

Streams

```
#include <iostream>
using std::cout;
using std::endl;
int main() {
    cout << "Hello, world!" << endl;
    return 0;
}
```



std::cout



<<"Hello, world"

Un Stream (flujo) es una abstracción para la entrada/salida, un objeto que puede enviar y recibir datos (caracteres de longitud indefinida). Usted ya ha estado utilizando streams, enviando cadenas a stream de salida (ostream) **cout**, usando el operador de inserción de cadenas: <<

Puede escribir datos de varios tipos para objetos stream. En particular, cualquier tipo primitivo puede ser insertado en un stream! Para otros tipos, debe indicar explícitamente a C++ cómo hacerlo.

```
#include <iostream>
using std::cout;
using std::endl;

void streamTypeTests() {
    cout << "Strings work!" << endl;
    cout << 1729 << endl;
    cout << 3.14 << endl;
    cout << "Mixed types - " << 1123 << endl;
    cout << 23*2 << endl;
}

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

En el programa funcionara esto?

```
//print a number appended with 4
void append4(int n) {
    cout << n + "4" << endl;
}
```

Output Streams

Puede utilizar un ostream para algo más que simplemente imprimir datos en una consola. También puede imprimir datos en un archivo utilizando: **ofstream**

```
#include <iostream>
#include <fstream>

using std::cout;
using std::endl;
using std::string;

void writeToStream(std::ostream& anyOutputStream, int favouriteNumber) {
    anyOutputStream << "Writing to stream: "
        << favouriteNumber << endl;
}

int main() {
    // Write an int to the user's console.
    int favouriteNumber = 1729;
    cout << "Result when writing an int directly to cout: " << favouriteNumber <<
endl;

    // Write an int to a file.
    std::ofstream outputFile("out.txt");
    outputFile << "Result when writing an int to file: " << favouriteNumber << endl;

    // Write method to take any ostream
    std::ofstream fileOut("out.txt");

    writeToStream(cout, favouriteNumber);
    writeToStream(fileOut, favouriteNumber + 1);
    return 0;
}
```

Input Stream

Cualquier stream que sólo pueda dar datos, como cin, se llama flujo de entrada, o **istream**. Se envían datos mediante el operador de extracción de cadenas: >>. Obtiene datos del stream y lo convierte en el tipo apropiado.

Al igual que con un ostream, un istream puede ser usado para más que sólo la consola IO. También puede leer datos de un archivo usando un ifstream.

```

#include <iostream>
#include <fstream>

using std::cout;    using std::endl;
using std::string;

void readNumbers() {
    // Create our ifstream and make it open the file
    std::ifstream input("numbers.txt");

    // This will store the values as we get them form the stream
    int value;
    while(true) {
        // Extract next number from input
        input >> value;

        // If input is in a fail state, either a value couldn't
        // be converted, or we are at the end of the file.

        if(input.fail())
            break;
        cout << "Value read: " << value << endl;
    }
}

void readHaikuWord() {
    // Create our ifstream and make it open the file
    std::ifstream input("haiku.txt");

    // This will store the values as we get them form the stream
    string word;
    while(true) {
        // Extract next word from input
        input >> word;

        // If input is in a fail state, either a value couldn't
        // be converted, or we are at the end of the file.

        if(input.fail())
            break;

        cout << "Word read: " << word << endl;
    }
}

void readHaikuLine() {
    // Create our ifstream and make it open the file
    std::ifstream input("haiku.txt");

    // This will store the lines as we get them form the stream
    string line;
    while(true) {
        std::getline(input, line);

        // If input is in a fail state, either a value couldn't
        // be converted, or we are at the end of the file.
        if(input.fail())
            break;
        cout << line << endl;
    }
}

int main() {
    readNumbers();
    cout << "=====" << endl;
    readHaikuWord();
    cout << "=====" << endl;
    readHaikuLine();
    return 0;
}

```

Para volver a releer un archivo, debes cerrarlo, borrarlo y volver a abrirlo.

```
input.close();  
input.clear();  
input.open("haiku.txt");  
cout << endl;
```

Ejercicio

Escriba un programa que lea un archivo e informe cuántas líneas, palabras y caracteres aparecen en él. Para los propósitos de este programa, una palabra consiste en una secuencia consecutiva de cualquier carácter excepto los espacios en blanco. Por ejemplo, supongamos que el archivo Lear.txt contiene el siguiente pasaje del Rey Lear de Shakespeare,

```
Poor naked wretches, wheresoe'er you are,  
That bide the pelting of this pitiless storm,  
How shall your houseless heads and unfed sides,  
Your loop'd and window'd raggedness, defend you  
From seasons such as these? O, I have ta'en  
Too little care of this!
```

Su programa debe ser capaz de generar el siguiente ejemplo de ejecución:

```
Enter a filename: Lear.txt  
Char count: 254  
Word count: 43  
Line count: 6
```

Para esto completar las funciones en el programa abajo

```
#include <iostream>  
#include <string>  
#include <fstream>  
  
#define SENTINEL ""  
  
using namespace std;  
  
void readFile(string filename);  
int decimalPlaces(int);  
  
/* Function: readFile(string filename);  
 * -----  
 * Read the file and print out the file statistics.  
 */  
void readFile(string filename)  
{  
}  
  
/*  
 * Function: int decimalPlaces(int number);
```

```

* -----
* Return the "size" of the int, e.g., if number is
* "9" we return "1", if number is "415" we return "3",
* if number is "1000000" we return "7".
*/
int decimalPlaces(int number)
{
}

int main() {

    string filename;
    while ( true ) {
        cout<< "Enter a filename: " ;
        cin>>filename;
        //filename = getLine("Enter a filename: ");
        if ( filename == SENTINEL ) break;
        readFile(filename);
        cout << endl;
    }

    return 0;
}

```