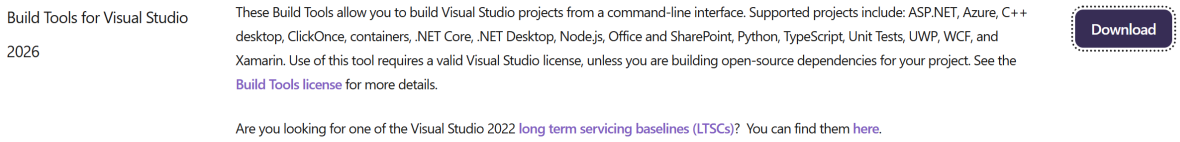
Discussion 1

See https://rust4ds.github.io/ds210-sp26-a1/discussion/00_setup.html for other codes and common issue solving.

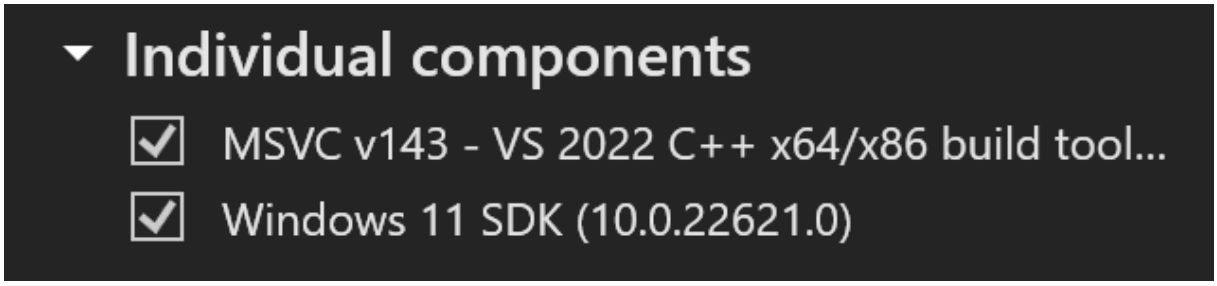
1 Install Prerequisite

1.1 Windows

1. If you have `Visual Studio` installed and have configured the C++ environments, then you possibly already installed it.
2. If not, download and install the MSVC Build Tools separately from https://visualstudio.microsoft.com/downloads/ (Scroll down a bit and click the *Tools for Visual Studio*)



3. **Alternatively**, install the **Windows (11) SDK** and **MSVC C++ Build Tools** from Visual Studio installer. (You don't need the full version of Visual Studio.)



2. The `rustup` installer will notify you if it can find the compiler in later step.

1.2 Linux

1. Install the development (meta-)packages.
   - For Ubuntu/Debian, it should be `build-essential`. Install it by

     ```
     sudo apt install build-essential
     ```
   - For Arch-based distribution,

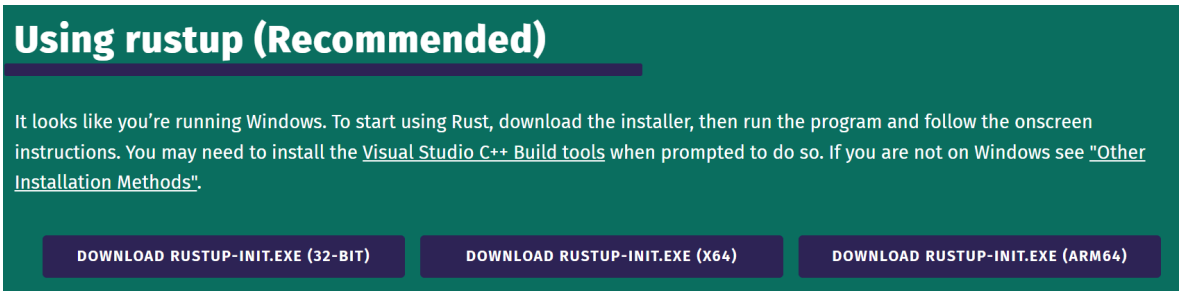
     ```
     sudo pacman -S base-devel
     ```

1.3 MacOS

2 Install Rust

2.1 Windows

1. Go to `https://rust-lang.org/tools/install/` and download the `rustup-init.exe` according to your machine (x32, x64 or arm64).



2. Run the `rustup-init.exe` for installation.

2.2 Linux

- For Ubuntu/Debian, install the packages by running (which is also on the website)

  ```
  curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
  ```

- For Arch-based, you can also run the same script or install from the official repo

  ```
  sudo pacman -S rust
  ```

2.3 MacOS

Use the same script as

```
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

3 Test your installation

1. Check your version by

   ```
   rustup check
   ```

   Currently the newest version is `1.92.0` for `rust` and `1.28.2` for `rustup`.

2. Open any folder and run the following command to initialize a sample project

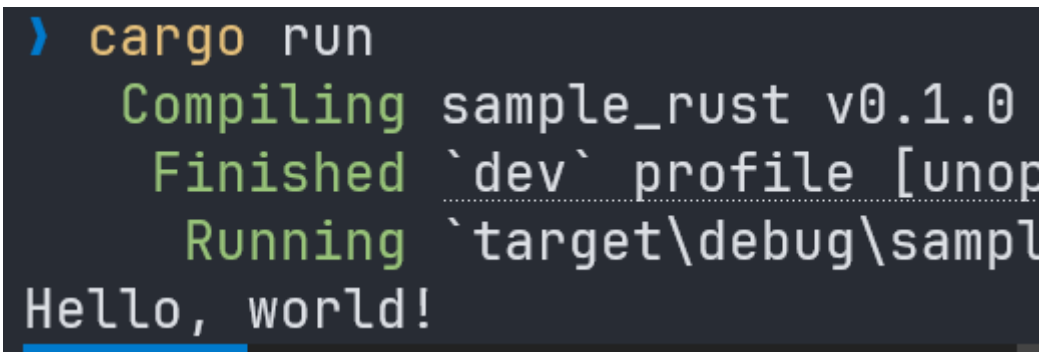
   ```
   cargo new sample_rust --bin
   # You can omit the --bin which is already the default behaviour now
   ```

3. Enter the project folder created

   ```
   cd sample_rust
   ```

4. Try run the project by `cargo`. Your should see an output of `Hello, world!`

   ```
   cargo run
   ```

5. Also try to build the project with `--release` flag and run it.

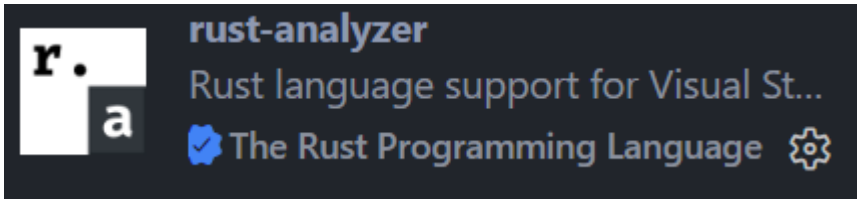
```
cargo build --release
cargo run --release
```

```
❯ cargo run --release
    Finished `release` profile [optimized] target(s) in 0.01s
     Running `target\release\sample_rust.exe`
Hello, world!
```

4 Install rust-analyzer for VS Code

In `Visual Studio Code`, install the `rust-analyzer` for linting, code completion, symbol searching, etc... You will find how powerful it is when writing the codes :)

Open the folder for project

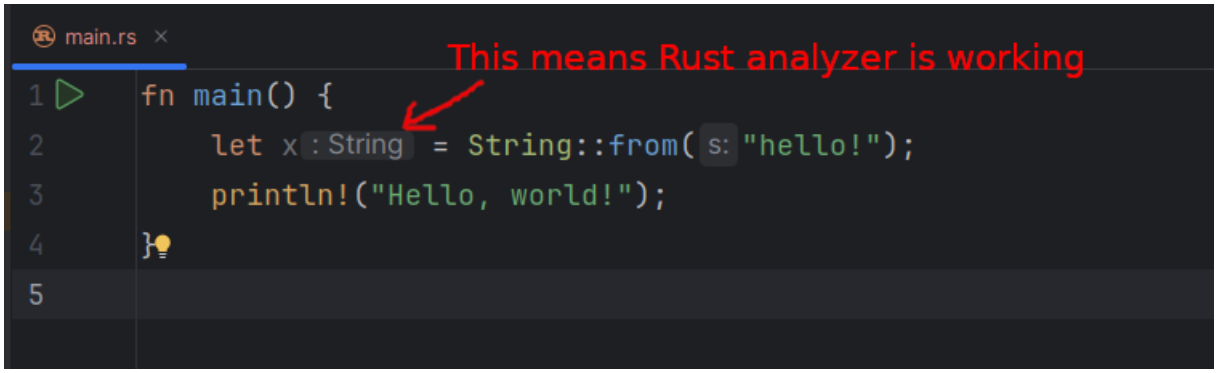


Test your plugin installation:

Write the following codes in your IDE:

```
fn main() {
    let x = String::from("hello!");  // Add this line
    println!("Hello, world!");
}
```

and check if you can see the following variable type deducing:



5 More...

Take a look at the official handbook! https://rust-lang.org/learn/