Cargo Workspace

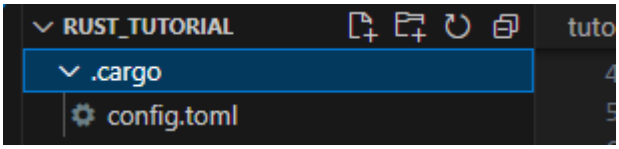
How to make a Workspace

- First make a directory
  - mkdir /../.././Rust_Tutorial
- Then make a `Cargo.toml` file
- Add the following contents:

```
[workspace]
members = [
    "tutorial1",
    "tutorial3_data_types",
    "tutorial4_console_input",
    "tutorial5_arithmetic",
    "tutorial6_conditionals",
    "tutorial7_functions",
    "tutorial8_Memory",
    "tutorial9_loops",
    "tutorial10_ownership",
    "tutorial11_Structs",
    "tutorial12_Enum_pattern_match",
    "tutorial13_Modules",
    "tutorial14_vector_string_hashmap",
    "tutorial15_Error",
    "tutorial16_GenericTypes",
    "tutorial17_Traits",
    "tutorial18_Lifetimes",
    "tutorial19_Closures",
    "tutorial20_Iterators",

]

workspace.resolver = "1"
```

  - Here we declare it as a workspace on top
  - Then the members are each new packages or crates from `cargo new`
  - Lastly we need to resolve any versioning error using the `workspace.resolver` property
- In case of SSL errors or network errors
  - In the workspace directory make a new folder called `.cargo`
  - Then add a file in that directory called `config.toml`
  - 

  - In that file add the following two lines

```
[http]
check-revoke = false
```

- Run `cargo build` on the workspace and it will build all projects together

Extending Cargo

- You can install external packages using `cargo install <package name>`
- This will be installed in the `~/.cargo/bin/` directory
- You can also extend cargo by naming any type of binary with a prefix of `cargo`
- Ex. if you have a binary named `cargo-something` you can run it as `cargo something`