# STL Tutorial Introduction

1. Standard Template Library of C++ 🡪 Containers, iterators and algorithms
2. All STL containers are implemented by template classes, which means that any given container can easily be used to hold elements of virtually any type.

### Analogies

1. At a superficial level at least, any STL container is analogous to an array, in that it is something that allows you to store and retrieve elements.
2. Use pointers to "point at" and manipulate array elements. STL iterators are used to "point at" and manipulate STL container elements in a manner quite analogous to the way in which pointers are used in the array context.

Each STL container class will have its own kind of iterator.

1. Often they come in the form of stand-alone or "free" functions which usually take in some data via a parameter list and perform some task and/or return one or more computed values. And all of this is true of STL algorithms as well.

STL algorithm get their additional power and flexibility is from the fact that many of their parameters are STL iterators rather than STL containers.

An algorithm can work on the elements of a container without knowing or caring what kind of container it is.

### Pushing the analogies further with some specific (vector) examples