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**
* [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)

[Public Member Functions](#_gjdgxs) | [Public Attributes](#_30j0zll) | [Related Functions](#_1fob9te) | [List of all members](http://docs.google.com/classsf_1_1Vector2-members.htm)

sf::Vector2< T > Class Template Reference

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

Utility template class for manipulating 2-dimensional vectors. [More...](http://docs.google.com/classsf_1_1Vector2.htm#details)

#include <[Vector2.hpp](http://docs.google.com/Vector2_8hpp_source.htm)>

| Public Member Functions | |
| --- | --- |
|  | [Vector2](http://docs.google.com/classsf_1_1Vector2.htm#a58c32383b5291380db4b43a289f75988) () |
|  | Default constructor. |
|  | |
|  | [Vector2](http://docs.google.com/classsf_1_1Vector2.htm#aed26a72164e59e8a4a0aeee2049568f1) (T X, T Y) |
|  | Construct the vector from its coordinates. |
|  | |
| template<typename U > | |
|  | [Vector2](http://docs.google.com/classsf_1_1Vector2.htm#a3da455e0ae3f8ff6d2fe36d10b332d10) (const [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< U > &vector) |
|  | Construct the vector from another type of vector. |
|  | |
| template<typename T > | |
|  | **Vector2** (T X, T Y) |
|  | |
| template<typename U > | |
|  | **Vector2** (const [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< U > &vector) |
|  | |

| Public Attributes | |
| --- | --- |
| T | [x](http://docs.google.com/classsf_1_1Vector2.htm#a1e6ad77fa155f3753bfb92699bd28141) |
|  | X coordinate of the vector. |
|  | |
| T | [y](http://docs.google.com/classsf_1_1Vector2.htm#a420f2481b015f4eb929c75f2af564299) |
|  | Y coordinate of the vector. |
|  | |

| Related Functions | |
| --- | --- |
| (Note that these are not member functions.) | |
| template<typename T > | |
| [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > | [operator-](http://docs.google.com/classsf_1_1Vector2.htm#a3885c2e66dc427cec7eaa178d59d8e8b) (const [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > &right) |
|  | Overload of unary operator -. |
|  | |
| template<typename T > | |
| [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > & | [operator+=](http://docs.google.com/classsf_1_1Vector2.htm#ad4b7a9d355d57790bfc7df0ade8bb628) ([Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > &left, const [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > &right) |
|  | Overload of binary operator +=. |
|  | |
| template<typename T > | |
| [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > & | [operator-=](http://docs.google.com/classsf_1_1Vector2.htm#a30a5a12ad03c9a3a982a0a313bf84e6f) ([Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > &left, const [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > &right) |
|  | Overload of binary operator -=. |
|  | |
| template<typename T > | |
| [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > | [operator+](http://docs.google.com/classsf_1_1Vector2.htm#a72421239823c38a6b780c86a710ead07) (const [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > &left, const [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > &right) |
|  | Overload of binary operator +. |
|  | |
| template<typename T > | |
| [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > | [operator-](http://docs.google.com/classsf_1_1Vector2.htm#ad027adae53ec547a86c20deeb05c9e85) (const [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > &left, const [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > &right) |
|  | Overload of binary operator -. |
|  | |
| template<typename T > | |
| [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > | [operator\*](http://docs.google.com/classsf_1_1Vector2.htm#a5f48ca928995b41c89f155afe8d16b02) (const [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > &left, T right) |
|  | Overload of binary operator \*. |
|  | |
| template<typename T > | |
| [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > | [operator\*](http://docs.google.com/classsf_1_1Vector2.htm#ad8b3e1cf7b156a984bc1427539ca8605) (T left, const [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > &right) |
|  | Overload of binary operator \*. |
|  | |
| template<typename T > | |
| [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > & | [operator\*=](http://docs.google.com/classsf_1_1Vector2.htm#abea24cb28c0d6e2957e259ba4e65d70e) ([Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > &left, T right) |
|  | Overload of binary operator \*=. |
|  | |
| template<typename T > | |
| [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > | [operator/](http://docs.google.com/classsf_1_1Vector2.htm#a7409dd89cb3aad6c3bc6622311107311) (const [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > &left, T right) |
|  | Overload of binary operator /. |
|  | |
| template<typename T > | |
| [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > & | [operator/=](http://docs.google.com/classsf_1_1Vector2.htm#ac4d293c9dc7954ccfd5e373972f38b03) ([Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > &left, T right) |
|  | Overload of binary operator /=. |
|  | |
| template<typename T > | |
| bool | [operator==](http://docs.google.com/classsf_1_1Vector2.htm#a9a7b2d36c3850828fdb651facfd25136) (const [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > &left, const [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > &right) |
|  | Overload of binary operator ==. |
|  | |
| template<typename T > | |
| bool | [operator!=](http://docs.google.com/classsf_1_1Vector2.htm#a01673da35ef9c52d0e54b8263549a956) (const [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > &left, const [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > &right) |
|  | Overload of binary operator !=. |
|  | |

## Detailed Description

template<typename T>

class sf::Vector2< T >

Utility template class for manipulating 2-dimensional vectors.

[sf::Vector2](http://docs.google.com/classsf_1_1Vector2.htm) is a simple class that defines a mathematical vector with two coordinates (x and y).

It can be used to represent anything that has two 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::Vector2<T>), the most common specializations have special typedefs:

* sf::Vector2<float> is sf::Vector2f
* sf::Vector2<int> is sf::Vector2i
* sf::Vector2<unsigned int> is sf::Vector2u

The [sf::Vector2](http://docs.google.com/classsf_1_1Vector2.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::Vector2f](http://docs.google.com/classsf_1_1Vector2.htm) v1(16.5f, 24.f);

v1.x = 18.2f;

float [y](http://docs.google.com/classsf_1_1Vector2.htm#a420f2481b015f4eb929c75f2af564299) = v1.y;

[sf::Vector2f](http://docs.google.com/classsf_1_1Vector2.htm) v2 = v1 \* 5.f;

[sf::Vector2f](http://docs.google.com/classsf_1_1Vector2.htm) v3;

v3 = v1 + v2;

bool different = (v2 != v3);

Note: for 3-dimensional vectors, see [sf::Vector3](http://docs.google.com/classsf_1_1Vector3.htm).

Definition at line [37](http://docs.google.com/Vector2_8hpp_source.htm#l00037) of file [Vector2.hpp](http://docs.google.com/Vector2_8hpp_source.htm).

## Constructor & Destructor Documentation

template<typename T>

| [sf::Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T >::[Vector2](http://docs.google.com/classsf_1_1Vector2.htm) | ( |  | ) |  |
| --- | --- | --- | --- | --- |

Default constructor.

Creates a Vector2(0, 0).

template<typename T>

| [sf::Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T >::[Vector2](http://docs.google.com/classsf_1_1Vector2.htm) | ( | T | *X*, |
| --- | --- | --- | --- |
|  |  | T | *Y* |
|  | ) |  |  |

Construct the vector from its coordinates.

Parameters

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

template<typename T>

template<typename U >

| | [sf::Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T >::[Vector2](http://docs.google.com/classsf_1_1Vector2.htm) | ( | const [Vector2](http://docs.google.com/classsf_1_1Vector2.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 [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > & | *left*, | | --- | --- | --- | --- | |  |  | const [Vector2](http://docs.google.com/classsf_1_1Vector2.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 >

| | [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > operator\* | ( | const [Vector2](http://docs.google.com/classsf_1_1Vector2.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 >

| | [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > operator\* | ( | T | *left*, | | --- | --- | --- | --- | |  |  | const [Vector2](http://docs.google.com/classsf_1_1Vector2.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 >

| | [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > & operator\*= | ( | [Vector2](http://docs.google.com/classsf_1_1Vector2.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 >

| | [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > operator+ | ( | const [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > & | *left*, | | --- | --- | --- | --- | |  |  | const [Vector2](http://docs.google.com/classsf_1_1Vector2.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 >

| | [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > & operator+= | ( | [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > & | *left*, | | --- | --- | --- | --- | |  |  | const [Vector2](http://docs.google.com/classsf_1_1Vector2.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 >

| | [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > operator- | ( | const [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > & | *right* | ) |  | | --- | --- | --- | --- | --- | --- | | related |
| --- | --- | --- | --- | --- | --- | --- | --- |

Overload of unary operator -.

Parameters

| right | Vector to negate |
| --- | --- |

ReturnsMemberwise opposite of the vector

template<typename T >

| | [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > operator- | ( | const [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > & | *left*, | | --- | --- | --- | --- | |  |  | const [Vector2](http://docs.google.com/classsf_1_1Vector2.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 >

| | [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > & operator-= | ( | [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > & | *left*, | | --- | --- | --- | --- | |  |  | const [Vector2](http://docs.google.com/classsf_1_1Vector2.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 >

| | [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > operator/ | ( | const [Vector2](http://docs.google.com/classsf_1_1Vector2.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 >

| | [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > & operator/= | ( | [Vector2](http://docs.google.com/classsf_1_1Vector2.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 [Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T > & | *left*, | | --- | --- | --- | --- | |  |  | const [Vector2](http://docs.google.com/classsf_1_1Vector2.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::Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T >::x |
| --- |

X coordinate of the vector.

Definition at line [75](http://docs.google.com/Vector2_8hpp_source.htm#l00075) of file [Vector2.hpp](http://docs.google.com/Vector2_8hpp_source.htm).

template<typename T>

| T [sf::Vector2](http://docs.google.com/classsf_1_1Vector2.htm)< T >::y |
| --- |

Y coordinate of the vector.

Definition at line [76](http://docs.google.com/Vector2_8hpp_source.htm#l00076) of file [Vector2.hpp](http://docs.google.com/Vector2_8hpp_source.htm).

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

* [Vector2.hpp](http://docs.google.com/Vector2_8hpp_source.htm)
* [Vector2.inl](http://docs.google.com/Vector2_8inl_source.htm)

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