Standard Input and Output Streams

This lesson explains standard input and output streams in general. Then we will see how character streams are used for input and output purposes. Finally, we will explore the usage of input and output streams in D language

WE'LL COVER THE FOLLOWING

What are standard input and output streams?

So far, the output of our programs has been displayed on the console. Although the console is often the ultimate target of output, this is not always the case. The objects that can accept the output of a program are called **standard output streams**. Similarly, the **standard input stream** comes into play when some input is taken from the user and used by the program.

Character stream

The standard output stream is *character-based*; everything displayed is first converted to the corresponding character representation and sent to the output as characters.

Example

The integer value 100 is not sent to the output as the value 100, but it is sent as three characters ‘1’, ‘0’, and ‘0’.

Similarly, the keyboard can be thought of as the standard input stream of a program, and it is also character-based. The information always comes as characters to be converted to data. For example, the integer value 42 actually comes through the standard input as the characters 4 and 2. These conversions happen automatically.

conversions happen automatically.

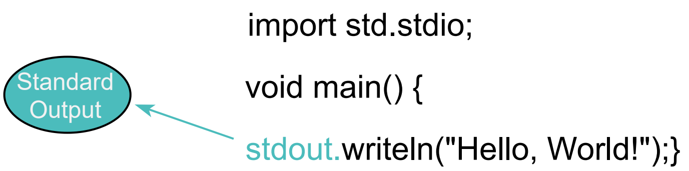
This concept of consecutive characters is called a **character stream**. As D’s standard input and standard output fit this description, they are character streams.

Standard input and output streams in D

The names of the standard input and output streams in D are `stdin` and `stdout`, respectively.

Operations on these streams normally require the *name* of the stream, a *dot* operator and the *operation*, as in `stream.operation()`. Because `stdin`'s reading methods and `stdout`'s writing methods are used very commonly, these operations can be called without the need of the stream name and the dot operator.

For example, `writeln`, which we’ve been using in the previous chapters, is actually short for `stdout.writeln`. Similarly, `write` is short for `stdout.write`. Accordingly, the “Hello World” program can also be written as follows:



```
import std.stdio;

void main() {

stdout.writeln("Hello, World!");}
```

Standard Output

Running the code below will generate the same output given in the first lesson of chapter one.

```
import std.stdio;

void main() {
    stdout.writeln("Hello, World!");
}
```

Standard output

For the sake of practice, you can modify the above code to make it work with the `write` function instead of `writeln`.

Now that we know how standard output works, let’s explore standard input in the next lesson.