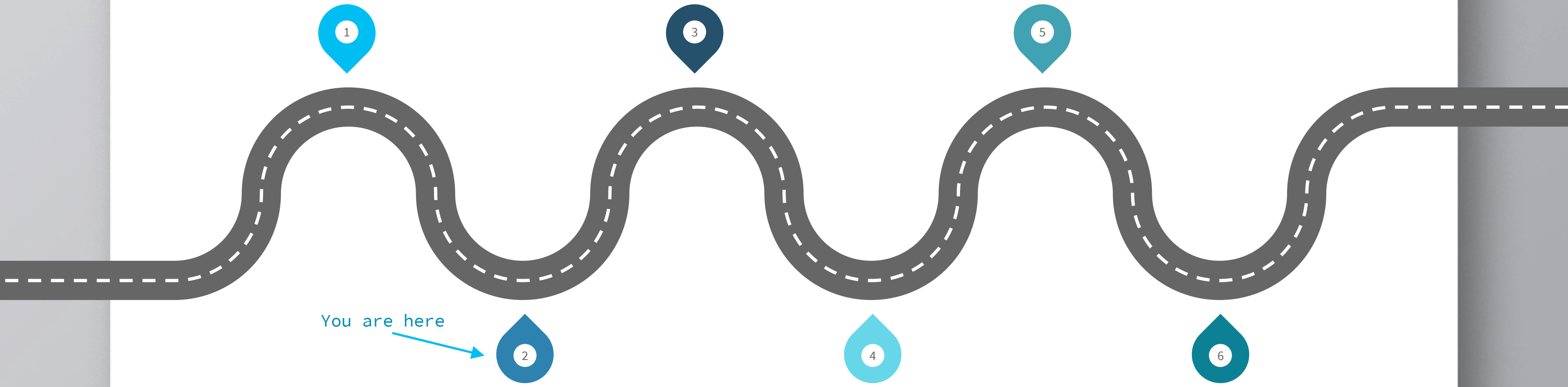
Introduction to Rust

Chapter 2

Rust

ROADMAP



Basic Tool Installation

Your first lines of Rust

Using Cargo and Crates

Data Types and User Input

Conditionals and Loops

Project

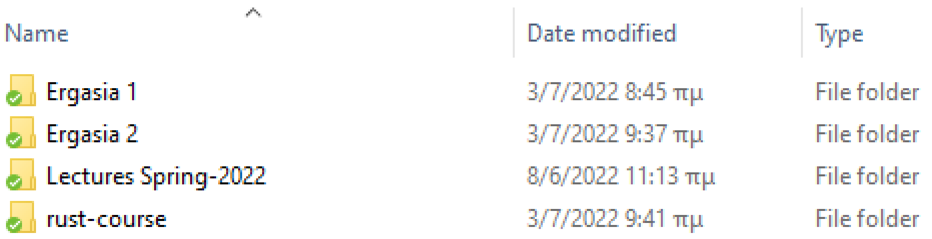
2.

Your first lines of Rust

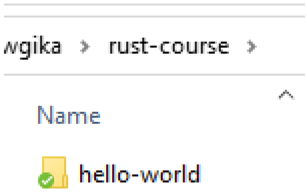
Creating project folder

To have everything organized lets first create a folder for all our course project files. I will name that folder “rust-course” but you can name it anything you like.

Inside that folder lets create another folder called “hello-world” for today’s lecture’s project.



Name	Date modified	Type
Ergasia 1	3/7/2022 8:45 πμ	File folder
Ergasia 2	3/7/2022 9:37 πμ	File folder
Lectures Spring-2022	8/6/2022 11:13 πμ	File folder
rust-course	3/7/2022 9:41 πμ	File folder



ᴡgika › rust-course ›

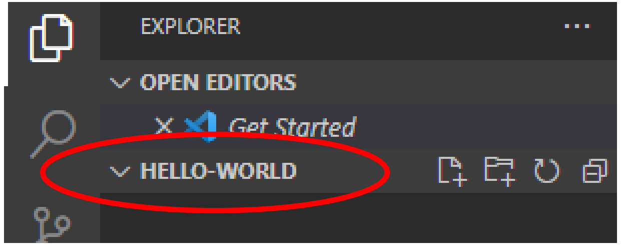
Name
hello-world

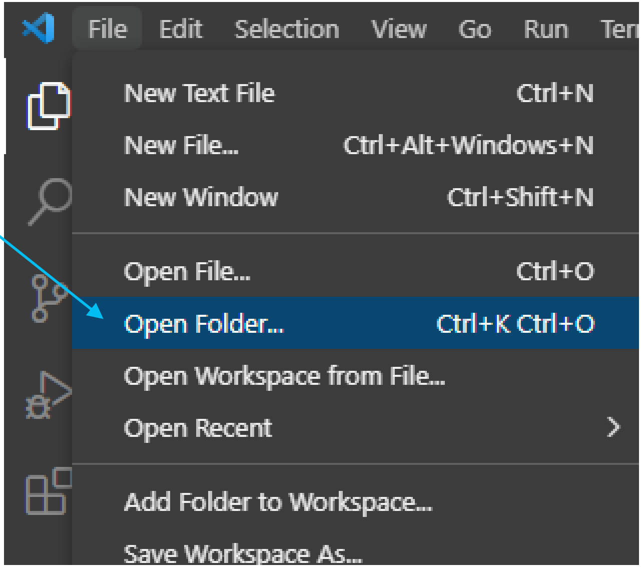
Creating main.rs

Now we can open this folder in Visual Studio Code.

Go to File -> Open Folder… and choose the folder we created previously called “hello-world”.

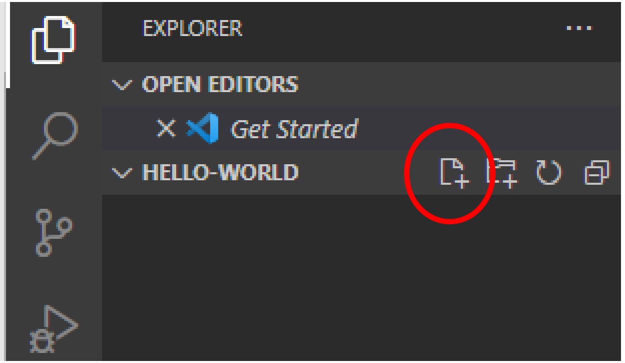
You will see that we have an empty folder called HELLO-WORLD open in our Visual Studio Code:



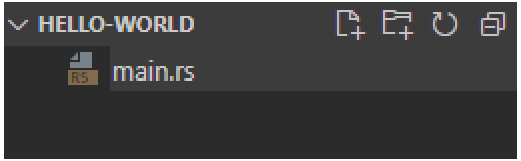


Creating main.rs

Now we can create a new File in this folder using the New File button:



A new file will be created and Visual Studio Code will prompt us to give this new file a name. We will name it “**main.rs**”



Rust source code files have the extension .rs

Visual Studio Code has automatically opened this file, ready to be edited.

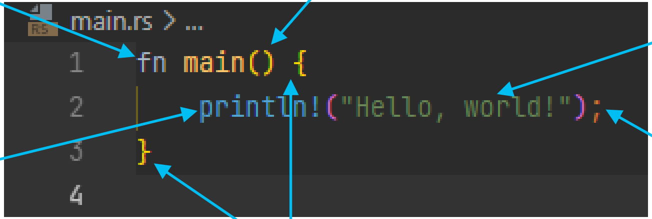
Hello World in Rust

Now we are ready to write our first lines of Rust code.

fn is short for function. The function is the basic building block of Rust

The **main()** function is where every Rust program starts from.

This is a **String slice**, also denoted **&str** which represents a sequence of characters (in this case Hello, world!)



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

```

This is the print line **macro**, it is used to output to the command line. Every macro in Rust ends with **!**

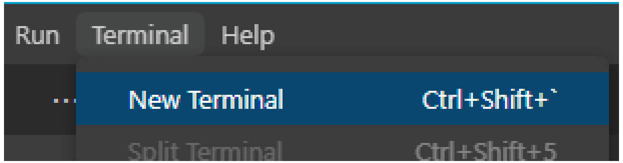
The **{}** brackets denote a block of code, also called **scope**. Every function has a start { and an end }

Every separate command of Rust must end with a semicolon

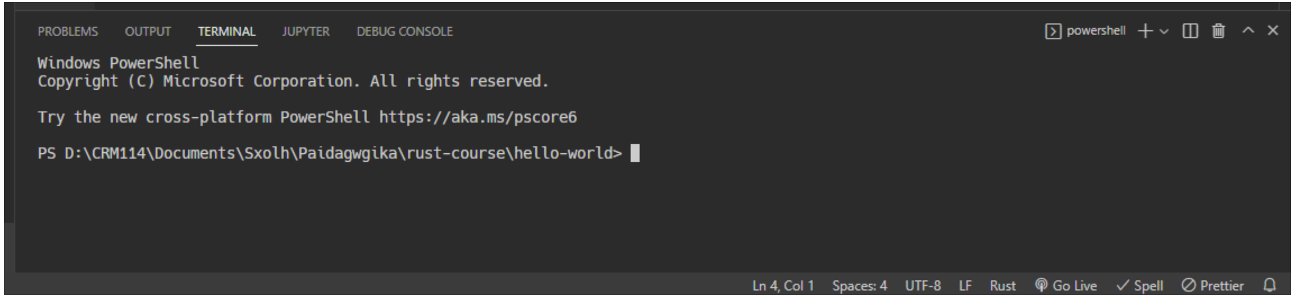
Compiling and running Rust from the command line

Now we are ready to **compile** and **run** our code.

We can open a command line (terminal) inside Visual Studio Code by going to Terminal -> New Terminal



On the bottom of Visual Studio Code you can see a new terminal appearing:



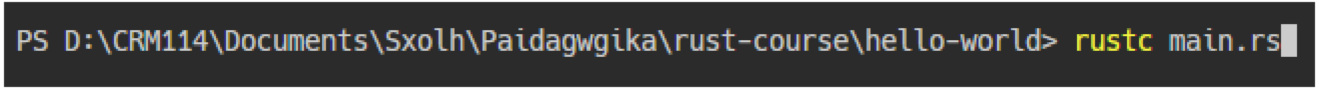
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\CRM114\Documents\Sxolh\Paidagwgika\rust-course\hello-world>
```

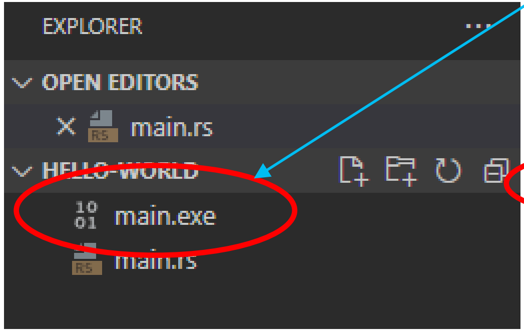
Compiling and running Rust from the command line

In order to **compile** our Rust program we will use the **rustc** command followed by our Rust file.

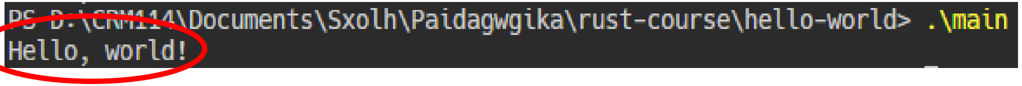


```
PS D:\CRM114\Documents\Sxolh\Paidagwgika\rust-course\hello-world> rustc main.rs
```

After running the **rustc** command we can see a new executable file called **main.exe** inside our folder:



Now we can run our program by typing **.\main** and we will see the output "Hello, world!" in the terminal:



```
PS D:\CRM114\Documents\Sxolh\Paidagwgika\rust-course\hello-world> .\main
Hello, world!
```

(the dot-slash **.** means we want the folder we are already in and then we write the name of the executable in this folder to run it)

Hello World of Rust!

Any questions?

You can find me at:

- vlasopoulos.v@gmail.com