**Install cargo**

**cargo new <project-name> :** it will create a new project named hello

18:43:36:~/Desktop/personal-projects/rust-basics **cargo new hello**

Created binary (application) `hello` package

18:59:39:~/Desktop/personal-projects/rust-basics **tree hello**

hello

├── Cargo.toml

└── src

└── main.rs

Cargo.toml is the config file and main.rs is the rust file.

We can run the rust using cargo run. For the first time it will compile and run and if there is no change then in next time it will directly run. Cargo run will create a target folder. By default, cargo will generate all binary file in debug folder, we can add --release to remove this debug symbol. It will generate the binary files in release folder.

let num=10;

println!("{}",num);

let (x,y)=(10,11);

println!("{} {}",x,y);

By default, variables in rust are immutable.

So, unlike other languages we cannot do this.

let num=10;

num=32;

We must define a variable mutable explicitly.

let num=10;

num=32;

If we a variable is not used, then cargo will give us a warning like: **warning: value assigned to `num` is never read**

We can also define constant using const variable.

const WORKER\_COUNT:i32=5;

println!("{}",WORKER\_COUNT);

Variables in rust block scoped.