What is a Function?

This lesson will define what functions are and why they should be used.

A Simple Example

Consider the code below which computes the sum of all the integers in an array:

```cpp
#include <iostream>
using namespace std;

int main() {
  int arr1[] = {40, 60, 20, 15, 50, 90};
  int size = sizeof(arr1)/sizeof(arr1[0]); // Compute length of an array
  int sum = 0;
  for(int i = 0; i < size; i++){ // A for loop to add each element of the array
    sum += arr1[i];
  }
  cout << "Sum of arr1: " << sum << endl;

  int arr2[] = {2, 4, 6, 8, 10};
  size = sizeof(arr2)/sizeof(arr2[0]);
  sum = 0;
  for(int i = 0; i < size; i++){ // A for loop to add each element of the array
    sum += arr2[i];
  }
  cout << "Sum of arr2: " << sum << endl;

  int arr3[] = {18, 40, 9, 17, 36};
  size = sizeof(arr3)/sizeof(arr3[0]);
  sum = 0;
  for(int i = 0; i < size; i++){ // A for loop to add each element of the array
    sum += arr3[i];
  }
  cout << "Sum of arr3: " << sum << endl;

}
```

We can immediately notice that the code for computing the sum of its elements is pretty redundant. We set up a `for` loop each time and traverse through the array. In general, the code looks unclean.

What if there was a shortcut to compute the sum of an array in a single command? Perhaps, something like this:

```cpp
int main() {
  int arr1[] = {40, 60, 20, 15, 50, 90};
  int size = sizeof(arr1)/sizeof(arr1[0]);
  int sum = sumArray(arr1, size);
  cout << "Sum of arr1: " << sum << endl;

  int arr2[] = {2, 4, 6, 8, 10};
  size = sizeof(arr2)/sizeof(arr2[0]);
  sum = sumArray(arr2, size);
  cout << "Sum of arr2: " << sum << endl;

  int arr3[] = {18, 40, 9, 17, 36};
  size = sizeof(arr3)/sizeof(arr3[0]);
  sum = sumArray(arr3, size);
  cout << "Sum of arr3: " << sum << endl;

}
```

That looks much cleaner! The `sumArray` command mysteriously returns the sum of the array. `sumArray` is an example of a **function**.

The Definition of a Function

> A function is a set of statements that performs operations in a program.

In the example above, our function, `sumArray`, handles the `for` loop that adds each element of the array to the total sum.

2 | 4 | 6 | 8 | 10 --→ sumArray() --→ 30

Array   sumArray()   Sum
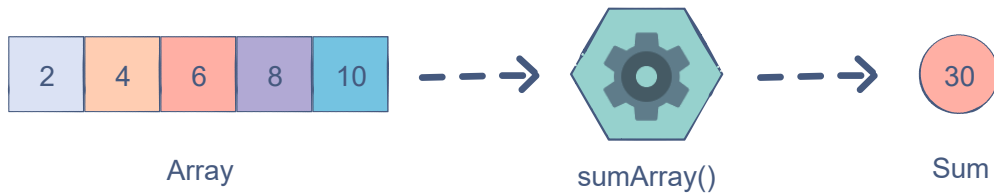
Why do We Use Functions?

In the example above, we do not need to write the code for finding the sum over and over again. The function does it for us. That is the beauty of functions in C++!

Functions save us a lot of time and make our code cleaner. They act as templates for code we need to write repeatedly. This is the main reason we use functions.

In the next lesson, we'll learn how to create our own function and call it in a program.