

# music | Pygame中文文档

## pygame.mixer.music

Pygame 中控制音频流的模块。

### 函数

- `pygame.mixer.music.load()` —— 载入一个音乐文件用于播放
- `pygame.mixer.music.play()` —— 开始播放音乐流
- `pygame.mixer.music.rewind()` —— 重新开始播放音乐
- `pygame.mixer.music.stop()` —— 结束音乐播放
- `pygame.mixer.music.pause()` —— 暂停音乐播放
- `pygame.mixer.music.unpause()` —— 恢复音乐播放
- `pygame.mixer.music.fadeout()` —— 淡出的效果结束音乐播放
- `pygame.mixer.music.set_volume()` —— 设置音量
- `pygame.mixer.music.get_volume()` —— 获取音量
- `pygame.mixer.music.get_busy()` —— 检查是否正在播放音乐
- `pygame.mixer.music.set_pos()` —— 设置播放的位置
- `pygame.mixer.music.get_pos()` —— 获取播放的位置
- `pygame.mixer.music.queue()` —— 将一个音乐文件放入队列中，并排在当前播放的音乐之后
- `pygame.mixer.music.set_endevent()` —— 当播放结束时发出一个事件
- `pygame.mixer.music.get_endevent()` —— 获取播放结束时发送的事件

Pygame 中播放音乐的模块和 `pygame.mixer` 模块是密切联系的。使用音乐模块去控制在调音器上的音乐播放。

音乐（music）播放和声音（sound）播放的不同之处在于音乐是流式的，并且绝对不会在一开始就把一个音乐文件全部载入。调音系统在工作刚开始时仅支持单音乐流。

注意：对于 MP3 格式的支持是受限制的。在一些系统上，一种不受支持的格式将会是系统崩溃，例如 Debian Linux。为了游戏的稳定性，建议使用 OGG 进行替代。

### 函数详解

#### `pygame.mixer.music.load()`

载入一个音乐文件用于播放。

*load(filename) -> None*

*load(object) -> None*

该函数将会载入一个音乐文件名或者文件对象，并且准备播放。如果已经有音乐流正在播放，该音乐流将被停止。另外，函数不会开始播放音乐。

### pygame.mixer.music.play()

开始播放音乐流。

*play(loops=0, start=0.0) -> None*

该函数用于播放已载入的音乐流。如果音乐已经开始播放，则将会重新开始播放。

loops 参数控制重复播放的次数，例如 play(5) 意味着被载入的音乐将会立即开始播放 1 次并且再重复 5 次，共 6 次。如果 loops = -1，则表示无限重复播放。

start 参数控制音乐从哪里开始播放。开始的位置取决于音乐的格式。MP3 和 OGG 使用时间表示播放位置（以秒为单位）。MOD使用模式顺序编号表示播放位置。如果音乐文件无法设置开始位置，则传递了start参数后会产生一个NotImplementedError错误。

### pygame.mixer.music.rewind()

重新开始播放音乐。

*rewind() -> None*

从文件开头开始重新播放音乐。

### pygame.mixer.music.stop()

结束音乐播放。

*stop() -> None*

如果音乐正在播放则立即结束播放。

### pygame.mixer.music.pause()

暂停音乐流的播放。

*pause() -> None*

通过调用 pygame.mixer.music.unpause() 函数继续播放音乐。

### pygame.mixer.music.unpause()

恢复音乐播放。

*unpause() -> None*

在播放暂停后使用该函数可以继续音乐流的播放。

### pygame.mixer.music.fadeout()

淡出的效果结束音乐播放。

*fadeout(time) -> None*

该函数将会在音乐淡出（也就是不在有声音放出）一段指定长度的时间（以毫秒为单位）后结束播放。

注意：该函数在调用后会一直处于阻塞状态，直到音乐已经淡出。

### `pygame.mixer.music.set_volume()`

设置音量。

*set\_volume(value) -> None*

设置音乐的播放音量。

value 参数值范围为 0.0~1.0。当新的音乐文件被载入，音量会被重置。

### `pygame.mixer.music.get_volume()`

获取音量。

*get\_volume() -> value*

返回正在播放的音乐的音量（此音量应该是调音器音量，注意与其他音量参数区分）。返回值范围为 0.0~1.0。

### `pygame.mixer.music.get_busy()`

检查是否正在播放音乐。

*get\_busy() -> bool*

如果有音乐流正在播放，此方法返回 True。否则返回 False。

### `pygame.mixer.music.set_pos()`

设置播放的位置。

*set\_pos(pos) -> None*

设置播放的起始位置。pos 参数是一个浮点数（或者一个可以转换为浮点数的数值），其值取决于音乐文件的格式：

- 对于 MOD 文件，它是模块中的整型模式号；
- 对于 OGG 文件，它是一个以音频开头为零点的绝对时间值（以秒为单位）；
- 对于 MP3 文件，它是以当前播放位置为零点的绝对时间值（以秒为单位）。

为了对一个 MP3 文件的进行绝对定位，建议首先调用 `rewind()` 函数（其他文件格式不受支持）。SDL\_mixer 更新的版本提供了更好的定位支持。如果一种特殊的格式不支持定位，将会产生一个 `SDL_Error` 错误。

该函数会调用 SDL\_mixer 内的 `Mix_SetMusicPosition()` 函数。

### `pygame.mixer.music.get_pos()`

获取播放的位置。

*get\_pos() -> time*

此函数会获得音乐的播放时长（以毫秒为单数的数值）。返回值仅代表已经音乐已经播放了多久，并不考虑任何起始位置偏移量。

### pygame.mixer.music.queue()

将一个音乐文件放入队列中，并排在当前播放的音乐之后。

*queue(filename) -> None*

此函数将会载入一个音乐文件并将其放入队列中。当前的音乐一旦播放完毕，正在排队的音乐文件就会开始播放。如果当前音乐被人为停止或者切换到其他音乐，则正在排队的音乐会被丢弃。

下面的示例意思是先播放 6 次 Bach 然后再播放 1 次 Mozart：

```
01. pygame.mixer.music.load('bach.ogg')
02. pygame.mixer.music.play(5) # Plays six times, not five!
03. pygame.mixer.music.queue('mozart.ogg')
```

### pygame.mixer.music.set\_endevent()

当播放结束时发出一个事件。

*set\_endevent() -> None*

*set\_endevent(type) -> None*

调用此函数会使 Pygame 在音乐结束播放后发出信号（通过事件队列）。

type 参数决定了什么样的事件将被放入事件队列中。

任何时候音乐结束，都会放入指定事件到队列中（不仅仅是第一次）。调用该函数并不带任何参数，表示停止投放事件到队列中。

### pygame.mixer.music.get\_endevent()

获取播放结束时发送的事件。

*get\_endevent() -> type*

返回音乐结束时被放入队列的事件类型。

如果没有指定 endevent 事件，此方法会返回 pygame.NOEVENT。