The C processor and the`#define` statement

We'll cover the following

The C prepocessor

The C Preprocessor is a part of the C compilation process that recognizes special statements, analyses them (before compilation) and acts on them as required.

The preprocessor can be used to make programs more efficient, more readable, and more portable to multiple systems.

The #define statement

You can use `#define` statements to assign symbolic names to program constants. For example:

```
#define YES 1
#define PI 3.14
#define MYNAME "paul"
```

These are all valid uses of `#define`. These statements are not like variable assignment ... no memory is allocated. Remember, the preprocessor acts **before** compilation. What is actually happening here, is quite simply, the preprocessor is going through the C code that follows, and simply substituting each instance of the symbolic name (e.g. `PI`) with the value that it's associated with in the `#define` statement (e.g. `3.14`). It's like a search-and-replace, basically.

The `#define` statments can appear anywhere in the program, although a common convention is to put them at the top of source code files.

Another use of `#define` is to create macros. More on this in the next lesson.