string <string>

Constantes: **string::npos** = -1
Funciones:

string::append
Append to string.

```
string& append ( const string& str );
string& append ( const string& str, size_t pos, size_t n );
string& append ( const char* s, size_t n );
string& append ( const char* s );
string& append ( size_t n, char c );
template <class InputIterator> string& append ( InputIterator first, InputIterator last );
```

string::assign
Assign content to string. Assigns new content to the string replacing its current content.

```
string& assign ( const string& str );
string& assign ( const string& str, size_t pos, size_t n );
string& assign ( const char* s, size_t n );
string& assign ( const char* s );
string& assign ( size_t n, char c );
template <class InputIterator> string& assign ( InputIterator first, InputIterator last );
```

string::at
Get character in string. Returns the character at position *pos* in the string. This member function behaves as *operator[]* , except that at also performs a range check, throwing an exception of type *out_of_range* in case that *pos* is not an actual position in the string.

```
const char& at ( size_t pos ) const;
char& at ( size_t pos );
```

string::begin
Return iterator to beginning. Returns an iterator referring to the first character in the string.

```
iterator begin();
const_iterator begin() const;
```

string::capacity
Return size of allocated storage. Returns the size of the allocated storage space in the string object.

```
size_t capacity ( ) const;
```

string::clear
Clear string. The string content is set to an empty string, erasing any previous content and thus leaving its size at 0 characters.

```
void clear();
```

string::compare
Compare strings.

```
int compare ( const string& str ) const;
int compare ( const char* s ) const;
int compare ( size_t pos1, size_t n1, const string& str ) const;
int compare ( size_t pos1, size_t n1, const char* s) const;
int compare ( size_t pos1, size_t n1, const string& str, size_t pos2, size_t n2 ) const;
int compare ( size_t pos1, size_t n1, const char* s, size_t n2) const;
```

string::copy
Copy sequence of characters from string. Copies a sequence of characters from the string content to the array pointed by *s*. This sequence of characters is made of the characters in the string that start at character position *pos* and span *n* characters from there.

```
size_t copy ( char* s, size_t n, size_t pos = 0) const;
```

string::c_str
Get C string equivalent.
```
const char* c_str ( ) const;
```

string::data

Get string data. Returns a pointer to an array of characters with the same content as the string. Notice that no terminating null character is appended (see member *c_str* for such a functionality).

```
const char* data() const;
```

string::empty

Test if string is empty. Returns whether the string is empty, i.e. whether its size is 0.

```
bool empty ( ) const;
```

string::end

Return iterator to end.

```
iterator end();
const_iterator end() const;
```

string::erase

Erase characters from string. Erases a part of the string content, shortening the length of the string.

```
string& erase ( size_t pos = 0, size_t n = npos );
iterator erase ( iterator position );
iterator erase ( iterator first, iterator last );
```

string::find

Find content in string. Searches the string for the content specified in either *str*, *s* or *c*, and returns the position of the first occurrence in the string.

```
size_t find ( const string& str, size_t pos = 0 ) const;
size_t find ( const char* s, size_t pos, size_t n ) const;
size_t find ( const char* s, size_t pos = 0 ) const;
size_t find ( char c, size_t pos = 0 ) const;
```

string::find_first_not_of

Find absence of character in string. Searches for the first character in the object which is not part of either *str*, *s* or *c*, and returns its position.

```
size_t find_first_not_of ( const string& str, size_t pos = 0 ) const;
size_t find_first_not_of ( const char* s, size_t pos, size_t n ) const;
size_t find_first_not_of ( const char* s, size_t pos = 0 ) const;
size_t find_first_not_of ( char c, size_t pos = 0 ) const;
```

string::find_first_of

Find character in string. Searches the string for any of the characters that are part of either *str*, *s* or *c*, and returns the position of the first occurrence in the string.

```
size_t find_first_of ( const string& str, size_t pos = 0 ) const;
size_t find_first_of ( const char* s, size_t pos, size_t n ) const;
size_t find_first_of ( const char* s, size_t pos = 0 ) const;
size_t find_first_of ( char c, size_t pos = 0 ) const;
```

string::find_last_not_of

Find absence of character in string from the end. Searches for the last character in the object which is not part of either *str*, *s* or *c*, and returns its position.

```
size_t find_last_not_of ( const string& str, size_t pos = npos ) const;
size_t find_last_not_of ( const char* s, size_t pos, size_t n ) const;
size_t find_last_not_of ( const char* s, size_t pos = npos ) const;
size_t find_last_not_of ( char c, size_t pos = npos ) const;
```

string::find_last_of

Find character in string from the end. Searches the string from the end for any of the characters that are part of either *str*, *s* or *c*, and returns the position of the last occurrence in the string.

```
size_t find_last_of ( const string& str, size_t pos = npos ) const;
size_t find_last_of ( const char* s, size_t pos, size_t n ) const;
size_t find_last_of ( const char* s, size_t pos = npos ) const;
size_t find_last_of ( char c, size_t pos = npos ) const;
```

string::get_allocator

Get allocator. Returns the allocator object used to constuct the object.

```
allocator<char> get_allocator( ) const;
```

string::insert

Insert into string. The current string content is extended by inserting some additional content at a specific location within the string content (this position is determined by either *pos1* or *p*, depending on the function version used).

```
string& insert ( size_t pos1, const string& str );
string& insert ( size_t pos1, const string& str, size_t pos2, size_t n );
string& insert ( size_t pos1, const char* s, size_t n);
string& insert ( size_t pos1, const char* s );
string& insert ( size_t pos1, size_t n, char c );
iterator insert ( iterator p, char c );
void insert ( iterator p, size_t n, char c );
template<class InputIterator> void insert ( iterator p, InputIterator first, InputIterator last );
```

string::length

Return length of string.

```
size_t length() const;
```

string::max_size

Return maximum size of string. Returns the maximum number of characters that the string object can hold.

```
size_t max_size ( ) const;
```

string::operator+=

Append to string.

```
string& operator+= ( const string& str );
string& operator+= ( const char* s );
string& operator+= ( char c );
```

string::operator=

String assignment.

```
string& operator= ( const string& str );
string& operator= ( const char* s );
string& operator= ( char c );
```

string::operator[]

Get character in string.

```
const char& operator[] ( size_t pos ) const;
char& operator[] ( size_t pos );
```

string::push_back

Append character to string.

```
void push_back ( char c );
```

string::rbegin

Return reverse iterator to reverse beginning. Returns a reverse iterator referring to the last character in the string, which is considered the reverse beginning.

```
reverse_iterator rbegin();
const_reverse_iterator rbegin() const;
```

string::rend

Return reverse iterator to reverse end. Returns a reverse iterator referring to the element right before the first character in the string, which is considered the reverse end. rend refers to the character right before the one that would be referred to by member begin.

```
reverse_iterator rend();
const_reverse_iterator rend() const;
```

string::reserve

Request a change in capacity. Requests that the capacity of the allocated storage space in the string be at least *res_arg*.

```
void reserve ( size_t res_arg=0 );
```

string::replace

Replace part of string.

```
string& replace ( size_t pos1, size_t n1,   const string& str );
string& replace ( iterator i1, iterator i2, const string& str );
string& replace ( size_t pos1, size_t n1, const string& str, size_t pos2, size_t n2 );
string& replace ( size_t pos1, size_t n1,   const char* s, size_t n2 );
string& replace ( iterator i1, iterator i2, const char* s, size_t n2 );
string& replace ( size_t pos1, size_t n1,   const char* s );
string& replace ( iterator i1, iterator i2, const char* s );
string& replace ( size_t pos1, size_t n1,   size_t n2, char c );
string& replace ( iterator i1, iterator i2, size_t n2, char c );
template<class InputIterator> string& replace ( iterator i1, iterator i2, InputIterator j1, InputIterator j2 );
```

string::resize

Resize string. Resizes the string content to *n* characters.

```
void resize ( size_t n, char c );
void resize ( size_t n );
```

string::rfind

Find last occurrence of content in string.

```
size_t rfind ( const string& str, size_t pos = npos ) const;
size_t rfind ( const char* s, size_t pos, size_t n ) const;
size_t rfind ( const char* s, size_t pos = npos ) const;
size_t rfind ( char c, size_t pos = npos ) const;
```

string::size

Return length of string.

```
size_t size() const;
```

string::substr

Generate substring.

```
string substr ( size_t pos = 0, size_t n = npos ) const;
```

string::swap

Swap contents with another string.

```
void swap ( string& str );
```

Funciones globales:

swap

Swap contents of two strings.

```
void swap ( string& lhs, string& rhs);
```

getline

Get line from stream. Extracts characters from is and stores them into str until a delimitation character is found. The delimiter character is delim for the first function version, and '\n' (newline character) for the second. The extraction also stops if the end of file is reached in is or if some other error occurs during the input operation. If the delimiter is found, it is extracted and discarded, i.e. it is not stored and the next input operation will begin after it.

```
istream& getline ( istream& is, string& str, char delim );
istream& getline ( istream& is, string& str );
```