set

```cpp
#include<set>
#include<string>
int main()
{
    std::set<std::string> setOfNumbers;

    // Lets insert four elements
    setOfNumbers.insert("first");
    setOfNumbers.insert("second");
    setOfNumbers.insert("third");
    setOfNumbers.insert("first");

    // Only 3 elements will be inserted
    std::cout<<"Set Size = "<<setOfNumbers.size()<<std::endl;

    // Iterate through all the elements in a set and display the value.
    for (std::set<std::string>::iterator it=setOfNumbers.begin(); it!=setOfNumbers.end(); ++it)
        std::cout << ' ' << *it;
    std::cout<<"\n";
    return 0;
}
```

ใช้ **iterator** เปลี่ยน **content** ไม่ได้ เพราะตัว **tree** ข้างในจะเจ๊ง

Output:

Set Size = 3

first second third

หาของใน set

```
// Search for element in set using find member function
std::set<std::string>::iterator it = setOfNumbers.find("second");
if(it != setOfNumbers.end())
    std::cout<<"'first'  found"<<std::endl;
else
    std::cout<<"'first' not found"<<std::endl;
```

ถ้าหาเจอจะรีเทิร์น **iterator** ชี้ตำแหน่งที่เจอ แต่ถ้าหาไม่เจอจะรีเทิร์น **iterator** ที่ชี้ที่ **end()**

เอาของออกจาก set

```
1 iterator  erase (const_iterator position);
2 size_type erase (const value_type& val);
3 iterator  erase (const_iterator first, const_iterator last);
```

```
setOfNumbers.erase("third");
```

เช็คว่าเซ็ตว่างมั้ย

ใช้ s.empty()

```
std::set<int> first;                                // empty set of ints

int myints[]= {10,20,30,40,50};
std::set<int> second (myints,myints+5);         // range

std::set<int> third (second);                       // a copy of second

std::set<int> fourth (second.begin(), second.end());  // iterator ctor.

std::set<int,classcomp> fifth;                      // class as Compare

bool(*fn_pt)(int,int) = fncomp;
std::set<int,bool(*)(int,int)> sixth (fn_pt);  // function pointer as Compare
```