Intro

1.0

Dowload and install tutorial page: https://docs.microsoft.com/en-za/dotnet/core/install/linux-package-manager-ubuntu-1804

2.0 Development

2.1 dotnet command line samples and looking

```
$dotnet --help
```

```
.NET Core SDK (3.0.100)
Usage: dotnet [runtime-options] [path-to-application] [arguments]

Execute a .NET Core application.

runtime-options:
  --additionalprobingpath <path>   Path containing probing policy and
assemblies to probe for.
  --additional-deps <path>         Path to additional deps.json file.
  --fx-version <version>           Version of the installed Shared
Framework to use to run the application.
  --roll-forward <setting>         Roll forward to framework version
(LatestPatch, Minor, LatestMinor, Major, LatestMajor, Disable).

path-to-application:
  The path to an application .dll file to execute.

Usage: dotnet [sdk-options] [command] [command-options] [arguments]

Execute a .NET Core SDK command.

sdk-options:
  -d|--diagnostics  Enable diagnostic output.
  -h|--help         Show command line help.
  --info            Display .NET Core information.
  --list-runtimes   Display the installed runtimes.
  --list-sdks       Display the installed SDKs.
  --version         Display .NET Core SDK version in use.

SDK commands:
  add               Add a package or reference to a .NET project.
  build             Build a .NET project.
  build-server      Interact with servers started by a build.
  clean             Clean build outputs of a .NET project.
```

```
  help              Show command line help.
  list              List project references of a .NET project.
  msbuild           Run Microsoft Build Engine (MSBuild) commands.
  new               Create a new .NET project or file.
  nuget             Provides additional NuGet commands.
  pack              Create a NuGet package.
  publish           Publish a .NET project for deployment.
  remove            Remove a package or reference from a .NET project.
  restore           Restore dependencies specified in a .NET project.
  run               Build and run a .NET project output.
  sln               Modify Visual Studio solution files.
  store             Store the specified assemblies in the runtime package
store.
  test              Run unit tests using the test runner specified in a
.NET project.
  tool              Install or manage tools that extend the .NET
experience.
  vstest            Run Microsoft Test Engine (VSTest) commands.

Additional commands from bundled tools:
  dev-certs         Create and manage development certificates.
  fsi               Start F# Interactive / execute F# scripts.
  sql-cache         SQL Server cache command-line tools.
  user-secrets      Manage development user secrets.
  watch             Start a file watcher that runs a command when files
change.

Run 'dotnet [command] --help' for more information on a command.
```

2.2 List new project templates

```
$dotnet new --list
```

```
Usage: new [options]

Options:
  -h, --help          Displays help for this command.
  -l, --list          Lists templates containing the specified name. If no
name is specified, lists all templates.
  -n, --name          The name for the output being created. If no name is
specified, the name of the current directory is used.
  -o, --output        Location to place the generated output.
  -i, --install       Installs a source or a template pack.
  -u, --uninstall     Uninstalls a source or a template pack.
  --nuget-source      Specifies a NuGet source to use during install.
  --type              Filters templates based on available types.
Predefined values are "project", "item" or "other".
  --dry-run           Displays a summary of what would happen if the given
command line were run if it would result in a template creation.
```

```
  --force             Forces content to be generated even if it would
change existing files.
  -lang, --language   Filters templates based on language and specifies the
language of the template to create.
  --update-check      Check the currently installed template packs for
updates.
  --update-apply      Check the currently installed template packs for
update, and install the updates.


Templates                                           Short Name
Language          Tags
--------------------------------------------------------------------------
-----------------------------------------------------
Console Application                                 console
[C#], F#, VB      Common/Console
Class library                                       classlib
[C#], F#, VB      Common/Library
WPF Application                                     wpf
[C#]              Common/WPF
WPF Class library                                   wpflib
[C#]              Common/WPF
WPF Custom Control Library                          wpfcustomcontrollib
[C#]              Common/WPF
WPF User Control Library                            wpfusercontrollib
[C#]              Common/WPF
Windows Forms (WinForms) Application                winforms
[C#]              Common/WinForms
Windows Forms (WinForms) Class library              winformslib
[C#]              Common/WinForms
Worker Service                                      worker
[C#]              Common/Worker/Web
Unit Test Project                                   mstest
[C#], F#, VB      Test/MSTest
NUnit 3 Test Project                                nunit
[C#], F#, VB      Test/NUnit
NUnit 3 Test Item                                   nunit-test
[C#], F#, VB      Test/NUnit
xUnit Test Project                                  xunit
[C#], F#, VB      Test/xUnit
Razor Component                                     razorcomponent
[C#]              Web/ASP.NET
Razor Page                                          page
[C#]              Web/ASP.NET
MVC ViewImports                                     viewimports
[C#]              Web/ASP.NET
MVC ViewStart                                       viewstart
[C#]              Web/ASP.NET
Blazor Server App                                   blazorserver
[C#]              Web/Blazor
ASP.NET Core Empty                                  web
[C#], F#          Web/Empty
ASP.NET Core Web App (Model-View-Controller)        mvc
[C#], F#          Web/MVC
```

```
ASP.NET Core Web App                                webapp
[C#]              Web/MVC/Razor Pages
ASP.NET Core with Angular                           angular
[C#]              Web/MVC/SPA
ASP.NET Core with React.js                          react
[C#]              Web/MVC/SPA
ASP.NET Core with React.js and Redux                reactredux
[C#]              Web/MVC/SPA
Razor Class Library                                 razorclasslib
[C#]              Web/Razor/Library/Razor Class Library
ASP.NET Core Web API                                webapi
[C#], F#          Web/WebAPI
ASP.NET Core gRPC Service                           grpc
[C#]              Web/gRPC
dotnet gitignore file                               gitignore
Config
global.json file                                    globaljson
Config
NuGet Config                                        nugetconfig
Config
Dotnet local tool manifest file                     tool-manifest
Config
Web Config                                          webconfig
Config
Solution File                                       sln
Solution
Protocol Buffer File                                proto
Web/gRPC
```

2.3 Create new solution

```
$dotnet new sln -o basic_projects
```

2.4 Create library project and add to solution

```
$dotnet new classlib -o common_utils
$dotnet sln add common_utils/common_utils.csproj
```

2.5 Create console project and add to solution

```
$dotnet new console -o text_analyzer
$dotnet sln add text_analyzer/text_analyzer.csproj
```

2.6 List current files

```
$tree
.
├── basic_projects.sln
├── common_utils
│   ├── Class1.cs
│   ├── common_utils.csproj
└── text_analyzer
    ├── Program.cs
    └── text_analyzer.csproj
```

2.7 Add nuget package to console app

```
$dotnet add text_analyzer/text_analyzer.csproj package CommandLineParser
```

2.8 Add custom library project to console app

```
$dotnet add text_analyzer/text_analyzer.csproj reference
common_utils/common_utils.csproj
```

2.9 Restore project

```
$dotnet restore
```

2.10 Build project

```
$dotnet build
```

2.11 Run poject

```
$dotnet run --project text_analyzer/text_analyzer.csproj -f test.txt
```

2.12 Create nuget package

Add the following line into library project .csproj file

```
<PropertyGroup>
  .
  .
```

```
  .
  <GeneratePackageOnBuild>true</GeneratePackageOnBuild>
<PropertyGroup>
```

2.13 Create different platform publish

```
$dotnet publish -c Release -r linux-x64 -o build-x64 -
p:PublishSingleFile=true -p:PublishTrimmed=true
text_analyzer/text_analyzer.csproj

$dotnet publish -c Release -r linux-arm -o build-arm -
p:PublishSingleFile=true -p:PublishTrimmed=true
text_analyzer/text_analyzer.csproj

$dotnet publish -c Release -r linux-arm64 -o build-arm64 -
p:PublishSingleFile=true -p:PublishTrimmed=true
text_analyzer/text_analyzer.csproj
```