Getting started

Hello world

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

The main function is always the first function that is executed in every Rust program. The body of the function is wrapped in `{}`. `println!` call is actually not a function call, but rather is calling a Rust macro. The macro is denoted by the `!` at the end of the function. Statements are end with semi colons.

Compiling and running rust program

To compile a single `*.rs` file, you would run

```
rustc main.rs
```

The rust compiler will compile your code into a binary where then you can execute the binary by running:

```
./main
```

Hello, cargo!

Cargo is rust's build system and package manager. Think of it as your build tool, like an automated Makefile that you don't have to write yourself. It will find and compile together all of the relevant rust files that you have written. It also serves as the package manager think of `npm` that let you download and use third party library that is written by other people.

If you are creating more than one rust files, then it would probably be a good idea to use cargo as your build tool and package manager. It is the de facto standard anyway.

Creating a project with cargo

To create a new project with cargo, run the following code, of course substitute the project name with what you want

```
cargo new hello_cargo
```

The project directory that is created with cargo will be initialize as a git repository for you automatically.

It also contain `Cargo.toml` which is used for managing dependencies used by your project.

`src` directory contains all of your rust source code.

Building and running cargo project

To build your cargo project run:

```
cargo build
```

This will create your executable file in *target/debug/<executable_name>* but still in the same root directory. You can then run the executable the same way.

If you want to build and run your executable immediately you can issue the command:

```
cargo run
```

This will build then immediately run your executable after it has finished building.

If you want to check if your code can compile run

```
cargo check
```

To see if your code can be compiled or not.

Building for release

If you are going to release your code for production then you can run the command:

```
cargo build --release
```

This will compile your code with optimization, and the executable will be under `target/release` instead of `target/debug`.

Revision #1
Created 2023-01-13 20:42:23 UTC by Tamarine
Updated 2023-01-13 21:13:27 UTC by Tamarine