

Crate | color-eyre

■ color-eyre

- ◆ The *color-eyre* crate helps propagate errors when writing *application* code
 - Do not use for libraries
- ◆ Features:
 - Terminal coloring
 - Capture backtraces
 - Add context information
 - Error resolution suggestions
 - *tracing* integration

■ Setup

```
use color_eyre::eyre::Result;
```

```
fn main() -> Result<()> {  
    color_eyre::install()?;
```

```
    // ...
```

```
    Ok(())
```

```
}
```

Basic Usage

```
use color_eyre::eyre::{Result, WrapErr, eyre};
```

```
fn process() -> Result<()> {  
    Err(eyre!("process failed"))  
}
```

```
fn do_something() -> Result<()> {  
    let ans = process().wrap_err("failure in do_something()")?  
    let ans = process().wrap_err_with(||  
        format!("{}", "failure in do_something()")  
    )?;  
    Ok(())  
}
```

Output

```
Running `target/debug/crate-lecture-color-eyre`
```

```
Error:
```

```
0: failure in do_something()
```

```
1: process failed
```

```
Location:
```

```
src/main.rs:4
```

■ Suggestions

```
use color_eyre::Section;
```

```
fn quiet(msg: &str) -> Result<()> {  
    if msg.find(char::is_uppercase).is_some() {  
        Err(  
            eyre!(format!("your input '{msg}' is too loud!"))  
                .suggestion("try using all lowercase letters")  
        )  
    } else {  
        Ok(())  
    }  
}
```

```
quiet("CRUISE CONTROL")?;
```

Output

```
Running `target/debug/crate-lecture-color-eyre`
```

```
Error:
```

```
0: your input 'CRUISE CONTROL' is too loud!
```

```
Location:
```

```
src/main.rs:19
```

```
Suggestion: try using all lowercase letters
```

Recap

- ◆ *color-eyre* makes working with application-level errors easier
- ◆ It must be set up with `color_eyre::install()`
- ◆ Use:
 - *eyre!* to create a new error
 - *.wrap_err* to add context to an error
 - *.wrap_err_with* to add dynamic context
 - *.suggestion* to add a help message to an error
- ◆ Never use *color-eyre* in library code