", longer_delimiter);
}
```

字节串

```
fn main() {
    let bytestring: &[u8; 21] = b"this is a byte string";
    println!("A byte string: {:?}", bytestring);
    let escaped = b"\x52\x75\x73\x74 as bytes";
    println!("Some escaped bytes: {:?}", escaped);
    let raw_bytestring = br"\u{211D} is not escaped here";
    println!("{:?}", raw_bytestring);
    let _quotes = br#"You can also use "fancier" formatting, \
                    like with normal raw strings"#;
    let shift_jis = b"\x82\xe6\x82\xa8\x82\xb1\x82\xbb"; // ようこそ" in SHIFT-JIS
}
```

数组 - 定长数组

```
fn main() {

    let a: [i32; 5] = [1, 2, 3, 4, 5];

    let a = [1, 2, 3, 4, 5];

    let months = ["January", "February", "March", "April",
"May", "June", "July", "August", "September", "October",
"November", "December"];

}
```

下标索引数组元素

```
fn main() {
    let a: [i32; 5] = [1, 2, 3, 4, 5];
    let b = a[0];
    println!("{}", b)
}
```

如果下标索引越界了会发生什么？

数组 - 动态数组

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

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

let mut v = Vec::new();

v.push(5);

v.push(6);

v.push(7);

v.push(8);
```

能否使用下标索引动态数组？

下标索引动态数组

```
fn main() {
    let v = vec![1, 2, 3];
    let a = v[0];
}

fn main() {
    let v = vec![1, 2, 3];
    let a = v[3];  // ??
}
```

哈希表

```
use std::collections::HashMap;

fn main() {
    let mut scores = HashMap::new();
    scores.insert(String::from("Blue"), 10);
    scores.insert(String::from("Yellow"), 50);
    println!(“{:?}”, scores);
}
```

元组

```
fn main() {
    let tup: (i32, f64, u8) = (500, 6.4, 1);
}
```

```
fn main() {
    let x: (i32, f64, u8) = (500, 6.4, 1);
    let five_hundred = x.0;
    let six_point_four = x.1;
    let one = x.2;
}
```

结构体

```
struct User {
    active: bool,
    username: String,
    email: String,
    age: u64,
}
```

枚举

```
enum IpAddrKind {
    V4,
    V6,
}
let four = IpAddrKind::V4;
let six = IpAddrKind::V6;
```

分支语句 if else

```
let number = 6;
if number % 4 == 0 {
    println!("number is divisible by 4");
} else if number % 3 == 0 {
    println!("number is divisible by 3");
} else if number % 2 == 0 {
    println!("number is divisible by 2");
} else {
    println!("number is not divisible by 4, 3, or 2");
}
```

if else 其实可以返回值

```
fn main() {
    let x = 1;
    let y =  if x == 0 {
         100
    } else {
         101
    };
    println!("y is {}", y);
}
```

循环语句 loop, while, for

```
fn main() {
    let mut counter = 0;
    let result = loop {
        counter += 1;
        if counter == 10 {
            break counter * 2;
        }
    };
    println!("The result is {result}");
}
```

while

```
fn main() {
    let mut number = 3;
    while number != 0 {
        println!("{number}!");
        number -= 1;
    }
    println!("LIFTOFF!!!");
}
```

while true {} 是不是等于 loop {}？

for

```
fn main() {
    let a = [10, 20, 30, 40, 50];
    for element in a {
        println!("the value is: {element}");
    }
}
```

TinTin

range ..

```
fn main() {
    for number in 1..4 {
        println!("{number}");
    }
    for number in 1..=4 {
        println!("{number}");
    }
    for number in (1..4).rev() {
        println!("{number}");
    }
}
```

range的文档

https://doc.rust-lang.org/std/ops/struct.Range.html

range a..z

```
fn main() {
    for ch in 'a'..='z' {
        println!("{ch}");
    }
}
```

https://play.rust-lang.org/?version=stable&mode=debug&edition=2021&gist=7ec48585c881acee685a6c33dda94dec

函数 fn

```
fn print_a_b(value: i32, unit_label: char) {
    println!("The value of a b is: {value}{unit_label}");
    println!("The value of a b is: {}{}", value, unit_label);
}
fn main() {
     print_a_b(5, 'h');
}
```

模块

```
backyard
├── Cargo.lock
├── Cargo.toml
└── src
    ├── garden
    │   └── vegetables.rs
    ├── garden.rs
    └── main.rs
```

模块的另一种组织形式

```
backyard
├── Cargo.lock
├── Cargo.toml
└── src
    ├── garden
    │   └── mod.rs
    │   └── vegetables.rs
    └── main.rs
```

Cargo.toml

https://doc.rust-lang.org/cargo/

https://github.com/hyperium/hyper/blob/master/Cargo.toml

作业

创建一个Rust工程，

- 添加一个一层子模块，循环打印从’a’~’Z’ 之间的所有字符
- 添加一个二层子模块，循环打印从’A’~’z’ 之间的所有字符
- 使用Cargo编译运行此工程

Q&A

TinTin