

**Crate** | dotenvy

# ■ Environment Variables

- ◆ Store configuration in the user environment
  - Login session, command line, system service
- ◆ Setting environment variables manually:
  - **Linux/Mac**
    - ▶ `export KEY=VALUE`
  - **Windows (PowerShell)**
    - ▶ `$env:KEY="VALUE"`
  - **Windows (CMD Prompt)**
    - ▶ `set KEY="VALUE"`

# ■ dotenvy

- ◆ Configure the environment from a file
  - *.env*
- ◆ Useful for development
  - Easily change environment variables
  - Test multiple different configurations

# ■ Example *.env* File

```
LOG_LEVEL=2
```

```
PORT=1234
```

```
GREETING="Welcome!"
```

# ■ Read Environment Variables

```
use std::env;
```

```
let port = match env::var("PORT") {  
    Ok(port) => port,  
    Err(_) => "555".to_owned(),  
};
```

# ■ Dotenv: Load Environment Variables

```
use dotenv::dotenv;
```

```
use std::env;
```

```
dotenv().ok();
```

```
let port = match env::var("PORT") {  
    Ok(port) => port,  
    Err(_) => "555".to_string(),  
};
```

# ■ Recap

- ◆ Environment variables can be set with the command line or with a script
- ◆ Programs can utilize environment variables as configuration values
- ◆ The environment can be saved to a **.env** file and automatically loaded using the **dotenvy** crate
  - **.env** files use key/value pairs
    - ▶ KEY=VALUE