

Apunte rápido de strings

Cómo ya mencionamos, los strings sirven para escribir cadenas de caracteres: **"casa"**, **"comida"**, **"hola152"**, etc

Además de cadenas, los strings son objetos, veremos esto más adelante. Pero lo que importa saber ahora es que poseen una serie de métodos (funciones) que nos permiten trabajar con ellos. A continuación alguno de ellos.

Constructores:

```
// Opcion 1
string nuevo1("hola a todos!");

// Opcion 2
string nuevo2 = "hola a todos!";

cout<< nuevo1<<endl; // hola a todos!
cout<< nuevo2<<endl; // hola a todos!
```

Algunos métodos:

size y length

Nos dan el tamaño del string

```
string nuevo1 = "hola a todos!";
string nuevo2 = "hola a todos!";
int tam1 = nuevo.size();
int tam2 = nuevo.length();

cout<< tam1<<endl; // 13
cout<< tam2<<endl; // 13
// los espacios tambien cuentan
```

empty

Nos informa con **true** o **false** (1 o 0) si el string está vacío

```
string nuevo = "hola a todos!";
cout << nuevo.empty() << endl; // 0 : es decir false

nuevo = "";
cout << nuevo.empty() << endl; // 1 : es decir true
```

Algunos métodos de acceso:

Métodos para poder conocer los valores dentro del string

Operador corchetes []

Nos permiten acceder a una posición puntual del string. Se coloca la posición entre los corchetes

```
string nuevo = "hola a todos!";
cout << nuevo[5]<< endl; // a

// la cadena comienza en la posición 0:
// 0: h
// 1: o
// 2: l
// 3: a
// 4:
// 5: a
// 6:
// 7: t
// 8: o
// 9: d
// 10: o
// 11: s
// 12: !
```

at

Nos permiten acceder a una posición puntual del string. Se pasa la posición entre paréntesis

```
string nuevo = "hola a todos!";
cout << nuevo.at(5)<< endl; // a
```

front

Nos muestra el primer valor del string

```
string nuevo = "hola a todos!";
cout << nuevo.front()<< endl; // h
```

back

Nos muestra el último valor del string

```
string nuevo = "hola a todos!";
cout << nuevo.back()<< endl; // !
```

Modificadores:

Métodos para modificar el string

Operador +=

Agrega un string al final

```
string nuevo = "hola a todos!";
nuevo += " Como les va?";
cout << nuevo << endl; // hola a todos! Como les va?
```

append

Agrega un string al final

```
string nuevo = "hola a todos!";
nuevo = nuevo.append(" Como les va?");
cout << nuevo << endl; // hola a todos! Como les va?
```

insert

Agrega un string en una posición definida

```
string nuevo = "hola a todos!";
// pos desde, string
nuevo = nuevo.insert(5, "hakuna ");
cout << nuevo << endl; // hola hakuna a todos!
```

erase

Borra una cadena desde una posición del string hasta otra posición

```
string nuevo = "hola a todos!";
// pos desde, pos hasta
nuevo = nuevo.erase(3, 6);
cout << nuevo << endl; // holdos!
```

replace

Reemplaza una porción de una cadena por otra nueva

```
string nuevo = "hola a todos!";
// pos desde, pos hasta, cadena a reemplazar
nuevo = nuevo.replace(1, 8, "ela");
cout << nuevo << endl; // helados!
```

pop_back

Borra el último caracter

```
string nuevo = "hola a todos!";
nuevo.pop_back();
cout << nuevo << endl; // hola a todos
```

find

Busca la primer aparición de una cadena desde la posición dada, recorre de principio a fin

```
string nuevo = "hola a todos!";
cout << nuevo.find("todos",0) << endl; // 7
cout << nuevo.find("o",0) << endl; // 1
```

rfind

Busca la primer aparición de una cadena desde la posición dada, recorre de fin a principio

```
string nuevo = "hola a todos!";
cout << nuevo.rfind("o",13) << endl; // 10
```

substr

Subtra una porción de cadena desde una posición hasta otra

```
string nuevo = "hola a todos!";
cout << nuevo.substr(3,6) << endl; // a a to
// Si no se coloca el 2do valor, se considera que va hasta el final
cout << nuevo.substr(3) << endl; // a a todos!
```

Operador ==

Subtra una porción de cadena desde una posición hasta otra

```
string nuevo = "hola a todos!";  
cout << (nuevo == "hola") << endl; // 0 : false  
cout << (nuevo == "hola a todos!") << endl; // 1 : true
```

Referencias

Para más información sobre los métodos y otros métodos de string: <http://www.cplusplus.com/reference/string/string/>