**Pygame 官方文档 - pygame.freetype**

pygame.freetype

增强的pygame模块，用于加载和呈现计算机字体  
[pygame.freetype.get\_error](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#get_error%28%29) - 返回最新的FreeType错误  
[pygame.freetype.get\_version](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#get_version%28%29) - 返回FreeType版本  
[pygame.freetype.init](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#init%28%29) - 初始化基础FreeType库。  
[pygame.freetype.quit](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#quit%28%29) - 关闭底层的FreeType库。  
[pygame.freetype.was\_init](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#was_init%28%29) - 返回是否初始化FreeType库。  
[pygame.freetype.get\_cache\_size](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#get_cache_size%28%29) - 返回字形大小  
[pygame.freetype.get\_default\_resolution](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#get_default_resolution%28%29) - 以每英寸点数返回默认像素大小  
[pygame.freetype.set\_default\_resolution](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#set_default_resolution%28%29) - 为模块设置默认像素大小，以每英寸点数为单位  
[pygame.freetype.SysFont](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#SysFont%28%29) - 从系统字体创建一个Font对象  
[pygame.freetype.get\_default\_font](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#get_default_font%28%29) - 获取默认字体的文件名  
[pygame.freetype.Font](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font) - 从支持的字体文件创建一个新的Font实例。

        pygame.freetype模块是pygame.font模块的替代品，用于加载和呈现字体。它具有原始的所有功能，以及许多新功能。然而，它绝对不依赖于SDL\_ttf库。它直接在FreeType 2库上实现。 pygame.freetype模块本身不与pygame.font模块向后兼容。相反，使用pygame.ftfont模块作为pygame.font模块的替代品。  
        FreeType支持的所有字体文件格式都可以由pygame.freetype呈现，即TTF，Type1，CFF，OpenType，SFNT，PCF，FNT，BDF，PFR和Type42字体。所有具有UTF-32码点的字形都是可访问的（参见Font.ucs4）。  
        大多数字体工作都是使用Font实例完成的。模块本身只有初始化和创建Font对象的程序。您可以使用SysFont()函数从系统加载字体。  
        可以使用位图字体作为额外支持。可以列出可用的位图大小（请参阅Font.get\_sizes()）。对于仅位图字体Font可以为您设置大小（请参阅Font.size属性）。  
现在，未定义的字符代码将替换为.notdef（not defined未定义）字符。如何处理未定义的代码可能在将来的版本中可配置。  
        Pygame带有内置的默认字体。通过将None作为字体名称传递给Font构造函数，可以始终访问它。  
        pygame.freetype.Font可用的额外渲染功能直接表面渲染（参见Font.render\_to()），字符字距调整（参见Font.kerning），垂直布局（参见Font.vertical），旋转渲染文本（参见Font.rotation）和strong样式（参见Font.strong）。某些属性是可配置的，例如strong  
样式强度（请参阅Font.strength）和下划线定位（请参阅Font.underline\_adjustment）。文本可以通过文本框的右上角或文本基线定位（请参阅Font.origin）。最后，可以单独调整字体的垂直和水平尺寸（请参阅Font.size）。 pygame.examples.freetype\_misc示例显示了这些正在使用的功能。  
pygame包在加载时不会自动导入freetype。必须明确导入此模块才能使用。

import pygame

import pygame.freetype

freetype模块是pygame 1.9.2中的新增功能

pygame.freetype.get\_error() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#title)

返回最新的FreeType错误  
get\_error() -> str  
返回FreeType2库中发生的最后一个错误的描述，如果没有错误，则返回None。  
[搜索pygame.freetype.get\_error的示例](https://github.com/search?q=pygame.freetype.get_error&type=Code)

pygame.freetype.get\_version() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#title)

返回FreeType版本  
get\_version() -> (int, int, int)  
返回此模块使用的用于FreeType库的版本。  
请注意，freetype模块依赖于FreeType 2库。它不会使用原始的FreeType 1.0进行编译。因此，元组的第一个元素将始终为“2”。  
[搜索pygame.freetype.get\_version的示例](https://github.com/search?q=pygame.freetype.get_version&type=Code)

pygame.freetype.init() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#title)

初始化基础FreeType库。  
init(cache\_size=64, resolution=72)  
此函数初始化基础FreeType库，必须在尝试使用freetype模块的任何函数之前调用它。  
但是，如果已导入freetype模块，则pygame.init()初始化所有导入的pygame模块将自动调用此函数。 多次调用此函数是安全的。  
（可选）您可以为Glyph缓存指定默认的cache\_size：模块在任何给定时间缓存的最大字形数。 超出小的值将自动调整性能。 还可以给出默认像素分辨率（以每英寸点数为单位）来调整字体缩放。  
[搜索pygame.freetype.init的示例](https://github.com/search?q=pygame.freetype.init&type=Code)

pygame.freetype.quit() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#title)

关闭底层的FreeType库。  
qiut()  
此函数关闭freetype模块。 调用此函数后，您不应调用与freetype模块相关的任何类，方法或函数，因为它们可能会失败或可能产生不可预测的结果。 即使模块尚未初始化，也可以安全地调用此函数。  
[搜索pygame.freetype.quit的示例](https://github.com/search?q=pygame.freetype.quit&type=Code)

pygame.freetype.was\_init() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#title)

返回是否初始化FreeType库。  
was\_init() -> bool  
返回是否初始化FreeType库。  
[搜索pygame.freetype.was\_init的示例](https://github.com/search?q=pygame.freetype.was_init&type=Code)

pygame.freetype.get\_cache\_size() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#title)

返回字形大小  
get\_cache\_size() -> long  
见pygame.freetype.init()  
[搜索pygame.freetype.get\_cache\_size的示例](https://github.com/search?q=pygame.freetype.get_cache_size&type=Code)

pygame.freetype.get\_default\_resolution() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#title)

以每英寸点数返回默认像素大小  
set\_default\_resolution([resolution])  
设置模块的默认像素大小（以每英寸点数为单位）。如果省略可选参数或为零，则分辨率将重置为72 DPI。  
[搜索pygame.freetype.get\_default\_resolution的示例](https://github.com/search?q=pygame.freetype.get_default_resolution&type=Code)

pygame.freetype.set\_default\_resolution() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#title)

为模块设置默认像素大小，以每英寸点数为单位  
SysFont(name, size, bold=False, italic=False) -> Font  
返回从系统字体加载的新Font对象。 该字体将匹配请求的bold粗体和italic斜体标志。 如果找不到合适的系统字体，则返回默认的Pygame。 字体名称可以是逗号分隔的要搜索的字体名称列表。  
[搜索pygame.freetype.set\_default\_resolution的示例](https://github.com/search?q=pygame.freetype.set_default_resolution&type=Code)

pygame.freetype.SysFont() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#title)

从系统字体创建一个Font对象  
SysFont(name, size, bold=False, italic=False) -> Font  
返回从系统字体加载的新Font对象。 该字体将匹配请求的bold粗体和italic斜体标志。 如果找不到合适的系统字体，则返回默认的Pygame。 字体名称可以是逗号分隔的要搜索的字体名称列表。  
[搜索pygame.freetype.SysFont的示例](https://github.com/search?q=pygame.freetype.SysFont&type=Code)

pygame.freetype.get\_default\_font() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#title)

获取默认字体的文件名  
get\_default\_font() -> string  
返回默认pygame字体的文件名。 这不是文件的完整路径。 该文件通常与字体模块位于同一目录中，但也可以捆绑在单独的存档中。  
[搜索pygame.freetype.get\_default\_font的示例](https://github.com/search?q=pygame.freetype.get_default_font&type=Code)

pygame.freetype.Font [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#title)

从支持的字体文件创建一个新的Font实例。  
Font(file, size=0, font\_index=0, resolution=0, ucs4=False) -> Font  
[pygame.freetype.Font.name](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#name) - 正确的字体名称。  
[pygame.freetype.Font.path](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#path) - 字体文件路径  
[pygame.freetype.Font.size](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#size) - 渲染中使用的默认点大小  
[pygame.freetype.Font.get\_rect](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#get_rect) - 返回渲染文本的大小和偏移量  
[pygame.freetype.Font.get\_metrics](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#get_metrics) - 返回给定文本的字形指标  
[pygame.freetype.Font.height](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#height) - 字体单位中未缩放的字体高度  
[pygame.freetype.Font.ascender](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#ascender) - 字体单位中字体的无缩放上升  
[pygame.freetype.Font.descender](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#descender) - 字体单位字体的未缩放下降  
[pygame.freetype.Font.get\_sized\_ascender](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#get_sized_ascender%28%29) - 字体的缩放上升（以像素为单位）  
[pygame.freetype.Font.get\_sized\_descender](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#get_sized_descender%28%29) - 字体的缩放下降（以像素为单位）  
[pygame.freetype.Font.get\_sized\_height](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#get_sized_height%28%29) - 字体的缩放高度（以像素为单位）  
[pygame.freetype.Font.get\_sized\_glyph\_height](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#get_sized_glyph_height%28%29) - 字体的缩放边界框高度（以像素为单位）  
[pygame.freetype.Font.get\_sizes](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#get_sizes%28%29) - 返回嵌入式位图的可用大小  
[pygame.freetype.Font.render](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#render%28%29) - 将渲染文本作为surface返回  
[pygame.freetype.Font.render\_to](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#render_to%28%29) - 将文本渲染到现有surface上  
[pygame.freetype.Font.render\_raw](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#render_raw%28%29) - 将呈现的文本作为字节字符串返回  
[pygame.freetype.Font.render\_raw\_to](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#render_raw_to%28%29) - 将文本渲染为一个int数组  
[pygame.freetype.Font.style](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#style) - 字体的样式标志  
[pygame.freetype.Font.underline](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#underline) - 字体下划线样式标志的状态  
[pygame.freetype.Font.strong](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#strong) - 字体的强样式标志的状态  
[pygame.freetype.Font.oblique](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#oblique) - 字体倾斜样式标志的状态  
[pygame.freetype.Font.wide](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#wide) - 字体的宽样式标志的状态  
[pygame.freetype.Font.strength](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#strength) - 与强字体或宽字体样式相关联的强度  
[pygame.freetype.Font.underline\_adjustment](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#underline_adjustment) - 下划线位置的调整因子  
[pygame.freetype.Font.fixed\_width](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#fixed_width) - 获取字体是否为固定宽度  
[pygame.freetype.Font.fixed\_sizes](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#fixed_sizes) - 字体的可用位图大小数  
[pygame.freetype.Font.scalable](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#scalable) - 获取字体是否可伸缩  
[pygame.freetype.Font.use\_bitmap\_strikes](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#use_bitmap_strikes) - 允许在大纲字体文件中使用嵌入的位图  
[pygame.freetype.Font.antialiased](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#antialiased) - 字体抗锯齿模式  
[pygame.freetype.Font.kerning](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#kerning) - 字符字距调整模式  
[pygame.freetype.Font.vertical](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#vertical) - 字体垂直模式  
[pygame.freetype.Font.rotation](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#rotation) - 以逆时针方向旋转的文本  
[pygame.freetype.Font.fgcolor](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#fgcolor) - 默认前景色  
[pygame.freetype.Font.origin](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#origin) - 字体渲染到文本原点模式  
[pygame.freetype.Font.pad](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#pad) - 填充边界模式  
[pygame.freetype.Font.ucs4](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#ucs4) - 启用UCS-4模式  
[pygame.freetype.Font.resolution](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#resolution) - 以每英寸点数为单位的像素分辨率

参数file可以是表示字体文件名的字符串，包含字体的类文件对象，也可以是None;如果为None，则使用默认的Pygame字体。

（可选）可以指定size参数来设置点的默认大小，这决定了渲染字符的大小。大小也可以显式传递给每个方法调用。由于缓存系统的工作方式，在构造函数上指定默认大小并不意味着在每次函数调用上手动传递大小都会带来性能提升。如果字体是位图并且没有给出大小，则默认大小设置为字体的第一个可用大小。

如果字体文件有多个字体，则可以使用index参数选择要加载的字体。针对超出范围的字体索引值引发异常。

可选的resolution参数设置像素大小（以每英寸点数为单位），用于缩放此Font实例的字形。如果为0，则使用由init()设置的默认模块值。只能通过重新初始化Font实例来更改Font对象的分辨率。

可选的ucs4参数（整数）设置默认文本转换模式：0（False）识别UTF-16代理项对，任何其他值（True），将Unicode文本视为UCS-4，没有代理项对。见Font.ucs4。

pygame.freetype.Font.name [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

正确的字体名称。  
name -> string  
只读。 返回字体的实际（long）名称，记录在字体文件中。  
[搜索pygame.freetype.Font.name的示例](https://github.com/search?q=pygame.freetype.Font.name&type=Code)

pygame.freetype.Font.path [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

字体文件路径  
path -> unicode  
只读。 返回加载的字体文件的路径  
[搜索pygame.freetype.Font.path的示例](https://github.com/search?q=pygame.freetype.Font.path&type=Code)

pygame.freetype.Font.size [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

渲染中使用的默认点大小  
size -> float  
size -> (float, float)  
获取或设置文本度量和呈现的默认大小。它可以是单点大小，以Python int或float形式给出，或者是字体ppem（宽度，高度）元组。大小值是非负的。零大小或宽度表示未定义的大小。在这种情况下，大小必须作为方法参数给出，或者引发异常。零宽度但非零高度是ValueError。  
对于可缩放字体，单个数字值相当于宽度相等的元组。字体可以垂直拉伸，高度设置大于宽度，或水平宽度设置大于高度。对于get\_sizes()列出的嵌入式位图，请使用标称宽度和高度来选择可用大小。  
字体大小因不可伸缩的字体而异.。在方法调用期间，它必须与方法get\_sizes()返回的可用大小之一匹配。如果没有，则引发异常。如果大小是单个数字，则首先将大小与点大小值进行匹配。如果不匹配，则选择具有相同标称宽度和高度的可用尺寸。  
[搜索pygame.freetype.Font.size的示例](https://github.com/search?q=pygame.freetype.Font.size&type=Code)

pygame.freetype.Font.get\_rect [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

返回渲染文本的大小和偏移量  
get\_rect(text, style=STYLE\_DEFAULT, rotation=0, size=0) -> rect  
使用点，样式和旋转中的可选大小获取文本的最终尺寸和原点（以像素为单位）。 对于其他相关的渲染属性，以及未给出的任何可选参数，将使用为Font实例设置的默认值。  
返回一个Rect实例，其中包含文本边界框的宽度和高度以及文本原点的位置。 原点可用于对齐单独渲染的文本片段。 它给出了文本开头的基线位置和方位。 有关示例，请参阅render\_to()方法。  
如果text是char（byte）字符串，则假定其编码为LATIN1。  
（可选）文本可以是None，它将返回传递给先前get\_rect()，render()，render\_to()，render\_raw()或render\_raw\_to()调用的文本的边界矩形。 有关更多详细信息，请参阅render\_to()。  
[搜索pygame.freetype.Font.get\_rect的示例](https://github.com/search?q=pygame.freetype.Font.get_rect&type=Code)

pygame.freetype.Font.get\_metrics [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

返回给定文本的字形指标  
get\_metrics(text, size=0) -> [(…), …]  
返回文本中每个字符的字形度量标准。  
字形度量标准作为元组列表返回。 每个元组都提供单个字符字形的度量。 字形指标是：

（min\_x，max\_x，min\_y，max\_y，horizontal\_advance\_x，horizontal\_advance\_y）

边界框min\_x，max\_y，min\_y和max\_y值作为int类型的网格拟合像素坐标返回。 高级值是浮点值。  
计算是使用字体的默认大小（以点为单位）完成的。也可以使用“size”参数指定另一个点大小。  
针对当前旋转，强度和倾斜设置调整度量标准。  
如果text是char（byte）字符串，则假定其编码为LATIN1。  
[搜索pygame.freetype.Font.get\_metrics的示例](https://github.com/search?q=pygame.freetype.Font.get_metrics&type=Code)

pygame.freetype.Font.height [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

字体单位中未缩放的字体高度  
height -> int  
只读。 获取字体的高度。 这是字体中所有字形的平均值。  
[搜索pygame.freetype.Font.height的示例](https://github.com/search?q=pygame.freetype.Font.height&type=Code)

pygame.freetype.Font.ascender [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

字体单位中字体的无缩放上升  
ascender -> int  
只读。 将单位数从字体的基线返回到边界框的顶部。  
[搜索pygame.freetype.Font.ascender的示例](https://github.com/search?q=pygame.freetype.Font.ascender&type=Code)

pygame.freetype.Font.descender [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

字体单位字体的未缩放下降  
descender -> int  
[搜索pygame.freetype.Font.descender的示例](https://github.com/search?q=pygame.freetype.Font.descender&type=Code)

pygame.freetype.Font.get\_sized\_ascender() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

字体的缩放上升（以像素为单位）  
get\_sized\_ascender(=0) -> int  
只读。 以字体单位返回字体下降的高度。 下降是从字体的基线到边界框底部的单位数。  
[搜索pygame.freetype.Font.get\_sized\_ascender的示例](https://github.com/search?q=pygame.freetype.Font.get_sized_ascender&type=Code)

pygame.freetype.Font.get\_sized\_descender() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

字体的缩放下降（以像素为单位）  
get\_sized\_descender(=0) -> int  
将字体基线的像素数返回到边界框的顶部。 它没有针对强度或旋转进行调整。  
[搜索pygame.freetype.Font.get\_sized\_descender的示例](https://github.com/search?q=pygame.freetype.Font.get_sized_descender&type=Code)

pygame.freetype.Font.get\_sized\_height() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

字体的缩放高度（以像素为单位）  
get\_sized\_height(=0) -> int  
返回字体的高度。 这是字体中所有字形的平均值。 它没有针对强度或旋转进行调整。  
[搜索pygame.freetype.Font.get\_sized\_height的示例](https://github.com/search?q=pygame.freetype.Font.get_sized_height&type=Code)

pygame.freetype.Font.get\_sized\_glyph\_height() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

字体的缩放边界框高度（以像素为单位）  
get\_sized\_glyph\_height(=0) -> int  
返回字体的字形边界框高度（以像素为单位）。 这是字体中所有字形的平均值。 它没有针对强度或旋转进行调整。  
[搜索pygame.freetype.Font.get\_sized\_glyph\_height的示例](https://github.com/search?q=pygame.freetype.Font.get_sized_glyph_height&type=Code)

pygame.freetype.Font.get\_sizes() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

返回嵌入式位图的可用大小  
get\_sizes() -> [(int, int, int, float, float), …]  
get\_sizes() -> []  
返回元组记录列表，每个点大小支持一个记录。 每个元组包含点大小，像素高度，像素宽度，分数像素中的水平ppem（标称宽度）和小数像素中的垂直ppem（标称高度）。  
[搜索pygame.freetype.Font.get\_sizes的示例](https://github.com/search?q=pygame.freetype.Font.get_sizes&type=Code)

pygame.freetype.Font.render() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

将渲染文本作为surface返回  
render(text, fgcolor=None, bgcolor=None, style=STYLE\_DEFAULT, rotation=0, size=0) -> (Surface, Rect)  
返回一个新的Surface，文本以’fgcolor’给出的颜色呈现给它。如果没有给出前景色，则使用默认前景色fgcolor。如果给出bgcolor，Surface将用这种颜色填充。如果没有给出背景颜色，Surface背景是透明的，0 alpha。通常，返回的Surface具有32位像素大小。但是，如果bgcolor为None且禁用了抗锯齿功能，则会返回单色8位颜色键Surface，并为背景颜色设置colorkey。  
返回值是一个元组：新Surface和边界矩形给出渲染文本的大小和原点。  
如果为文本传递空字符串，则返回的Rect为零宽度和字体高度。  
可选的fgcolor，样式，旋转和大小参数会覆盖为Font实例设置的默认值。  
如果text是char（byte）字符串，则假定其编码为LATIN1。  
可选地，文本可以是None，这将呈现传递给先前的get\_rect()，render()，render\_to()，render\_raw()或render\_raw\_to()调用的文本。有关详细信息，请参阅render\_to()。  
[搜索pygame.freetype.Font.render的示例](https://github.com/search?q=pygame.freetype.Font.render&type=Code)

pygame.freetype.Font.render\_to() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

将文本渲染到现有surface上  
render\_to(surf, dest, text, fgcolor=None, bgcolor=None, style=STYLE\_DEFAULT, rotation=0, size=0) -> Rect  
将字符串文本呈现给pygame.Surface对象，位于dest，（x，y）表面坐标对。如果x或y不是整数，则尽可能将其转换为整数。接受前两个项是x和y位置元素的任何序列，包括Rect实例。与render()一样，可以选择fgcolor，style，rotation和size参数。  
如果给出背景颜色bgcolor，则首先用该颜色填充文本边界框。接下来是文本。背景填充和文本渲染都涉及完整的alpha blits。也就是说，前景，背景和目标目标表面的alpha值都会影响blit。  
返回值是一个矩形，用于表示渲染文本在surface内的大小和位置。  
如果为文本传递空字符串，则返回的Rect为零宽度和字体高度。 rect将测试False。  
可选地，可以将文本设置为None，这将重新呈现传递给先前render\_to()，get\_rect()，render()，render\_raw()或render\_raw\_to()调用的文本。首先，此功能有助于将render\_to()与get\_rect()结合使用。一个例子：

def word\_wrap(surf, text, font, color=(0, 0, 0)):

font.origin = True

words = text.split(' ')

width, height = surf.get\_size()

line\_spacing = font.get\_sized\_height() + 2

x, y = 0, line\_spacing

space = font.get\_rect(' ')

for word in words:

bounds = font.get\_rect(word)

if x + bounds.width + bounds.x >= width:

x, y = 0, y + line\_spacing

if x + bounds.width + bounds.x >= width:

raise ValueError("word too wide for the surface")

if y + bounds.height - bounds.y >= height:

raise ValueError("text to long for the surface")

font.render\_to(surf, (x, y), None, color)

x += bounds.width + space.width

return x, y

当使用相同的字体属性（size，style，strength，wide，antialiased，vertical，rotation，kerning和use\_bitmap\_strikes）调用render\_to()作为get\_rect()时，render\_to()将使用get\_rec()计算的布局。 否则，如果使用文本字符串调用，则render\_to()将重新计算布局，或者在get\_rect()调用之后，上述属性之一已更改。  
如果text是char（byte）字符串，则假定其编码为LATIN1。  
[搜索pygame.freetype.Font.render\_to的示例](https://github.com/search?q=pygame.freetype.Font.render_to&type=Code)

pygame.freetype.Font.render\_raw() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

将呈现的文本作为字节字符串返回  
render\_raw(text, style=STYLE\_DEFAULT, rotation=0, size=0, invert=False) -> (bytes, (int, int))  
与render()类似，但返回的像素为8位灰度值的字节字符串。 前景色为255，背景为0，可用作前景图案的alpha蒙版。  
[搜索pygame.freetype.Font.render\_raw的示例](https://github.com/search?q=pygame.freetype.Font.render_raw&type=Code)

pygame.freetype.Font.render\_raw\_to() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

将文本渲染为一个int数组  
render\_raw\_to(array, text, dest=None, style=STYLE\_DEFAULT, rotation=0, size=0, invert=False) -> (int, int)  
渲染到公开数组结构接口的数组对象。 数组必须是带有整数项的二维数。 默认目标值None，等同于position（0,0）。 请参阅render\_to（）。 与其他渲染方法一样，text可以为None，以呈现先前传递给另一个方法的文本字符串。  
[搜索pygame.freetype.Font.render\_raw\_to的示例](https://github.com/search?q=pygame.freetype.Font.render_raw_to&type=Code)

pygame.freetype.Font.style [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

字体的样式标志   
style -> int  
获取或设置Font的默认样式。 除非特别重写render或get\_rect()调用，否则此默认样式将用于所有文本呈现和大小计算。 样式值可以是以下一个或多个常量的按位或（bit-wise OR）：

STYLE\_NORMAL

STYLE\_UNDERLINE

STYLE\_OBLIQUE

STYLE\_STRONG

STYLE\_WIDE

STYLE\_DEFAULT

这些常量可以在FreeType常量模块中找到。 （可选）可以修改或获取默认样式，以访问各个样式属性（下划线，倾斜，强）。  
STYLE\_OBLIQUE和STYLE\_STRONG样式仅适用于可缩放字体。 尝试设置位图字体会引发AttributeError。 尝试设置为Font .\_\_ new \_\_()返回的非活动字体会引发RuntimeError。  
将STYLE\_DEFAULT分配给style属性会使属性保持不变，因为此属性定义了默认值。 style属性永远不会返回STYLE\_DEFAULT。  
[搜索pygame.freetype.Font.style的示例](https://github.com/search?q=pygame.freetype.Font.style&type=Code)

pygame.freetype.Font.underline [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

字体下划线样式标志的状态  
underline -> bool  
获取或设置绘制文本时字体是否加下划线。 此默认样式值将用于所有文本呈现和大小计算，除非通过’style’参数在render或get\_rect()调用中特别重写。  
[搜索pygame.freetype.Font.underline的示例](https://github.com/search?q=pygame.freetype.Font.underline&type=Code)

pygame.freetype.Font.strong [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

字体的强样式标志的状态  
strong -> bool  
获取或设置绘制文本时字体是否为粗体。 此默认样式值将用于所有文本呈现和大小计算，除非通过’style’参数在render或get\_rect()调用中特别重写。  
[搜索pygame.freetype.Font.strong的示例](https://github.com/search?q=pygame.freetype.Font.strong&type=Code)

pygame.freetype.Font.oblique [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

字体倾斜样式标志的状态  
oblique -> bool  
获取或设置字体是否呈现为倾斜。 此默认样式值将用于所有文本呈现和大小计算，除非通过style参数在render或get\_rect()调用中特别重写。  
倾斜样式仅支持可缩放（轮廓）字体。 尝试在位图字体上设置此样式将引发AttributeError。 如果字体对象处于非活动状态，如Font .\_\_ new \_\_()返回，则设置此属性会引发RuntimeError。  
[搜索pygame.freetype.Font.oblique的示例](https://github.com/search?q=pygame.freetype.Font.oblique&type=Code)

pygame.freetype.Font.wide [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

字体的宽样式标志的状态  
wide -> bool  
获取或设置在绘制文本时是否水平拉伸字体。 它产生类似于pygame.font的结果。从文件的粗体创建一个新的Font对象。 此样式不适用于旋转文本。  
[搜索pygame.freetype.Font.wide的示例](https://github.com/search?q=pygame.freetype.Font.wide&type=Code)

pygame.freetype.Font.strength [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

与强字体或宽字体样式相关联的强度  
strength -> float  
为strong强转换或宽转换扩大字体字形大小的量，作为未转换大小的一部分。 对于宽样式，仅增加水平尺寸。 对于strong强文本，水平和垂直尺寸都被放大。 宽度为0.08333（1/12）的强度相当于pygame.font.Font从文件粗体样式创建一个新的Font对象。 默认值为0.02778（1/36）。  
强度样式仅支持可缩放（大纲）字体。 尝试在位图字体上设置此属性将引发AttributeError。 如果字体对象处于非活动状态（由Font .\_\_ new \_\_（）返回，则对此属性的赋值会引发RuntimeError。  
[搜索pygame.freetype.Font.strength的示例](https://github.com/search?q=pygame.freetype.Font.strength&type=Code)

pygame.freetype.Font.underline\_adjustment [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

下划线位置的调整因子  
underline\_adjustment -> float  
获取或设置一个因子，当为正时，与字体的下划线偏移相乘以调整下划线位置。 负值将下划线转换为删除线或上划线。 它与ascender相乘。 可接受的值介于-2.0和2.0之间。 值0.5与Tango下划线非常匹配。 值为1.0模仿pygame.font.Font从文件下划线创建一个新的Font对象。  
[搜索pygame.freetype.Font.underline\_adjustment的示例](https://github.com/search?q=pygame.freetype.Font.underline_adjustment&type=Code)

pygame.freetype.Font.fixed\_width [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

获取字体是否为固定宽度  
fixed\_width -> bool  
只读。 如果字体包含固定宽度字符，则返回True（例如Courier，Bitstream Vera Sans Mono，Andale Mono）。  
[搜索pygame.freetype.Font.fixed\_width的示例](https://github.com/search?q=pygame.freetype.Font.fixed_width&type=Code)

pygame.freetype.Font.fixed\_sizes [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

字体的可用位图大小数  
fixed\_sizes -> int  
只读。返回字体包含位图字符图像的点大小数。如果为零，则字体不是位图字体。可缩放字体可以包含预渲染的点大小作为删除。  
[搜索pygame.freetype.Font.fixed\_sizes的示例](https://github.com/search?q=pygame.freetype.Font.fixed_sizes&type=Code)

pygame.freetype.Font.scalable [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

获取字体是否可伸缩  
scalable -> bool  
只读。 如果字体包含轮廓字形，则返回True。 如果是这样，则点大小不限于可用的位图大小。  
[搜索pygame.freetype.Font.scalable的示例](https://github.com/search?q=pygame.freetype.Font.scalable&type=Code)

pygame.freetype.Font.use\_bitmap\_strikes [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

允许在大纲字体文件中使用嵌入的位图  
use\_bitmap\_strikes -> bool  
一些可缩放字体包括特定点大小的嵌入位图。此属性控制是否使用这些位图删除。设置为false将禁用加载任何位图删除。将其设置为True（默认值），以允许非旋转渲染的位图删除，除了“宽”或“下划线”之外没有其他样式。位图字体忽略此属性。  
另请参见固定的\_大小和获取\_大小（）。  
[搜索pygame.freetype.Font.use\_bitmap\_strikes的示例](https://github.com/search?q=pygame.freetype.Font.use_bitmap_strikes&type=Code)

pygame.freetype.Font.antialiased [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

字体抗锯齿模式  
antialiased -> bool  
获取或设置字体的消除锯齿模式。 对于所有字体，默认为True，使用完整的8位混合进行渲染。  
设置为False以进行单色渲染。 这应该提供小的速度增益并减少高速缓冲存储器的大小。  
[搜索pygame.freetype.Font.antialiased的示例](https://github.com/search?q=pygame.freetype.Font.antialiased&type=Code)

pygame.freetype.Font.kerning [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

字符字距调整模式  
kerning -> bool  
获取或设置字体的字距调整模式。 所有字体的默认值均为False，将在不进行字距调整的情况下呈现。  
设置为True以在字符对之间添加字距，如果字体支持，则在定位字形时。  
[搜索pygame.freetype.Font.kerning的示例](https://github.com/search?q=pygame.freetype.Font.kerning&type=Code)

pygame.freetype.Font.vertical [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

字体垂直模式  
vertical -> bool  
获取或设置字符是垂直布局而不是水平布局。 在渲染汉字或其他垂直脚本时可能很有用。  
设置为True以切换到垂直文本布局。 默认值为False，水平放置。  
请注意，Font类不会自动确定脚本方向。 必须明确选择垂直布局。  
另请注意，几种字体格式（尤其是基于位图的格式）不包含垂直绘制字形的必要指标，因此在这些情况下绘制将给出未指定的结果。  
[搜索pygame.freetype.Font.vertical的示例](https://github.com/search?q=pygame.freetype.Font.vertical&type=Code)

pygame.freetype.Font.rotation [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

以逆时针方向旋转的文本  
rotation -> int  
获取或设置渲染文本的基线角度。 角度表示为整数度。 默认角度为0，水平文本沿X轴呈现，垂直文本沿Y轴呈现。 正值将这些轴逆时针旋转很多度。 负角度对应于顺时针旋转。 旋转值标准化为0到359（含）范围内的值（例如390 - > 390 - 360 - > 30，-45 - > 360 + -45 - > 315,720 - > 720 - （2 \* 360） - > 0）。  
只能旋转可缩放（轮廓）字体。 尝试更改位图字体的旋转会引发AttributeError。 尝试更改Font .\_\_ new \_\_（）返回的非活动字体实例的旋转，会引发RuntimeError。  
[搜索pygame.freetype.Font.rotation的示例](https://github.com/search?q=pygame.freetype.Font.rotation&type=Code)

pygame.freetype.Font.fgcolor [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

默认前景色  
fgcolor -> Color  
获取或设置默认的字形呈现颜色。 它最初是不透明的黑色 - （0,0,0,255）。 适用于render()和render\_to()。  
[搜索pygame.freetype.Font.fgcolor的示例](https://github.com/search?q=pygame.freetype.Font.fgcolor&type=Code)

pygame.freetype.Font.origin [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

字体渲染到文本原点模式  
origin -> bool  
如果设置为True，则render\_to()和render\_raw\_to()将使dest位置成为文本原点的位置，而不是边界框的左上角。 有关详细信息，请参阅get\_rect()。  
[搜索pygame.freetype.Font.origin的示例](https://github.com/search?q=pygame.freetype.Font.origin&type=Code)

pygame.freetype.Font.pad [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

填充边界模式  
pad -> bool  
如果设置为True，则文本边界矩形将被膨胀以匹配font.Font。 否则，边界矩形对于文本来说足够大。  
[搜索pygame.freetype.Font.pad的示例](https://github.com/search?q=pygame.freetype.Font.pad&type=Code)

pygame.freetype.Font.ucs4 [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

启用UCS-4模式  
ucs4 -> bool  
获取或设置Unicode文本的解码。 默认情况下，freetype模块对Unicode文本执行UTF-16代理对解码。 这允许在0x10000和0x10FFFF之间的32位转义序列（‘Uxxxxxxxx’）表示在使用UCS-2 Unicode类型（例如，在Windows上）构建的Python解释器上的相应UTF-32代码点。 它还意味着UTF-16代理区域（0xD800到0xDFFF）内的字符值被视为代理项对的一部分。 格式错误的代理项对将引发UnicodeEncodeError。 设置ucs4 True关闭代理对解码，允许访问完整的UCS-4字符范围到使用四字节Unicode字符支持构建的Python解释器。  
[搜索pygame.freetype.Font.ucs4的示例](https://github.com/search?q=pygame.freetype.Font.ucs4&type=Code)

pygame.freetype.Font.resolution [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88326853#Font)

以每英寸点数为单位的像素分辨率  
resolution -> int  
只读。 获取用于缩放此Font实例的字体字形的像素大小。  
[搜索pygame.freetype.Font.resolution的示例](https://github.com/search?q=pygame.freetype.Font.resolution&type=Code)