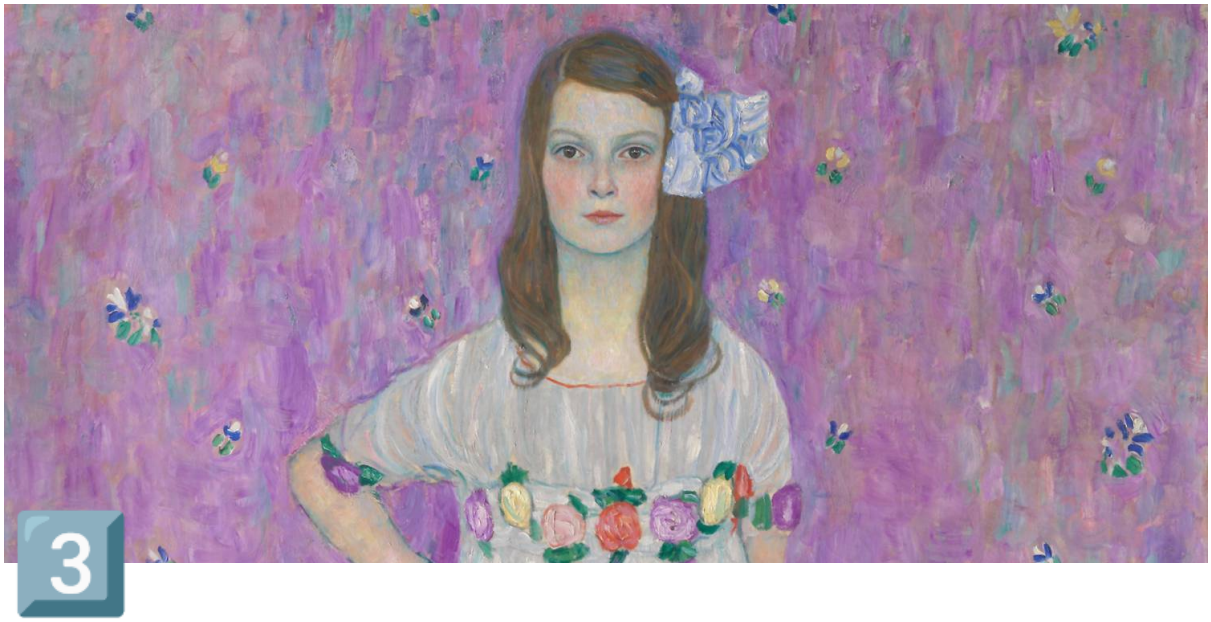

Lists

What is a List?

A list in STL is a contiguous container that allows the inserting and erasing of elements in constant time and iterating in both directions.

Syntax:

```
list<object_type> variable_name;
```

Example:

```
list<int> li;
list<string> li;
```

The Whole Code:

```
// Containers -- Lists
#include <bits/stdc++.h>
using namespace std;
```

```
void explainList()
{
	list<int> ls;
	ls.push_back(2); // {2}
	ls.emplace_back(4); // {2,4}
	ls.push_front(5);   //{5,2,4};
	ls.emplace_front(); // {2,4};

	// rest function same as vector
	// begin, end, rbegin, rend, clear, insert, size, swap

}
int main()
{
	return 0;
}
```

Functions in List:

push_back() – to insert an element at the end of the list.

```
list<int> li;
li.push_back(110);
li.push_back(220);
```

push_front() – to insert an element at the front of the list.

```
list<int> li;
li.push_front(110);
li.push_front(220);
```

pop_back() – deletes the last element of the list.

```
li.pop_back();
```

pop_front() – deletes the front element of the list.

```
li.pop_front();
```

front() – it gives a reference to the first element of the list.

```
li.front();
```

back() – it gives a reference to the last element of the list.

```
li.back();
```

reverse() – reverse the list.

```
li.reverse();
```

sort() – sorts the list in ascending order.

```
li.sort();
```

size() – returns the number of elements on the list.

```
li.size();
```

empty() – to check if the list is empty or not.

```
li.empty();
```

Striver Code for List

```
#include<bits/stdc++.h>
using namespace std;
void printlist(list<int> li)
{
    list<int>::iterator it;
    for(it=li.begin();it!=li.end();it++)
    {
        cout<<*it<<" ";
    }
    cout<<endl;
}
int main()
{
    list<int> li;
    li.push_back(10);
    li.push_back(20);
    li.push_front(30);
    li.push_front(40);
    li.push_front(50);

    cout<<"The elements in the list are: ";
    printlist(li);
    cout<<"Reversing the list: ";
    li.reverse();
    printlist(li);
    cout<<"Sorting the list: ";
    li.sort();
    printlist(li);
    cout<<"The size of the list is: "<<li.size()<<endl;
    cout<<"The first element in the list: "<<li.front()<<endl;
    cout<<"Deleting the first element"<<endl;
    li.pop_front();
    printlist(li);
    cout<<"The last element of the list: "<<li.back()<<endl;
    cout<<"Deleting the last element"<<endl;
    li.pop_back();
    printlist(li);

}

Output:

The elements in the list are: 50 40 30 10 20
Reversing the list: 20 10 30 40 50
Sorting the list: 10 20 30 40 50
The size of the list is: 5
The first element in the list: 10
Deleting the first element
20 30 40 50
The last element of the list: 50
Deleting the last element
20 30 40
```

Other functions of the list:

- **begin()** – it refers to the first element of the list.
- **end()** – it refers to the theoretical element after the last element of the list.
- **cbegin()** – it refers to the first element of the list.
- **cend()** – it refers to the theoretical element after the last element of the list.
- **rbegin()** – it points to the last element of the list.
- **rend()** – it points to the theoretical element before the first element of the list.
- **emplace_front()** – to insert an element at the front of the list.
- **emplace_back()** – to insert an element at the end of the list.
- **max_size()** – the maximum elements a list can hold.
- **clear()** – to delete all the elements of the list.
- **erase()** – to delete a single element or elements between a particular range.