Parameters of main( )

In this lesson, you will learn about the parameters of main( ).

WE'LL COVER THE FOLLOWING

Parameters of `main()`

It is common for programs to take parameters from the environment that started them. For example, we have already passed a file name as a command-line option to `ls` in the previous lesson. There are two command-line options in the following line:

```
# ls -l deneme
-rwxr-xr-x 1 acehreli users 460668 Nov 6 20:38 deneme
```

The set of command-line parameters and their meanings are defined entirely by the program. Every program documents its usage, including what every parameter means.

The arguments that are used when starting a D program are passed to that program's `main()` as a slice of strings. Defining `main()` as taking a parameter of type `string[]` is sufficient to have access to program arguments. The name of this parameter is commonly abbreviated as `args`. The following program displays all of the arguments with which it is started:

```
import std.stdio;

void main(string[] args) {
    foreach (i, arg; args) {
        writefln("Argument %-3s: %s", i, arg);
    }
}
```

Code to accept string arguments

In almost all systems, the first argument is the name of the program, as entered by the user. The other arguments appear in the order that they were entered.
It is completely up to the program how it makes use of the arguments. The following program displays its two arguments in reverse order:

```
import std.stdio;

int main(string[] args) {
    if (args.length != 3) {
        stderr.writefln("ERROR! Correct usage:\n" ~
                        " %s word1 word2", args[0]);
        return 1;
    }

    writeln(args[2], ' ', args[1]);

    return 0;
}
```

Code to print arguments in reverse order

```
# ./deneme
ERROR! Correct usage:
  ./deneme word1 word2
# ./deneme word hello
hello world
```

In the next lesson, we will see command-line options and the `std.getopt` module.