List

Lists are sequence containers that allow **non-contiguous** memory allocation. As compared to the vector, the list has slow traversal, but once a position has been found, insertion and deletion are quick (constant time). Normally, when we say a list, we talk about a doubly linked list.

1. To use a list, include the list header file:

```
#include <list>
```

2. The syntax to define a list is:

```
std::list <data-type> name_of_list;
```

3. It is mostly similar to vectors in terms of initialization and functions.

Functions

Most of the functions are same as vectors in list and work the same way.
There are some extra functions like **emplace_front()**, **push_front()** and **pop_front()** which perform their operations on the front element.

Points to remember about list

- It is generally implemented using a dynamic doubly linked list with traversal in both directions.
- Faster insert and delete operation as compared to arrays and vectors.
- It provides only sequential access. Random Access to any middle element is not possible
- It is defined as a template so it is able to hold any data type.
- It operates as an unsorted list would, which implies that by default, the list's order is not preserved. However, there are techniques for sorting.