I/O 格式旗標

〈I/O 格式控制器〉可以對當時處理中的串流改變格式，如果想在程式進行過程中，始終維持指定的格式，可以使用格式旗標，透過 setf 與 unsetf 方法來設定與取消。

以下列出一些常用的格式旗標：

- ios::boolalpha：將真與假以 true 與 false 顯示
- ios::dec：10 進位顯示
- ios::fixed：正常方式顯示（非科學記號）
- ios::hex：16 進位顯示
- ios::left：靠左
- ios::oct：8 進位顯示
- ios::scientific：科學記號
- ios::showbase：顯示基底
- ios::showpoint：顯示小數點
- ios::showpos：正數顯示 +
- ios::skipws：忽略空白字元
- ios::uppercase：字母大寫

可以一次設定一個格式旗標，若要設定多個格式旗標，可以使用 | 來連結，例如：

```
cout.setf(ios::showbase | ios::hex);
```

下面這個程式顯示一些基本的格式旗標作用：

```
#include <iostream>
using namespace std;
```

```cpp
int main() {
    cout.unsetf(ios::dec); // 取消 10 進位顯示
    cout.setf(ios::hex | ios::scientific); // 16 進位顯示或科學記號顯示
    cout << 12345 << "  " << 100 << endl;

    cout.setf(ios::showpos | ios::showpoint); // 正數顯示 + 號且顯示小數點
    cout << 10.0 << ": " << -10.0 << endl;

    return 0;
}
```

執行結果:

```
3039  64
+1.000000e+01: -1.000000e+01
```

在程式中先解除了 ios::dec 格式旗標，這個動作並不一定需要，但在某些編譯器中，這個旗標會覆蓋其它的旗標，先清除比較保險。

ios 類別的 flags 方法會傳回目前串流的格式設定，如果傳遞參數給它，會設定指定的格式，並傳回上一個格式設定:

```cpp
fmtflags flags();
fmtflags flags(fmtflags);
```

想一次設定指定的格式旗標，可以如下:

```cpp
ios::fmtflags f = ios::showpos | ios::showbase | ios::oct | ios::right;
cout.flags(f);
```

下面這個程式可以用來測試串流的格式設定:

```cpp
#include <iostream>
using namespace std;
```

```
void info(ios::fmtflags current, const ios::fmtflags &flag, const string &flagName)
    if(current & flag) {
        cout << flagName << " on" << endl;
    }
    else {
        cout << flagName << " off" << endl;
    }
}

int main() {
    cout.unsetf(ios::dec);
    cout.setf(ios::oct | ios::showbase);

    ios::fmtflags flags = cout.flags();

    info(flags, ios::left, "left");
    info(flags, ios::dec, "dec");
    info(flags, ios::showbase, "showbase");
    info(flags, ios::oct, "oct");

    return 0;
}
```

執行結果:

```
left off
dec off
showbase on
oct on
```