

## # Comments

All programmers strive to make their code easy to understand, but sometimes extra explanation is warranted. In these cases, programmers leave comments in their source code that the compiler will ignore but people reading the source code may find useful.

Here's a simple comment:

```
```cairo,noplayground
// hello, world
```
```

In Cairo, the idiomatic comment style starts a comment with two slashes, and the comment continues until the end of the line. For comments that extend beyond a single line, you'll need to include `//` on each line, like this:

```
```cairo,noplayground
// So we're doing something complicated here, long enough that we need
// multiple lines of comments to do it! Whew! Hopefully, this comment will
// explain what's going on.
```
```

Comments can also be placed at the end of lines containing code:

```
```cairo
{{#include ../listings/ch02-common-programming-concepts/no_listing_25_comments/src/
lib.cairo}}
```
```

But you'll more often see them used in this format, with the comment on a separate line above the code it's annotating:

```
```cairo
{{#include ../listings/ch02-common-programming-concepts/no_listing_26_comments/src/
lib.cairo}}
```
```

## ## Item-level Documentation

Item-level documentation comments refer to specific items such as functions, implementations, traits, etc. They are prefixed with three slashes (`///`). These comments provide a detailed description of the item, examples of usage, and any conditions that might cause a panic. In case of functions, the comments may also include separate sections for parameter and return value descriptions.

```
```cairo,noplayground
{{#include ../listings/ch02-common-programming-concepts/
no_listing_37_item_doc_comments/src/lib.cairo}}
```
```

## ## Module Documentation

Module documentation comments provide an overview of the entire module, including its purpose and examples of use. These comments are meant to be placed above the module they're describing and are prefixed with `//!`. This type of documentation gives a broad understanding of what the module does and how it can be used.

```
```cairo,noplayground
{{#include ../listings/ch02-common-programming-concepts/
no_listing_38_mod_doc_comments/src/lib.cairo}}
```
```

{{#quiz ../quizzes/ch02-04-comments.toml}}