

STL

- vector
- deque
- list

vector

□□□□□□□□□□

int

□□□□

```
vector<int> v1;           //□□□□□□
vector<int> v2(n);        //□□□□□□   n□□□□□□□□□□   0
vector<int> v3(n,value);  //□□□□□□   n□□□□□□□□□□   value
vector<int> v4(start,end); //□ start□ end□ int*□□□□□□□□   start□ end□□□□□□□□□□
vector<int> v5(v4);       //□□□□□□□□□□   v4□□□□
vector<int> v6={1,2,3,4,5}; //□□□□□□
```

□□□□

vector □□□□□□□□□□□□□□□□

```
vector<int> v1={1,2,3},v2={4,5,6},v3; //□□□□
```

```
v3.assign(n,value);          //□□□□□□□□   n□ value□□□□
v3.assign(v1.begin(),v1.end()); //□□□□□□□□   v1□□□□□□□□□□□□   v3□□□□□□□□□□
v3=v1;                       //□ v1□□□□   v3□□□□   =□
v1.swap(v2);                 //□□ v1□ v2
```

□□□□□□

```
vector.size();//□□□□□□
vector.empty();//□□□□□□□□
vector.resize(n);//□□□□□□□□   n□□□□□□□□□□□□□□□□□□
vector.resize(n,value);//□□□□□□□□   n□□□□□□□□   value□□□□□□□□□□
vector.clear();//□□□□□□
```

□□□□□□

vector □□□□□□□□□□□□□□□□□□□□□□

```
vector[n];           //□□□□□□□□□□□□□□□□□□□□
vector.at(n);         //vector□ at□□□□□□□□   n□□□□□□□□□□
for(vector<int>::iterator it=vector.begin();it!=vector.end();it++){ //□□□□
    cout<<*it;
```

}

vector

```
vector.push_back(value); // 在末尾添加 value
vector.pop_back(); // 删除末尾元素
vector.erase(pos); // 删除 pos 位置的元素
vector.insert(pos, value); // 在 pos 位置插入 value
vector.insert(pos, n, value); // 在 pos 位置插入 n 个 value
vector.insert(pos, begin, end); // 在 pos 位置插入 [begin, end) 范围内的元素
```

vector

方法	说明
begin()	返回指向第一个元素的迭代器
end()	返回指向末尾元素的迭代器
rbegin()	返回指向最后一个元素的逆序迭代器
rend()	返回指向末尾元素的逆序迭代器

vector




vector

```
vector.insert(pos, value); // 在 pos 位置插入 value
vector.insert(pos, n, value); // 在 pos 位置插入 n 个 value
vector.insert(pos, begin, end); // 在 pos 位置插入 [begin, end) 范围内的元素
```

deque

deque  vector  vector   _



```
deque.push_front(value);// u value
deque.pop_front();//
```

list

list

```
list<int> l1;           //
list<int> l2(5);        //          5
list<int> l3(5,9);      //          5          9
list<int> l4={1,2,3,4,5}; //          5          1 2 3 4 5
list<int> l5(l4);       //          l4
```

```
list.push_front(0); //
list.push_back(6);  //
list.pop_front();   //
list.pop_back();    //
```

```
list.front(); //
list.back();  //
```






front() back()

```
list.front()=1; //          1
list.back()=2;  //          2
```

list (it+5) (it++)

list

	list<int> v={1,2,3,4}
begin()	v[0]
end()	v[4]
rbegin()	v[3]

	 <code>list<int> v={1,2,3,4}</code> 
<code>rend()</code>	  <code>v[-1]</code>