Element Access

The variables stored inside a vector are called its **elements**. To access an individual element, you use its **subscript** (or **index**). This is called **selecting** the element. To **select** an element, pass the subscript as an argument to the `vector::at()` member function, or, place the subscript in **square brackets** after the variable name. (The square brackets are called the **subscript operator**).

```
cout << v[1] << endl;   // the subscript operator
cout << v.at(1) << endl;  // the at member function
```

Both the **subscript operator** and the `at()` member function **return a reference** to the selected element, so they may appear on either side of an assignment.

vector Size

You can find out how many elements a `vector` contains by using the `size()` member function. The first element in the `vector` is at subscript `0`, while the last is at `v.at(v.size() - 1)`. C++11 (and later) added two additional member functions, `front()` and `back` which return a reference to the first and last elements.

Calling `front()` or `back()` on an empty `vector` **is undefined**. You can find out if a `vector` contains elements by calling its `empty()` member function, or by checking if its `size()` is greater than zero.

Undefined Behavior

Subscripting a `vector` past its bounds is an error which invokes **undefined behavior** in C++. Undefined behavior means that the compiler is completely free to ignore the error (which is what usually happens). However, the compiler is also free to format your hard disk, send your credit-card credentials to Timbucktu, or, to make demons fly out of your nose.

C++ also has **implementation-defined** and **unspecified** behavior. Implementation-defined means the compiler **must pick** a particular implementation, and document it, such as the size of an integer. Unspecified means that the compiler **must do something reasonable**, but need not document what it does. The order in which arguments are evaluated when passed to a function is unspecified; it may be different each time.