

# Chapter 1: Getting Ready

rustup is an installer for  
the systems programming language [Rust](#)

Run the following in your terminal, then follow  
the onscreen instructions.

```
curl https://sh.rustup.rs -sSf | sh
```

You appear to be running Unix. If not, [display all supported installers](#).

Need help? [Ask on #rust-beginners](#).



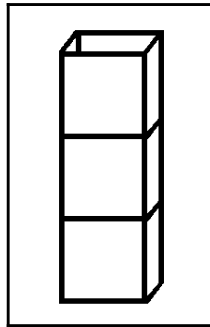
rustup is an official Rust project.  
[other installation options](#) · [about rustup](#)

```

$ cargo build
  Compiling chapter02 v0.1.0 (file:///home/djarb/writing/RustQSG/Rust-Quick-Start-Guide/chapter02)
warning: unused variable: `cons_invalid`
--> src/structs_and_behavior.rs:65:13
65 |         let mut cons_invalid = new_constrained(100, 0, 0)?;
    |         ^^^^^^^^^^^^^^^^^ help: consider using `_cons_invalid` instead
    = note: #[warn(unused_variables)] on by default
warning: variable does not need to be mutable
--> src/structs_and_behavior.rs:65:9
65 |         let mut cons_invalid = new_constrained(100, 0, 0)?;
    |         ----^^^^^^^^^^^^^^^^
    |         |
    |         help: remove this `mut`
    = note: #[warn(unused_mut)] on by default
warning: unused arithmetic operation which must be used
--> src/expressions.rs:54:37
54 |         println!("Block result {:?}", { 2 + 2; 19 % 3; println!("In a block"); true});
    |                                         ^^^^^
    = note: #[warn(unused_must_use)] on by default
warning: unused arithmetic operation which must be used
--> src/expressions.rs:54:44
54 |         println!("Block result {:?}", { 2 + 2; 19 % 3; println!("In a block"); true});
    |                                         ^^^^^
    Finished dev [unoptimized + debuginfo] target(s) in 4.51s
$

```

# Chapter 3: The Big Ideas – Ownership and Borrowing



```
error[E0382]: use of moved value: `point.x`
--> src/main.rs:18:50
15 |         receive_ownership(point);
    |         ----- value moved here
...
18 |         println!("point is Point2D{{x: {}, y: {}}}", point.x, point.y);
    |                                     ^^^^^^^ value used here after move
= note: move occurs because `point` has type `Point2D`, which does not implement the `Copy` trait

error[E0382]: use of moved value: `point.y`
--> src/main.rs:18:59
15 |         receive_ownership(point);
    |         ----- value moved here
...
18 |         println!("point is Point2D{{x: {}, y: {}}}", point.x, point.y);
    |                                     ^^^^^^^ value used here after move
= note: move occurs because `point` has type `Point2D`, which does not implement the `Copy` trait
```

```
error[E0308]: mismatched types
--> src/main.rs:42:13
42 |         value = 6;
    |         ^
    |         |
    |         expected &mut u32, found integral variable
    |         help: consider mutably borrowing here: `&mut 6`
= note: expected type `&mut u32`
       found type `{integer}`
```

$$\sqrt{x^2 + y^2}$$



# Chapter 5: One Data Type Representing Multiple Kinds of Data

```
error[E0308]: mismatched types
--> src/traitobjs.rs:54:9
54 |         &Turn{ slight: true, right: false },
    |         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ expected struct `traitobjs::Forward`, found struct `traitobjs::Turn`
    = note: expected type `&traitobjs::Forward`
           found type `&traitobjs::Turn`

error: aborting due to previous error

For more information about this error, try `rustc --explain E0308`.
error: Could not compile `chapter05`.
```

```
Trait object-style driving directions:
Go forward 5 blocks
Turn slightly left
Go forward 1 blocks
Turn right
Go forward 2 blocks
You have reached your destination
Turn 180 degrees
Go forward 2 blocks
Turn left
Go forward 1 blocks
Turn slightly right
Go forward 5 blocks
```

```
error[E0597]: `wrong` does not live long enough
--> src/any.rs:25:37
25 |         &DoesNotHaveAnyTrait{ name: wrong.as_str(), count: 16},
    |         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ borrowed value does not live long enough
...
60 |     }
    |     - borrowed value only lives until here
    = note: borrowed value must be valid for the static lifetime...
```

```
error[E0282]: type annotations needed
  --> src/any.rs:32:17
32 |         if step.is() {
   |                    ^^ cannot infer type for `T`

error: aborting due to previous error

For more information about this error, try `rustc --explain E0282`.
error: Could not compile `chapter05`.
```

```
error[E0004]: non-exhaustive patterns: `&Reverse` not covered
  --> src/enums.rs:31:15
31 |         match step {
   |                ^^^^ pattern `&Reverse` not covered

error: aborting due to previous error

For more information about this error, try `rustc --explain E0004`.
error: Could not compile `chapter05`.
```

# Chapter 6: Heap Memory and Smart Pointers

```
error[E0072]: recursive type `boxes::TreeNode` has infinite size
--> src/boxes.rs:6:1

6 | pub struct TreeNode {
  | ^^^^^^^^^^^^^^^^^^^ recursive type has infinite size
7 |     pub value: i32,
8 |     pub left: TreeNode,
  |     ^^^^^^^^^ recursive without indirection
9 |     pub right: TreeNode,
  |     ^^^^^^^^^ recursive without indirection
= help: insert indirection (e.g., a `Box`, `Rc`, or `&`) at some point to make `boxes::TreeNode` representable
```

```
error[E0308]: mismatched types
--> src/vectors.rs:5:17

5 |     vector.push("nope");
  |                   ^^^^^ expected floating-point variable, found reference
= note: expected type `{float}`
       found type `&'static str`

error: aborting due to previous error

For more information about this error, try `rustc --explain E0308`.
error: Could not compile `chapter06`.
```

```
["Ada", "Mel", "Ada", "Ada", "Mel", "Ada", "Mel"]
Remove which: 3
["Ada", "Mel", "Ada", "Mel", "Ada", "Mel"]
Remove which: 5
["Ada", "Mel", "Ada", "Mel", "Ada"]
Remove which: 2
["Ada", "Mel", "Mel", "Ada"]
Remove which: 3
["Ada", "Mel", "Mel"]
Remove which: 2
["Ada", "Mel"]
Remove which: 1
["Ada"]
Remove which: 0
```

```
error[E0382]: use of moved value: `cell`
--> src/cell_and_refcell.rs:9:20

8 |     println!("{}", cell.into_inner());
  |                   --- value moved here
9 |     println!("{}", cell.replace("I still didn't do anything.".to_string()));
  |                   ^^^^ value used here after move
= note: move occurs because `cell` has type `std::cell::Cell<std::string::String>`, which does not implement the `Copy` trait
```



## Chapter 7: Generic Types

```
error[E0243]: wrong number of type arguments: expected 1, found 0
--> src/main.rs:2:12

2 | let x: Option = None;
  |               ^^^^^^ expected 1 type argument
```

```
error[E0599]: no method named `get_ref` found for type `Tree<NotOrdered, f32>` in the current scope
--> src/main.rs:145:62

88 | pub struct Tree<K, V> where K: PartialOrd + PartialEq {
    |----- method `get_ref` not found for this
...
145 |     println!("tree.get_ref(\"third key\") is {}", match tree.get_ref("third key") {
    |                                                                ^^^^^^
    = note: the method `get_ref` exists but the following trait bounds were not satisfied:
            `NotOrdered : std::cmp::PartialOrd`
            `NotOrdered : std::cmp::PartialEq`
```

```
error[E0277]: can't compare `NotOrdered` with `NotOrdered`
--> src/main.rs:138:19

138 |     let mut tree: Tree<NotOrdered, f32> = Tree::new();
    |                                     ^^^^^^^^^^^^^^^^^ no implementation for `NotOrdered < NotOrdered` and `NotOrdered > NotOrdered`
    = help: the trait `std::cmp::PartialOrd` is not implemented for `NotOrdered`
note: required by `Tree`
--> src/main.rs:88:1

88 | pub struct Tree<K, V> where K: PartialOrd + PartialEq {
    |-----
```

```
error[E0308]: mismatched types
--> src/main.rs:155:16

155 |     return "0h no";
    |     ^^^^^^ expected integral variable, found reference

= note: expected type `{integer}`
       found type `&'static str`
```

```
error[E0387]: cannot borrow data mutably in a captured outer variable in an `Fn` closure
--> src/main.rs:185:9
185 |         y.push('X');
    |         ^
help: consider changing this closure to take self by mutable reference
--> src/main.rs:184:18
184 |         higher_order(|x: u32| {
185 |             y.push('X');
186 |             println!("In the closure, y is now {}", y);
187 |             x
188 |         });
```