



TinTin

Office Hour 1

讲师：Mike Tang
助教：雷昕鑫

本节课程介绍了数据类型、控制流和模块化

- 数据类型：基本数据类型（i8、u8、char等）、基本复合类型（定长数组、元组等）、自定义类型（结构体、枚举）
- 控制流：if 表达式和循环
- 模块化：嵌套模块和文件模块（2015写法和2018写法）

有没有类似于JavaScript中的那中统一的数字类型？

- 在 Rust 标准库中，没有像 JavaScript 中的统一数字类型（例如 JavaScript 中的 `number` 类型）那样的概念。Rust 选择使用不同的整数和浮点数类型来提供更精确和可控的数值表示和操作。这种设计选择有几个原因：
- 1. 性能和空间效率： 在低级语言中，如 Rust，直接使用具体的整数和浮点数类型可以更好地控制数值的存储和操作方式，以提高性能和空间效率。不同大小的整数和浮点数类型在内存中占据不同的空间，并且具有不同的操作和运算规则。
- 2. 类型安全和可靠性： Rust 是一门强类型的语言，它强调类型安全和可靠性。使用特定的整数和浮点数类型可以在编译时进行类型检查，避免一些常见的数值相关错误，如溢出和精度丢失。
- 3. 明确的数值语义： 不同的整数和浮点数类型在语义上有所不同。使用特定的类型可以明确表示数值的含义和范围。例如，使用有符号整数类型表示有符号的数值范围，使用无符号整数类型表示非负数值范围，使用浮点数类型表示带有小数部分的数值。
- 总的来说，Rust 的设计目标是提供安全、高效和可靠的编程体验，而不是追求一种统一的数字类型。通过使用具体的整数和浮点数类型，开发者可以更精确地控制数值的表示和操作，并且能够在编译时捕捉到一些潜在的错误。
- 第三方库有一些类似的实现，例如：[num-traits](#)，它提供了一组用于泛型数学操作的 Trait。它旨在为数值类型（如整数和浮点数）提供一个通用的接口，无论具体的实现是什么。

思考:如何将字符串转换为字符数组?

```
fn main() {  
    let s = "Hello, world!";  
    let chars: Vec<char> = s.chars().collect();  
    println!("vec: {chars:?}");  
  
    let mut array = ['\0'; 13];  
    for (idx, item) in s.chars().enumerate() {  
        array[idx] = item;  
    }  
    println!("array: {array:?}");  
}
```

- 在上面的代码中, `s.chars()` 返回一个迭代器, 然后使用 `collect` 方法将迭代器中的字符收集到一个 `Vec<char>` 中。也可以定义一个长度为 13 的字符数组 `array`, 初始值为 `'\0'` (空字符)。通过 `for` 循环遍历字符串 `s` 的每个字符, 并使用 `enumerate()` 方法获取字符在字符串中的索引。在循环体内, 将字符赋值给数组 `array` 对应索引的位置。
- <https://play.rust-lang.org/?version=stable&mode=debug&edition=2021&gist=76509261b353de2b7c0814226b9d19e5>

思考:while true {} 是不是等于 loop {}?

- 在 Rust 中，loop 可以在循环体内使用 break 语句并携带一个表达式作为返回值。当 break 语句执行时，它将中断当前循环并将指定的值作为整个 loop 表达式的结果返回。
- while true 里不能通过 break 返回值。
- loop 比 while true 更高效，因为是无条件判断的

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

考点:

1. 模块化
2. Range 和 RangeInclusive 的区别
3. ASCII 编码
4. Cargo 编译

模块化

1. 嵌套模块

2. 文件模块

● 2015 写法

- 模块组织需要依赖目录中的 `mod.rs` 文件。
- 每个目录可以包含一个 `mod.rs` 文件，该文件用于声明和定义当前目录下的模块。
- 在 `mod.rs` 文件中，可以使用 `mod` 关键字声明子模块，并通过 `pub` 关键字来指定子模块的可见性。

● 2018 写法

- 在2018版中，不再需要使用特定的`mod.rs`文件来管理模块，而是通过目录结构和同名的`.rs`文件来进行模块定义。

Range 结构体

```
pub struct Range<Idx> {  
    pub start: Idx,  
    pub end: Idx,  
}
```

The range start..end contains all values with start $\leq x < \text{end}$. It is empty if start $\geq \text{end}$.

RangInclusive 结构体

The RangInclusive start..=end contains all values with $x \geq \text{start}$ and $x \leq \text{end}$.

ASCII 编码

65	A		80	P		95	_		110	n	
66	B		81	Q		96	`		111	o	
67	C		82	R		97	a		112	p	
68	D		83	S		98	b		113	q	
69	E		84	T		99	c		114	r	
70	F		85	U		100	d		115	s	
71	G		86	V		101	e		116	t	
72	H		87	W		102	f		117	u	
73	I		88	X		103	g		118	v	
74	J		89	Y		104	h		119	w	
75	K		90	Z		105	i		120	x	
76	L		91	[106	j		121	y	
77	M		92	\		107	k		122	z	
78	N		93]		108	l		123	{	
79	O		94	^		109	m		124		

<https://play.rust-lang.org/?version=stable&mode=debug&edition=2021&gist=ada6c1d848607fa75ac55a4b742d2843>

I 优秀作业展示

ququzone

https://github.com/ququzone/rust-course/tree/main/course_01

Compile

```
cargo build --release
```

Run

```
./target/release/course_01  
cargo run --release
```

▼ course_01

▼ src

▼ l2_loop

mod.rs

print.rs

l1_loop.rs

main.rs

Cargo.lock

Cargo.toml

📖 README.md

1. [cargo clippy](https://github.com/rust-lang/rust-clippy): A collection of lints to catch common mistakes and improve your Rust code.
(<https://github.com/rust-lang/rust-clippy>)

2. rust-analyzer: Bringing a great IDE experience to the Rust programming language.
(<https://rust-analyzer.github.io/>)

答疑讨论



TinTin

THANKS

[Twitter](#)

[YouTube](#)

[Discord](#)