Function

- 下面是一个Rust的function的案例：

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

    another_function();
}

fn another_function(){
    println!("Another function.");
}
```

  - Rust 的function以`fn`开头，括号中传入的是参数

传参的Example

- 跟C++一样，要规定参数的数据类型，用`数据类型：参数名`的方式。

```
print_labeled_measurement(5, 'h');
fn print_labeled_measurement(value: i32, unit_label: char){
    println!("The measurement is {} {}", value, unit_label);
}
```

Statement & Expressions

- Statement：是一个不返回值的instruction，比如`let`语句，就是不返回值的，下面的代码会报错：

```
let x = (let y = 6);
```

- Expression：相反，是会返回值的，并且应用场景非常广：
  - 一个算式： e.g. 5 + 6
  - 调用一个函数
  - 调用一个macro
  - 一个scope block with curly brackets：

```
let y = {
    let x = 3;
    x + 1
};
println!("The value of y is {}", y);
```

function返回值

- rust 用`-> 数据类型`来指示函数返回的数据类型。

```
let x = five();
println!("the value of x is {}", x);

fn five() -> i32{
    5
}
```

    - 但注意，不能写成`5;`因为这样就变成一个statement了。像上面所说，statement是不返回值的。