

sf::Vector2<T> Class

- sf::Vector2 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<int>** is **sf::Vector2i** // Integer type
 - **sf::Vector2<float>** is **sf::Vector2f** // float type
 - **sf::Vector2<unsigned int>** is **sf::Vector2u** // unsigned Integer

Accessing x and y members of vector2< T> class

The sf::Vector2 class has a small and simple interface, its x and y members can be accessed directly (there are no accessors like setX(), getX()) and it contains no mathematical function like dot product, cross product, length, etc.

```
sf::Vector2f v1(16.5f, 24.f);
v1.x = 18.2f;
float y = v1.y;

sf::Vector2f v2 = v1 * 5.f;

sf::Vector2f v3;
v3 = v1 + v2;
bool different = (v2 != v3);

// f is suffix that indicates float values, by default compiler consider double values in c++
```

For three coordinates we use sf::Vector3<T> which takes three arguments (x, y, z).

To know more about sf::Vector3<T> then check out the below link :)

sf::Vector3 Class Template Reference (SFML / Learn / 2.5.1 Documentation)

Utility template class for manipulating 3-dimensional vectors. sf::Vector3 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.

👉 https://www.sfml-dev.org/documentation/2.5.1/classsf_1_1Vector3.php#details

