An Array Example

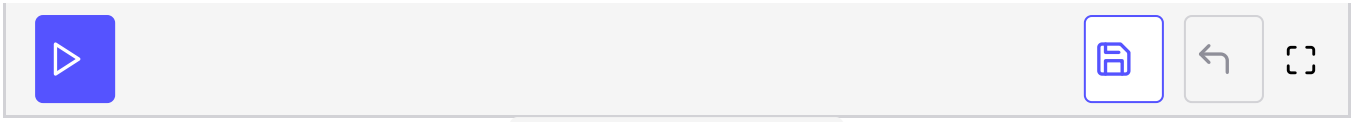
Here is a simple program that shows how arrays can simplify things in a code.

WE'LL COVER THE FOLLOWING

Example code

Let's revisit the challenge where we printed the twice of the value, but this time we will print twice of five different values using an array:

```
import std.stdio;

void main() {
    // This variable is used as a loop counter
    int counter;

    // The definition of a fixed-length array of five
    // elements of type double
    double[5] values;

    // setting the values in a loop
    writeln("values");
    while (counter < values.length) {
        values[counter]=counter*2;
        writeln(values[counter]);
        ++counter;
    }

    writeln("Twice the values:");
    counter = 0;
    while (counter < values.length) {
        writeln(values[counter] * 2);
        ++counter;
    }

    // The loop that calculates the fifths of the values would
    // be written similarly
}
```



Array code example

Observations

The value of `counter` determines how many times the loops are repeated (iterated). Iterating the loop while its value is less than `values.length` ensures that the loops are executed once per element. As the value of that variable is incremented at the end of each iteration, the `values[counter]` expression refers to the elements of the array one by one: `values[0]`, `values[1]`, etc.

How is this program better than the previous one?

To see how this program is better than the previous one, imagine needing to read 20 values. The program above would require a single change: replacing 5 with 20. On the other hand, a program that did not use an array would need to have 20 variable definitions. Furthermore, since you would be unable to use a loop to iterate the 20 values, you would also have to repeat several lines 20 times, one time for each single-valued variable.

In the next lesson, we will see array initialization and basic operations on arrays.