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**
* [Font](http://docs.google.com/classsf_1_1Font.htm)

[Classes](#1fob9te) | [Public Member Functions](#_gjdgxs) | [List of all members](http://docs.google.com/classsf_1_1Font-members.htm)

sf::Font Class Reference

[Graphics module](http://docs.google.com/group__graphics.htm)

Class for loading and manipulating character fonts. [More...](http://docs.google.com/classsf_1_1Font.htm#details)

#include <[Font.hpp](http://docs.google.com/Font_8hpp_source.htm)>

| Public Member Functions | |
| --- | --- |
|  | [Font](http://docs.google.com/classsf_1_1Font.htm#a506404655b8869ed60d1e7709812f583) () |
|  | Default constructor. |
|  | |
|  | [Font](http://docs.google.com/classsf_1_1Font.htm#a72d7322b355ee2f1be4500f530e98081) (const [Font](http://docs.google.com/classsf_1_1Font.htm) &copy) |
|  | Copy constructor. |
|  | |
|  | [~Font](http://docs.google.com/classsf_1_1Font.htm#aa18a3c62e6e01e9a21c531b5cad4b7f2) () |
|  | Destructor. |
|  | |
| bool | [loadFromFile](http://docs.google.com/classsf_1_1Font.htm#ab020052ef4e01f6c749a85571c0f3fd1) (const std::string &filename) |
|  | Load the font from a file. |
|  | |
| bool | [loadFromMemory](http://docs.google.com/classsf_1_1Font.htm#abf2f8d6de31eb4e1db02e061c323e346) (const void \*data, std::size\_t sizeInBytes) |
|  | Load the font from a file in memory. |
|  | |
| bool | [loadFromStream](http://docs.google.com/classsf_1_1Font.htm#abc3f37a354ce8b9a21f8eb93bd9fdafb) ([InputStream](http://docs.google.com/classsf_1_1InputStream.htm) &stream) |
|  | Load the font from a custom stream. |
|  | |
| const [Glyph](http://docs.google.com/classsf_1_1Glyph.htm) & | [getGlyph](http://docs.google.com/classsf_1_1Font.htm#a148eb92890113052f12f8a231ad619b9) (Uint32 codePoint, unsigned int characterSize, bool bold) const |
|  | Retrieve a glyph of the font. |
|  | |
| int | [getKerning](http://docs.google.com/classsf_1_1Font.htm#a4093f7d2d195c88ea90b34cf14e003c8) (Uint32 first, Uint32 second, unsigned int characterSize) const |
|  | Get the kerning offset of two glyphs. |
|  | |
| int | [getLineSpacing](http://docs.google.com/classsf_1_1Font.htm#a05f23b88b13bd094083da5b7efc94371) (unsigned int characterSize) const |
|  | Get the line spacing. |
|  | |
| const [Texture](http://docs.google.com/classsf_1_1Texture.htm) & | [getTexture](http://docs.google.com/classsf_1_1Font.htm#a887368a4e6a3dfa32dea89d2af315951) (unsigned int characterSize) const |
|  | Retrieve the texture containing the loaded glyphs of a certain size. |
|  | |
| [Font](http://docs.google.com/classsf_1_1Font.htm) & | [operator=](http://docs.google.com/classsf_1_1Font.htm#a232515549846e3172a514d0b47918399) (const [Font](http://docs.google.com/classsf_1_1Font.htm) &right) |
|  | Overload of assignment operator. |
|  | |

## Detailed Description

Class for loading and manipulating character fonts.

Fonts can be loaded from a file, from memory or from a custom stream, and supports the most common types of fonts.

See the loadFromFile function for the complete list of supported formats.

Once it is loaded, a [sf::Font](http://docs.google.com/classsf_1_1Font.htm) instance provides three types of information about the font:

* Global metrics, such as the line spacing
* Per-glyph metrics, such as bounding box or kerning
* Pixel representation of glyphs

Fonts alone are not very useful: they hold the font data but cannot make anything useful of it. To do so you need to use the [sf::Text](http://docs.google.com/classsf_1_1Text.htm) class, which is able to properly output text with several options such as character size, style, color, position, rotation, etc. This separation allows more flexibility and better performances: indeed a [sf::Font](http://docs.google.com/classsf_1_1Font.htm) is a heavy resource, and any operation on it is slow (often too slow for real-time applications). On the other side, a [sf::Text](http://docs.google.com/classsf_1_1Text.htm) is a lightweight object which can combine the glyphs data and metrics of a [sf::Font](http://docs.google.com/classsf_1_1Font.htm) to display any text on a render target. Note that it is also possible to bind several [sf::Text](http://docs.google.com/classsf_1_1Text.htm) instances to the same [sf::Font](http://docs.google.com/classsf_1_1Font.htm).

It is important to note that the [sf::Text](http://docs.google.com/classsf_1_1Text.htm) instance doesn't copy the font that it uses, it only keeps a reference to it. Thus, a [sf::Font](http://docs.google.com/classsf_1_1Font.htm) must not be destructed while it is used by a [sf::Text](http://docs.google.com/classsf_1_1Text.htm) (i.e. never write a function that uses a local [sf::Font](http://docs.google.com/classsf_1_1Font.htm) instance for creating a text).

Usage example:

// Declare a new font

[sf::Font](http://docs.google.com/classsf_1_1Font.htm) font;

// Load it from a file

if (!font.[loadFromFile](http://docs.google.com/classsf_1_1Font.htm#ab020052ef4e01f6c749a85571c0f3fd1)("arial.ttf"))

{

// error...

}

// Create a text which uses our font

[sf::Text](http://docs.google.com/classsf_1_1Text.htm) text1;

text1.[setFont](http://docs.google.com/classsf_1_1Text.htm#a2927805d1ae92d57f15034ea34756b81)(font);

text1.[setCharacterSize](http://docs.google.com/classsf_1_1Text.htm#ae96f835fc1bff858f8a23c5b01eaaf7e)(30);

text1.[setStyle](http://docs.google.com/classsf_1_1Text.htm#ad791702bc2d1b6590a1719aa60635edf)([sf::Text::Regular](http://docs.google.com/classsf_1_1Text.htm#aa8add4aef484c6e6b20faff07452bd82a2af9ae5e1cda126570f744448e0caa32));

// Create another text using the same font, but with different parameters

[sf::Text](http://docs.google.com/classsf_1_1Text.htm) text2;

text2.[setFont](http://docs.google.com/classsf_1_1Text.htm#a2927805d1ae92d57f15034ea34756b81)(font);

text2.[setCharacterSize](http://docs.google.com/classsf_1_1Text.htm#ae96f835fc1bff858f8a23c5b01eaaf7e)(50);

text1.[setStyle](http://docs.google.com/classsf_1_1Text.htm#ad791702bc2d1b6590a1719aa60635edf)([sf::Text::Italic](http://docs.google.com/classsf_1_1Text.htm#aa8add4aef484c6e6b20faff07452bd82aee249eb803848723c542c2062ebe69d8));

Apart from loading font files, and passing them to instances of [sf::Text](http://docs.google.com/classsf_1_1Text.htm), you should normally not have to deal directly with this class. However, it may be useful to access the font metrics or rasterized glyphs for advanced usage.

See Also[sf::Text](http://docs.google.com/classsf_1_1Text.htm)

Definition at line [50](http://docs.google.com/Font_8hpp_source.htm#l00050) of file [Font.hpp](http://docs.google.com/Font_8hpp_source.htm).

## Constructor & Destructor Documentation

| sf::Font::Font | ( |  | ) |  |
| --- | --- | --- | --- | --- |

Default constructor.

This constructor defines an empty font

| sf::Font::Font | ( | const [Font](http://docs.google.com/classsf_1_1Font.htm) & | *copy* | ) |  |
| --- | --- | --- | --- | --- | --- |

Copy constructor.

Parameters

| copy | Instance to copy |
| --- | --- |

| sf::Font::~Font | ( |  | ) |  |
| --- | --- | --- | --- | --- |

Destructor.

Cleans up all the internal resources used by the font

## Member Function Documentation

| const [Glyph](http://docs.google.com/classsf_1_1Glyph.htm)& sf::Font::getGlyph | ( | Uint32 | *codePoint*, |
| --- | --- | --- | --- |
|  |  | unsigned int | *characterSize*, |
|  |  | bool | *bold* |
|  | ) |  | const |

Retrieve a glyph of the font.

Parameters

| codePoint | Unicode code point of the character to get |
| --- | --- |
| characterSize | Reference character size |
| bold | Retrieve the bold version or the regular one? |

ReturnsThe glyph corresponding to *codePoint* and *characterSize*

| int sf::Font::getKerning | ( | Uint32 | *first*, |
| --- | --- | --- | --- |
|  |  | Uint32 | *second*, |
|  |  | unsigned int | *characterSize* |
|  | ) |  | const |

Get the kerning offset of two glyphs.

The kerning is an extra offset (negative) to apply between two glyphs when rendering them, to make the pair look more "natural". For example, the pair "AV" have a special kerning to make them closer than other characters. Most of the glyphs pairs have a kerning offset of zero, though.

Parameters

| first | Unicode code point of the first character |
| --- | --- |
| second | Unicode code point of the second character |
| characterSize | Reference character size |

ReturnsKerning value for *first* and *second*, in pixels

| int sf::Font::getLineSpacing | ( | unsigned int | *characterSize* | ) | const |
| --- | --- | --- | --- | --- | --- |

Get the line spacing.

Line spacing is the vertical offset to apply between two consecutive lines of text.

Parameters

| characterSize | Reference character size |
| --- | --- |

ReturnsLine spacing, in pixels

| const [Texture](http://docs.google.com/classsf_1_1Texture.htm)& sf::Font::getTexture | ( | unsigned int | *characterSize* | ) | const |
| --- | --- | --- | --- | --- | --- |

Retrieve the texture containing the loaded glyphs of a certain size.

The contents of the returned texture changes as more glyphs are requested, thus it is not very relevant. It is mainly used internally by [sf::Text](http://docs.google.com/classsf_1_1Text.htm).

Parameters

| characterSize | Reference character size |
| --- | --- |

Returns[Texture](http://docs.google.com/classsf_1_1Texture.htm) containing the glyphs of the requested size

| bool sf::Font::loadFromFile | ( | const std::string & | *filename* | ) |  |
| --- | --- | --- | --- | --- | --- |

Load the font from a file.

The supported font formats are: TrueType, Type 1, CFF, OpenType, SFNT, X11 PCF, Windows FNT, BDF, PFR and Type 42. Note that this function know nothing about the standard fonts installed on the user's system, thus you can't load them directly.

Parameters

| filename | Path of the font file to load |
| --- | --- |

ReturnsTrue if loading succeeded, false if it failed See Also[loadFromMemory](http://docs.google.com/classsf_1_1Font.htm#abf2f8d6de31eb4e1db02e061c323e346), [loadFromStream](http://docs.google.com/classsf_1_1Font.htm#abc3f37a354ce8b9a21f8eb93bd9fdafb)

| bool sf::Font::loadFromMemory | ( | const void \* | *data*, |
| --- | --- | --- | --- |
|  |  | std::size\_t | *sizeInBytes* |
|  | ) |  |  |

Load the font from a file in memory.

The supported font formats are: TrueType, Type 1, CFF, OpenType, SFNT, X11 PCF, Windows FNT, BDF, PFR and Type 42. Warning: SFML cannot preload all the font data in this function, so the buffer pointed by *data* has to remain valid as long as the font is used.

Parameters

| data | Pointer to the file data in memory |
| --- | --- |
| sizeInBytes | Size of the data to load, in bytes |

ReturnsTrue if loading succeeded, false if it failed See Also[loadFromFile](http://docs.google.com/classsf_1_1Font.htm#ab020052ef4e01f6c749a85571c0f3fd1), [loadFromStream](http://docs.google.com/classsf_1_1Font.htm#abc3f37a354ce8b9a21f8eb93bd9fdafb)

| bool sf::Font::loadFromStream | ( | [InputStream](http://docs.google.com/classsf_1_1InputStream.htm) & | *stream* | ) |  |
| --- | --- | --- | --- | --- | --- |

Load the font from a custom stream.

The supported font formats are: TrueType, Type 1, CFF, OpenType, SFNT, X11 PCF, Windows FNT, BDF, PFR and Type 42. Warning: SFML cannot preload all the font data in this function, so the contents of *stream* have to remain valid as long as the font is used.

Parameters

| stream | Source stream to read from |
| --- | --- |

ReturnsTrue if loading succeeded, false if it failed See Also[loadFromFile](http://docs.google.com/classsf_1_1Font.htm#ab020052ef4e01f6c749a85571c0f3fd1), [loadFromMemory](http://docs.google.com/classsf_1_1Font.htm#abf2f8d6de31eb4e1db02e061c323e346)

| [Font](http://docs.google.com/classsf_1_1Font.htm)& sf::Font::operator= | ( | const [Font](http://docs.google.com/classsf_1_1Font.htm) & | *right* | ) |  |
| --- | --- | --- | --- | --- | --- |

Overload of assignment operator.

Parameters

| right | Instance to assign |
| --- | --- |

ReturnsReference to self

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

* [Font.hpp](http://docs.google.com/Font_8hpp_source.htm)

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