SFML

* [Main Page](http://docs.google.com/index.htm)
* [Modules](http://docs.google.com/modules.htm)
* [Classes](http://docs.google.com/annotated.htm)
* [Files](http://docs.google.com/files.htm)
* [Class List](http://docs.google.com/annotated.htm)
* [Class Index](http://docs.google.com/classes.htm)
* [Class Hierarchy](http://docs.google.com/hierarchy.htm)
* [Class Members](http://docs.google.com/functions.htm)
* **sf**
* [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)

[Public Member Functions](#_gjdgxs) | [Public Attributes](#_30j0zll) | [Related Functions](#_1fob9te) | [List of all members](http://docs.google.com/classsf_1_1Vector3-members.htm)

sf::Vector3< T > Class Template Reference

[System module](http://docs.google.com/group__system.htm)

Utility template class for manipulating 3-dimensional vectors. [More...](http://docs.google.com/classsf_1_1Vector3.htm#details)

#include <[Vector3.hpp](http://docs.google.com/Vector3_8hpp_source.htm)>

| Public Member Functions | |
| --- | --- |
|  | [Vector3](http://docs.google.com/classsf_1_1Vector3.htm#aee8be1985c6e45e381ad4071265636f9) () |
|  | Default constructor. |
|  | |
|  | [Vector3](http://docs.google.com/classsf_1_1Vector3.htm#a99ed75b68f58adfa3e9fa0561b424bf6) (T X, T Y, T Z) |
|  | Construct the vector from its coordinates. |
|  | |
| template<typename U > | |
|  | [Vector3](http://docs.google.com/classsf_1_1Vector3.htm#adb2b2e150025e97ccfa96219bbed59d1) (const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< U > &vector) |
|  | Construct the vector from another type of vector. |
|  | |
| template<typename T > | |
|  | **Vector3** (T X, T Y, T Z) |
|  | |
| template<typename U > | |
|  | **Vector3** (const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< U > &vector) |
|  | |

| Public Attributes | |
| --- | --- |
| T | [x](http://docs.google.com/classsf_1_1Vector3.htm#a3cb0c769390bc37c346bb1a69e510d16) |
|  | X coordinate of the vector. |
|  | |
| T | [y](http://docs.google.com/classsf_1_1Vector3.htm#a6590d50ccb862c5efc5512e974e9b794) |
|  | Y coordinate of the vector. |
|  | |
| T | [z](http://docs.google.com/classsf_1_1Vector3.htm#a2f36ab4b552c028e3a9734c1ad4df7d1) |
|  | Z coordinate of the vector. |
|  | |

| Related Functions | |
| --- | --- |
| (Note that these are not member functions.) | |
| template<typename T > | |
| [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > | [operator-](http://docs.google.com/classsf_1_1Vector3.htm#a9b75d2fb9b0f2fd9fe33f8f06f9dda75) (const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > &left) |
|  | Overload of unary operator -. |
|  | |
| template<typename T > | |
| [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | [operator+=](http://docs.google.com/classsf_1_1Vector3.htm#abc28859af163c63318ea2723b81c5ad9) ([Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > &left, const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > &right) |
|  | Overload of binary operator +=. |
|  | |
| template<typename T > | |
| [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | [operator-=](http://docs.google.com/classsf_1_1Vector3.htm#aa465672d2a4ee5fd354e585cf08d2ab9) ([Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > &left, const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > &right) |
|  | Overload of binary operator -=. |
|  | |
| template<typename T > | |
| [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > | [operator+](http://docs.google.com/classsf_1_1Vector3.htm#a6500a0cb00e07801e9e9d7e96852ddd3) (const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > &left, const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > &right) |
|  | Overload of binary operator +. |
|  | |
| template<typename T > | |
| [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > | [operator-](http://docs.google.com/classsf_1_1Vector3.htm#abe0b9411c00cf807bf8a5f835874bd2a) (const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > &left, const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > &right) |
|  | Overload of binary operator -. |
|  | |
| template<typename T > | |
| [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > | [operator\*](http://docs.google.com/classsf_1_1Vector3.htm#a44ec312b31c1a85dcff4863795f98329) (const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > &left, T right) |
|  | Overload of binary operator \*. |
|  | |
| template<typename T > | |
| [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > | [operator\*](http://docs.google.com/classsf_1_1Vector3.htm#aa6f2b0d9f79c1b9774759b7087affbb1) (T left, const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > &right) |
|  | Overload of binary operator \*. |
|  | |
| template<typename T > | |
| [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | [operator\*=](http://docs.google.com/classsf_1_1Vector3.htm#ad5fb972775ce8ab58cd9670789e806a7) ([Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > &left, T right) |
|  | Overload of binary operator \*=. |
|  | |
| template<typename T > | |
| [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > | [operator/](http://docs.google.com/classsf_1_1Vector3.htm#ad4ba4a83de236ddeb92a7b759187e90d) (const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > &left, T right) |
|  | Overload of binary operator /. |
|  | |
| template<typename T > | |
| [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | [operator/=](http://docs.google.com/classsf_1_1Vector3.htm#a8995a700f9dffccc6dddb3696ae17b64) ([Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > &left, T right) |
|  | Overload of binary operator /=. |
|  | |
| template<typename T > | |
| bool | [operator==](http://docs.google.com/classsf_1_1Vector3.htm#a388d72db973306a35ba467016b3dee30) (const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > &left, const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > &right) |
|  | Overload of binary operator ==. |
|  | |
| template<typename T > | |
| bool | [operator!=](http://docs.google.com/classsf_1_1Vector3.htm#a608500d1ad3b78082cb5bb4356742bd4) (const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > &left, const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > &right) |
|  | Overload of binary operator !=. |
|  | |

## Detailed Description

template<typename T>

class sf::Vector3< T >

Utility template class for manipulating 3-dimensional vectors.

[sf::Vector3](http://docs.google.com/classsf_1_1Vector3.htm) is a simple class that defines a mathematical vector with three coordinates (x, y and z).

It can be used to represent anything that has three dimensions: a size, a point, a velocity, etc.

The template parameter T is the type of the coordinates. It can be any type that supports arithmetic operations (+, -, /, \*) and comparisons (==, !=), for example int or float.

You generally don't have to care about the templated form (sf::Vector3<T>), the most common specializations have special typedefs:

* sf::Vector3<float> is sf::Vector3f
* sf::Vector3<int> is sf::Vector3i

The [sf::Vector3](http://docs.google.com/classsf_1_1Vector3.htm) class has a small and simple interface, its x and y members can be accessed directly (there's no accessor like setX(), getX()) and it contains no mathematical function like dot product, cross product, length, etc.

Usage example:

[sf::Vector3f](http://docs.google.com/classsf_1_1Vector3.htm) v1(16.5f, 24.f, -8.2f);

v1.x = 18.2f;

float y = v1.y;

float [z](http://docs.google.com/classsf_1_1Vector3.htm#a2f36ab4b552c028e3a9734c1ad4df7d1) = v1.z;

[sf::Vector3f](http://docs.google.com/classsf_1_1Vector3.htm) v2 = v1 \* 5.f;

[sf::Vector3f](http://docs.google.com/classsf_1_1Vector3.htm) v3;

v3 = v1 + v2;

bool different = (v2 != v3);

Note: for 2-dimensional vectors, see [sf::Vector2](http://docs.google.com/classsf_1_1Vector2.htm).

Definition at line [37](http://docs.google.com/Vector3_8hpp_source.htm#l00037) of file [Vector3.hpp](http://docs.google.com/Vector3_8hpp_source.htm).

## Constructor & Destructor Documentation

template<typename T>

| [sf::Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T >::[Vector3](http://docs.google.com/classsf_1_1Vector3.htm) | ( |  | ) |  |
| --- | --- | --- | --- | --- |

Default constructor.

Creates a Vector3(0, 0, 0).

template<typename T>

| [sf::Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T >::[Vector3](http://docs.google.com/classsf_1_1Vector3.htm) | ( | T | *X*, |
| --- | --- | --- | --- |
|  |  | T | *Y*, |
|  |  | T | *Z* |
|  | ) |  |  |

Construct the vector from its coordinates.

Parameters

| X | X coordinate |
| --- | --- |
| Y | Y coordinate |
| Z | Z coordinate |

template<typename T>

template<typename U >

| | [sf::Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T >::[Vector3](http://docs.google.com/classsf_1_1Vector3.htm) | ( | const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< U > & | *vector* | ) |  | | --- | --- | --- | --- | --- | --- | | explicit |
| --- | --- | --- | --- | --- | --- | --- | --- |

Construct the vector from another type of vector.

This constructor doesn't replace the copy constructor, it's called only when U != T. A call to this constructor will fail to compile if U is not convertible to T.

Parameters

| vector | Vector to convert |
| --- | --- |

## Friends And Related Function Documentation

template<typename T >

| | bool operator!= | ( | const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | *left*, | | --- | --- | --- | --- | |  |  | const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | *right* | |  | ) |  |  | | related |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Overload of binary operator !=.

This operator compares strict difference between two vectors.

Parameters

| left | Left operand (a vector) |
| --- | --- |
| right | Right operand (a vector) |

ReturnsTrue if *left* is not equal to *right*

template<typename T >

| | [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > operator\* | ( | const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | *left*, | | --- | --- | --- | --- | |  |  | T | *right* | |  | ) |  |  | | related |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Overload of binary operator \*.

Parameters

| left | Left operand (a vector) |
| --- | --- |
| right | Right operand (a scalar value) |

ReturnsMemberwise multiplication by *right*

template<typename T >

| | [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > operator\* | ( | T | *left*, | | --- | --- | --- | --- | |  |  | const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | *right* | |  | ) |  |  | | related |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Overload of binary operator \*.

Parameters

| left | Left operand (a scalar value) |
| --- | --- |
| right | Right operand (a vector) |

ReturnsMemberwise multiplication by *left*

template<typename T >

| | [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & operator\*= | ( | [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | *left*, | | --- | --- | --- | --- | |  |  | T | *right* | |  | ) |  |  | | related |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Overload of binary operator \*=.

This operator performs a memberwise multiplication by *right*, and assigns the result to *left*.

Parameters

| left | Left operand (a vector) |
| --- | --- |
| right | Right operand (a scalar value) |

ReturnsReference to *left*

template<typename T >

| | [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > operator+ | ( | const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | *left*, | | --- | --- | --- | --- | |  |  | const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | *right* | |  | ) |  |  | | related |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Overload of binary operator +.

Parameters

| left | Left operand (a vector) |
| --- | --- |
| right | Right operand (a vector) |

ReturnsMemberwise addition of both vectors

template<typename T >

| | [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & operator+= | ( | [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | *left*, | | --- | --- | --- | --- | |  |  | const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | *right* | |  | ) |  |  | | related |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Overload of binary operator +=.

This operator performs a memberwise addition of both vectors, and assigns the result to *left*.

Parameters

| left | Left operand (a vector) |
| --- | --- |
| right | Right operand (a vector) |

ReturnsReference to *left*

template<typename T >

| | [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > operator- | ( | const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | *left* | ) |  | | --- | --- | --- | --- | --- | --- | | related |
| --- | --- | --- | --- | --- | --- | --- | --- |

Overload of unary operator -.

Parameters

| left | Vector to negate |
| --- | --- |

ReturnsMemberwise opposite of the vector

template<typename T >

| | [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > operator- | ( | const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | *left*, | | --- | --- | --- | --- | |  |  | const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | *right* | |  | ) |  |  | | related |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Overload of binary operator -.

Parameters

| left | Left operand (a vector) |
| --- | --- |
| right | Right operand (a vector) |

ReturnsMemberwise subtraction of both vectors

template<typename T >

| | [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & operator-= | ( | [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | *left*, | | --- | --- | --- | --- | |  |  | const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | *right* | |  | ) |  |  | | related |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Overload of binary operator -=.

This operator performs a memberwise subtraction of both vectors, and assigns the result to *left*.

Parameters

| left | Left operand (a vector) |
| --- | --- |
| right | Right operand (a vector) |

ReturnsReference to *left*

template<typename T >

| | [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > operator/ | ( | const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | *left*, | | --- | --- | --- | --- | |  |  | T | *right* | |  | ) |  |  | | related |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Overload of binary operator /.

Parameters

| left | Left operand (a vector) |
| --- | --- |
| right | Right operand (a scalar value) |

ReturnsMemberwise division by *right*

template<typename T >

| | [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & operator/= | ( | [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | *left*, | | --- | --- | --- | --- | |  |  | T | *right* | |  | ) |  |  | | related |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Overload of binary operator /=.

This operator performs a memberwise division by *right*, and assigns the result to *left*.

Parameters

| left | Left operand (a vector) |
| --- | --- |
| right | Right operand (a scalar value) |

ReturnsReference to *left*

template<typename T >

| | bool operator== | ( | const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | *left*, | | --- | --- | --- | --- | |  |  | const [Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T > & | *right* | |  | ) |  |  | | related |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Overload of binary operator ==.

This operator compares strict equality between two vectors.

Parameters

| left | Left operand (a vector) |
| --- | --- |
| right | Right operand (a vector) |

ReturnsTrue if *left* is equal to *right*

## Member Data Documentation

template<typename T>

| T [sf::Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T >::x |
| --- |

X coordinate of the vector.

Definition at line [76](http://docs.google.com/Vector3_8hpp_source.htm#l00076) of file [Vector3.hpp](http://docs.google.com/Vector3_8hpp_source.htm).

template<typename T>

| T [sf::Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T >::y |
| --- |

Y coordinate of the vector.

Definition at line [77](http://docs.google.com/Vector3_8hpp_source.htm#l00077) of file [Vector3.hpp](http://docs.google.com/Vector3_8hpp_source.htm).

template<typename T>

| T [sf::Vector3](http://docs.google.com/classsf_1_1Vector3.htm)< T >::z |
| --- |

Z coordinate of the vector.

Definition at line [78](http://docs.google.com/Vector3_8hpp_source.htm#l00078) of file [Vector3.hpp](http://docs.google.com/Vector3_8hpp_source.htm).

The documentation for this class was generated from the following files:

* [Vector3.hpp](http://docs.google.com/Vector3_8hpp_source.htm)
* [Vector3.inl](http://docs.google.com/Vector3_8inl_source.htm)

Copyright � Laurent Gomila  ::  Documentation generated by [doxygen](http://www.doxygen.org/)  ::