# 常量

Rust 常量的命名规范是使用下划线分隔的大写字母单词，并且可以在数字字面值中插入下划线来提升可读性。

例如：

|  |
| --- |
| const MAX\_POINTS: u32 = 100\_000; |

# 变量

Rust 变量的命名规范是 snake case 规范风格。所有字母都是小写并使用下划线分隔单词。

例如：

|  |
| --- |
| let x\_y = 5; |

# 全局变量

Rust 全局变量的命名规范是使用下划线分隔的大写字母单词，并且可以在数字字面值中插入下划线来提升可读性。

例如：

|  |
| --- |
| static HELLO\_WORLD: &str = "Hello, world!"; |

# 函数

Rust 函数的命名规范是 snake case 规范风格。所有字母都是小写并使用下划线分隔单词。

例如：

|  |
| --- |
| fn another\_function() |

# 结构体

Rust 结构体的命名规范是camel case(大驼峰)规范风格。其字段的命名规范是 snake case 规范风格

例如：

|  |
| --- |
| struct User {  username: String,  email: String,  sign\_in\_count: u64,  active: bool,  } |

# 枚举

Rust 枚举的命名规范是camel case(大驼峰)规范风格。其成员的命名规范是camel case(大驼峰)规范风格

例如：

|  |
| --- |
| enum Option<T> {  Some(T),  None,  } |

# Trait

Rust Trait的命名规范是camel case(大驼峰)规范风格。

例子：

|  |
| --- |
| pub trait Summary {  fn summarize(&self) -> String;  } |

# 模块

Rust 模块的命名规范是 snake case 规范风格。所有字母都是小写并使用下划线分隔单词。

例如：

|  |
| --- |
| mod front\_of\_house {  mod hosting {  fn add\_to\_waitlist() {}  fn seat\_at\_table() {}  }  mod serving {  fn take\_order() {}  fn server\_order() {}  fn take\_payment() {}  }  } |

# 文件

Rust 文件的命名规范是 snake case 规范风格。所有字母都是小写并使用下划线分隔单词。

# 目录

Rust 目录的命名规范是 snake case 规范风格。所有字母都是小写并使用下划线分隔单词。