



博客 (<http://blog.csdn.net?ref=toolbar>) 学院 (<http://edu.csdn.net?ref=toolbar>)

下载 (<http://download.csdn.net?ref=toolbar>)

GitChat (<http://gitbook.cn/?ref=csdn>) 更多 ▼



1

pydub简单介绍



原创

2017年03月01日 11:10:11



标签：python (<http://so.csdn.net/so/search/s.do?q=python&t=blog>) /



音频 (<http://so.csdn.net/so/search/s.do?q=音频&t=blog>) /



数字信号处理 (<http://so.csdn.net/so/search/s.do?q=数字信号处理&t=blog>) /



第三方库 (<http://so.csdn.net/so/search/s.do?q=第三方库&t=blog>) /



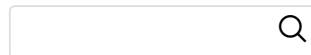
API文档 (<http://so.csdn.net/so/search/s.do?q=API文档&t=blog>)



2862

pydub是python的高级一个音频处理库，可以让你以一种不那么蠢的方法处理音频。（开发者自己是这么说的，不过个人感觉有时候仍然很蠢。。。。）

pydub传送门 (<https://github.com/jiaaro/pydub>)



weixin_3506...

(<http://my.csdn.net/postcode/123456789>)

(<http://my.csdn.net/postcode/123456789>)

Debatrix (<http://blog.csdn.net/debatrix>)

+ 关注



望京soho

码云

未开通

(<https://github.com>)

原创
8

粉丝
3

喜欢
1

他的最新文章

更多文章 (<http://blog.csdn.net/debatrix>)

ubuntu broadcom驱动无法正常工作 (<http://blog.csdn.net/Debatrix/article/details/59056269>)

Anaconda多环境多版本python配置指导 (<http://blog.csdn.net/Debatrix/article/details/56277761>)

Anaconda安装二三事 (<http://blog.csdn.net/Debatrix/article/details/56024008>)

Python开发简单爬虫学习笔记（2） (<http://blog.csdn.net/Debatrix/article/details/51103503>)



内容举报



返回顶部

安装

```
1 pip install pydub
```

注意：pydub直接支持wav格式音频读取，如果需要处理其它格式音频（如MP3，ogg等）需要安装ffmpeg。

ffmpeg的安装在各平台上略有不同，具体方法自行google

1

快速开始

打开WAV文件



```
1 from pydub import AudioSegment
2
3 song = AudioSegment.from_wav("never_gonna_give_you_up.wav")
```



...或mp3

```
1 song = AudioSegment.from_mp3("never_gonna_give_you_up.mp3")
```

...或ogg，flv或ffmpeg支持的任何其他东西

Python开发简单爬虫学习笔记
tp://blog.csdn.net/Debatrix/s/51103378)



日语常用语



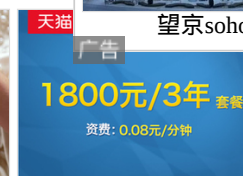
创意产品设计



望京soho



阿宝结婚照



申请400号码

他的热门文章

关于Matlab中括号用法的总结 (<http://blog.csdn.net/debatrix/article/details/50585624>)

7911

[转载]MATLAB读取音频文件并进行音频特征提取 (<http://blog.csdn.net/debatrix/article/details/50612362>)

7065



内容举报

Anaconda安装二三事 (<http://blog.csdn.net/debatrix/article/details/56024008>)

5223



返回顶部

pydub简单介绍 (<http://blog.csdn.net/debatrix/article/details/59058762>)

```

1  ogg_version = AudioSegment.from_ogg("never_gonna_give_you_up.ogg")
2  flv_version = AudioSegment.from_flv("never_gonna_give_you_up.flv")
3
4  mp4_version = AudioSegment.from_file("never_gonna_give_you_up.mp4", "mp4")
5  wma_version = AudioSegment.from_file("never_gonna_give_you_up.wma", "wma")
6  aac_version = AudioSegment.from_file("never_gonna_give_you_up.aiff", "aac")

```

👍
音频切片：

```

1  # pydub以毫秒为单位工作
2  ten_seconds = 10 * 1000
3
4  first_10_seconds = song[:ten_seconds]
5
6  last_5_seconds = song[-5000:]

```



放大开头音量，减小结尾音量



```

1  # 放大6dB音量
2  beginning = first_10_seconds + 6
3
4  # 衰减3dB音量
5  end = last_5_seconds - 3

```

连接音频（将一个文件添加到另一个文件的末尾）

```

1  without_the_middle = beginning + end

```

音频时长？

📖 2861

Anaconda多环境多版本py
(<http://blog.csdn.net/debatr/s/56277761>)

📖 816



日语常用语



移民美国的好处



望京soho



术前
咬肌肥大



WebRTC



内容举报





返回顶部

```
1 without_the_middle.duration_seconds == 15.0
```



注意：AudioSegments是不可变的

```
1 backwards = song.reverse()
```


 交叉淡化（开头和结束不修改）
1

```
 1 with_style = beginning.append ( end , crossfade = 1500 )
```

 重复

```
 1 # 重复剪辑两次  
 2 do_it_over = with_style * 2
```


 淡出（注意，你可以链接操作，因为一切返回一个AudioSegment）

```
 1 # 2秒的淡入，3秒的淡出  
2 awesome = do_it_over.fade_in ( 2000 ) .fade_out ( 3000 )
```


保存结果（可能需要ffmpeg）

```
1 awesome.export ( " mashup.mp3 " , format = " mp3 " )
```

保存结果（使用标签/元数据）



日语常用语



望京soho

广告

 内容举报

 返回顶部

```
1 awesome.export("mashup.mp3", format="mp3", tags={'artist': 'Various artists', 'album': 'Best of 2011', 'comments': 'This a
```

可以使用ffmpeg支持的任何语法将可选的bitrate参数传递给export。(这句想不到一个通顺的翻译了)

```
1 awesome.export ( " mashup.mp3 " , format = " mp3 " , bitrate = " 192k " )
```



ffmpeg支持的任何进一步的参数可以作为'参数'参数中的列表进行传递，其中第一个为开关，第二个为参数。注意，不对这些参数进行验证，并且您可能受到您的特定构建的ffmpeg / avlib支持的限制。



```
1 # 使用预设的mp3品质0 (相当于lame V0)
```

```
2 awesome.export ( " mashup.mp3 " , format = " mp3 " , parameters = [ " -q : a " , " 0 " ] )
```



```
3
```

```
4 # 混合下来两个通道并设置硬输出音量
```



```
5 awesome.export ( " mashup.mp3 " , format = " mp3 " , parameters = [ " -ac " , " 2 " , " -vol " , " 150 " ] )
```



API文档

AudioSegment 对象是不可变的，并且支持多个运算符。



内容举报

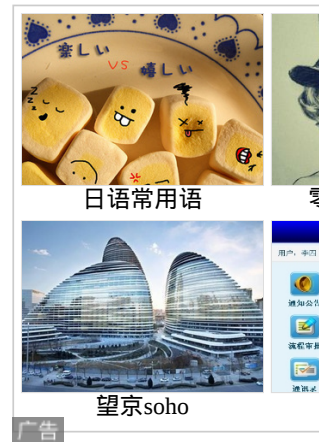


返回顶部

```
1 from pydub import AudioSegment
2 sound1 = AudioSegment.from_file("/path/to/sound.wav", format="wav")
3 sound2 = AudioSegment.from_file("/path/to/another_sound.wav", format="wav")
4
5 # sound1增大6dB,之后衰减3.5dB
6 louder = sound1 + 6
7 quieter = sound1 - 3.5
8
9 # 连接sound1与sound2
10 combined = sound1 + sound2
11
12 # sound1重复三次
13 repeated = sound1 * 3
14
15 # 持续时间
16 duration_in_milliseconds = len(sound1)
17
18 # sound1的开头5秒
19 beginning = sound1[:5000]
20
21 # sound1的结尾5秒
22 end = sound1[-5000:]
23
24 # 高级用法：原始数据处理
25 sound = AudioSegment(
26     # 音频原始数据(bytes)
27     data=b'...',
28
29     # 2 byte (16 bit) 采样
30     sample_width=2,
31
32     # 44.1 kHz 采样率 frame_rate=44100,
```



1



内容举报



返回顶部

```

33
34     # 声道
35     channels=2
36 )

```

AudioSegment以任何方式组合多个对象的任何操作将首先确保它们具有相同数量的通道，帧速率，采样率，比特深度等。当这些东西不匹配时，较低质量的声音被修改以匹配更高质量的声音，使得质量不会丢失：单声道转换为立体声，位深度和帧速率/采样率根据需要增加。如果您不想这种行为，您可以使用适当的AudioSegment方法显式减少通道数，位数等。

AudioSegment (...) .from_file ()

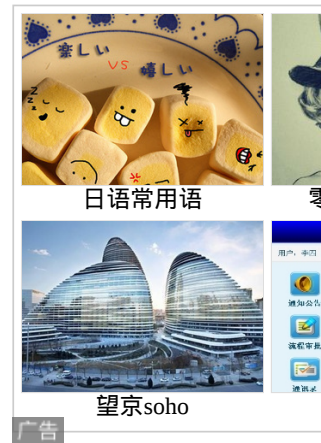
将音频文件作为AudioSegment实例打开并返回。还有一些为了方便提供的包装器，但您可能应该直接使用它。



```

1  from pydub import AudioSegment
2
3  # wave 和 raw 格式不使用ffmpeg
4  wav_audio = AudioSegment.from_file("/path/to/sound.wav", format="wav")
5  raw_audio = AudioSegment.from_file("/path/to/sound.raw", format="raw", frame_rate=44100, channels=2, sample_width:
6
7  # 其他格式均需要ffmpeg
8  mp3_audio = AudioSegment.from_file("/path/to/sound.mp3", format="mp3")
9
10 # 使用一个已经打开的文件句柄
11 with open("/path/to/sound.wav", "rb") as wav_file:
12     audio_segment = AudioSegment.from_file(wav_file, format="wav")

```



内容举报



返回顶部

第一个参数是要读取的文件或要读取的文件句柄的路径（作为字符串）。

支持的关键字参数：

`format`| 示例：“aif”| default：“mp3” 输出文件的格式。支持“wav”和“raw”本机，需要ffmpeg的所有其他格式。“raw”文件需要3个额外的关键字参数，`sample_width`，`frame_rate`，和`channels`，下面表示：raw只。这个额外的信息是必需的，因为原始音频文件没有标题包括这个信息在文件本身像wav文件。

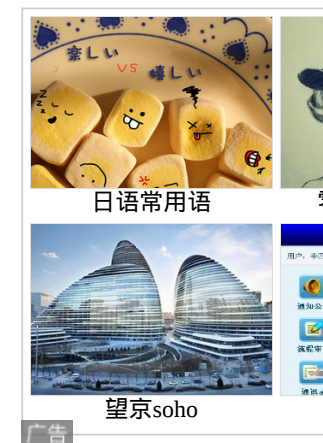
`sample_width`| 示例：2 rawonly - 1用于8位音频2用于16位（CD质量）和4用于32位。它是每个样本的字节数。

`channels`| 示例：1 raw仅 - 1用于单声道，2用于立体声。

`frame_rate`| 示例：2 rawonly - 也称为采样率，常用值为44100（44.1kHz - CD音频）和48000（48kHz - DVD音频）

`AudioSegment (...).export ()`

将AudioSegment对象写入文件 - 返回输出文件的文件句柄（虽然这个句柄可能没什么卵用）。



内容举报



返回顶部


```

1 from pydub import AudioSegment
2 sound = AudioSegment.from_file("/path/to/sound.wav", format="wav")
3
4 # 简单的输出
5 file_handle = sound.export("/path/to/output.mp3", format="mp3")
6
7 # 复杂一些的输出
8 file_handle = sound.export("/path/to/output.mp3",
9                             format="mp3",
10                             bitrate="192k",
11                             tags={"album": "The Bends", "artist": "Radiohead"},
12                             cover="/path/to/albumcovers/radioheadthebends.jpg")

```

第一个参数是写入输出的位置（作为字符串），或要写入的文件句柄。如果不传递输出文件或路径，则会生成一个临时文件。



支持的关键字参数：



format| 示例：“aif”| default：“mp3” 输出文件的格式。支持“wav”和“raw”本机，需要ffmpeg的所有其他格式。



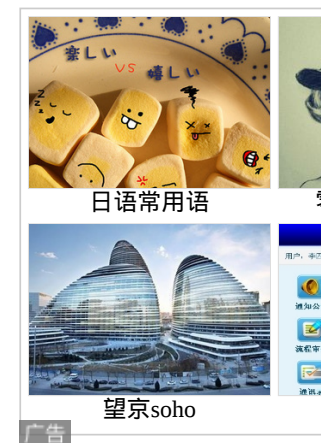
codec| 示例：“libvorbis” 对于可能包含使用不同编解码编码的内容的格式，您可以指定希望编码器使用的编解码器。例如，“ogg”格式经常与“libvorbis”编解码器一起使用。（需要ffmpeg）

bitrate| 示例：“128k” 对于压缩格式，您可以传递希望编码器使用的比特率（需要ffmpeg）。每个编解码器接受不同的比特率参数，所以请查看ffmpeg文档的详细信息（比特率通常显示为-b，-ba或-a:b）。

tags| 示例：{"album": "1989", "artist": "Taylor Swift"} 允许您为编码器提供媒体信息标签（需要ffmpeg）。并非所有格式都可以接收标签（mp3可以）。

parameters| 示例：["-ac", "2"] 将其他ffmpeg参数传递到ffmpeg调用。这些被添加到调用结束（在输出文件部分）。

id3v2_version| 示例：“3”| default：“4” 设置ffmpeg使用的ID3v2版本，将标签添加到输出文件中。如果您希望



内容举报



返回顶部

Windows Explorer显示标签，请”3”在此处使用（源）。

cover| 示例：”/path/to/imgfile.png” 允许您提供封面图像（图像文件的路径）。目前，只有MP3文件允许此关键字参数。封面图片必须是jpeg，png，bmp或tiff文件。

AudioSegment.empty（）

创建空AudioSegment。

```

1
1 from pydub import AudioSegment
2 empty = AudioSegment.empty()
3
4 len(empty) == 0

```

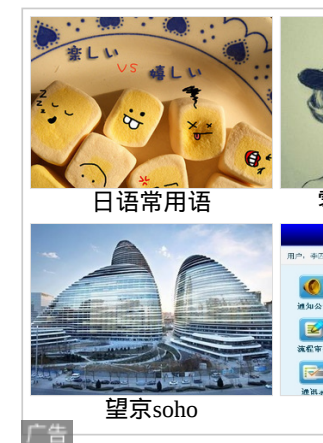
这对于使用循环合成长音频很有用：

```

1 from pydub import AudioSegment
2
3 sounds = [
4     AudioSegment.from_wav("sound1.wav"),
5     AudioSegment.from_wav("sound2.wav"),
6     AudioSegment.from_wav("sound3.wav"),
7 ]
8
9 playlist = AudioSegment.empty()
10 for sound in sounds:
11     playlist += sound

```

AudioSegment.silent（）



内容举报

返回顶部

创建无声Audiosegment，可以作为一个占位符 间隔，或作为一个画布覆盖其他声音的顶部。

```
1 from pydub import AudioSegment
2
3 ten_second_silence = AudioSegment.silent(duration=10000)
```

支持的键字参数：

1

duration| 示例：3000| 默认值：1000（1秒）静默的长度AudioSegment，以毫秒为单位

frame_rate| 示例44100| 默认值：11025（11.025 kHz）静音的帧速率（即采样速率），AudioSegment单位为

Hz

AudioSegment.from_mono_audiosegments ()

从多个单声道音频（两个或更多）创建多声道音频输出。传入的每个单声道音频片段应该具有完全相同的长度。

```
1 from pydub import AudioSegment
2
3 left_channel = AudioSegment.from_wav("sound1.wav")
4 right_channel = AudioSegment.from_wav("sound1.wav")
5
6 stereo_sound = AudioSegment.from_mono_audiosegments(left_channel, right_channel)
```

AudioSegment(...).dBFS



内容举报

返回顶部

返回AudioSegment以dBFS 为单位的响度（db相对于最大可能响度）。最大幅度的方波大致为0dBFS（最大响度），而最大幅度的正弦波大致为-3dBFS。

```
1 from pydub import AudioSegment
2 sound = AudioSegment.from_file("sound1.wav")
3
4 loudness = sound.dBFS
```

1

AudioSegment(...).channels

此音频段中的声道数（1表示单声道，2表示立体声）

```
1 from pydub import AudioSegment
2 sound = AudioSegment.from_file("sound1.wav")
3
4 channel_count = sound.channels
```

AudioSegment(...).sample_width

每个采样中的字节数（1表示8位，2表示16位等）。CD音频是16位，（采样宽度为2字节）。

```
1 from pydub import AudioSegment
2 sound = AudioSegment.from_file("sound1.wav")
3
4 bytes_per_sample = sound.sample_width
5 AudioSegment(...).frame_rate
```



内容举报

返回顶部

AudioSegment(...).frame_rate

CD音频有44.1kHz的采样率，这意味着frame_rate将44100（同采样率，见frame_width）。共同的价值观是44100（CD），48000（DVD），，，22050和。240001200011025

```
1 from pydub import AudioSegment
2 sound = AudioSegment.from_file("sound1.wav")
3
4 frames_per_second = sound.frame_rate
```

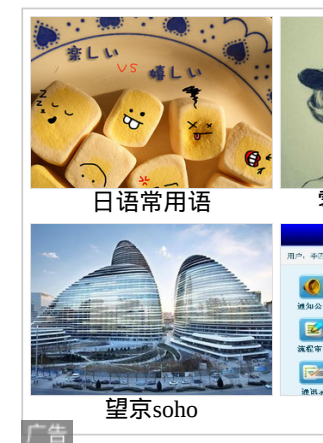
AudioSegment(...).frame_width

每个“帧”的字节数。帧包含每个通道的采样（因此对于立体声，每帧有2个采样，这两个采样同时播放）。frame_width等于channels * sample_width。对于CD音频，它将4（2通道乘以每个样本2字节）。

```
1 from pydub import AudioSegment
2 sound = AudioSegment.from_file("sound1.wav")
3
4 bytes_per_frame = sound.frame_width
```

AudioSegment(...).rms

响度的测量。用于计算dBFS，这是您在大多数情况下应该使用的单位。响度是对数的（rms不是），这使得dB是一个更自然的标度。



内容举报

返回顶部

```

1 from pydub import AudioSegment
2 sound = AudioSegment.from_file ( " sound1.wav " )
3
4 loudness = sound.rms

```

AudioSegment (...) .max



¹AudioSegment中全部采样的最大的幅度。适用于标准化（在pydub.effects.normalize有应用）。



```

1 from pydub import AudioSegment
2 sound = AudioSegment.from_file ( " sound1.wav " )
3
4 peak_amplitude = sound.max

```

AudioSegment (...) .max_dBFS



任何采样的最高幅度AudioSegment，以dBFS为单位（相对于最高可能的幅度值）。适用于标准化（在提供pydub.effects.normalize）。

```

1 from pydub import AudioSegment
2 sound = AudioSegment.from_file ( " sound1.wav " )
3
4 normalized_sound = sound.apply_gain ( - sound.max_dBFS )

```

AudioSegment (...) .duration_seconds



内容举报



返回顶部

返回AudioSegment以秒为单位的持续时间（len(sound)返回毫秒）。它内部调用len()，这是一个方便使用的实现。

```
1 from pydub import AudioSegment
2 sound = AudioSegment.from_file("sound1.wav")
3
4 assert sound.duration_seconds == (len(sound) / 1000.0)
```

1

AudioSegment(...).raw_data

AudioSegment的原始音频数据。用于与其他音频库或想要以字节形式的音频数据的古怪的API进行交互。如果你正在实现效果或其他直接信号处理，也派上用场。



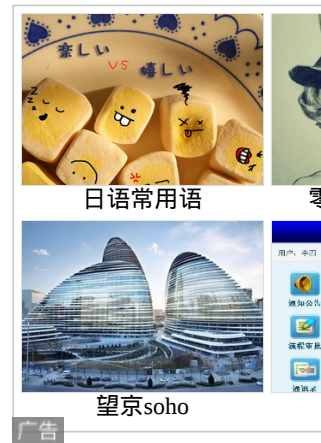
你可能不需要这个，但如果你真的...这些会有用的。



```
1 from pydub import AudioSegment
2 sound = AudioSegment.from_file("sound1.wav")
3
4 raw_audio_data = sound.raw_data
```

AudioSegment(...).frame_count()

返回AudioSegment的帧数。或者，您可以传递ms键控参数以检索在该AudioSegment切片的毫秒数的帧数。



内容举报



返回顶部


```

1 from pydub import AudioSegment
2 sound = AudioSegment.from_file (" sound1.wav ")
3
4 number_of_frames_in_sound = sound.frame_count ( )
5
6 number_of_frames_in_200ms_of_sound = sound.frame_count ( ms = 200 )

```



支持的关键字参数：

1

ms 示例：3000| default：（None整个持续时间AudioSegment）当指定时，方法返回X毫秒内的帧数

AudioSegment



AudioSegment (...) .append ()

返回一个新的AudioSegment，通过附加另一个AudioSegment到这一个（即，添加到末尾），可选使用交叉淡化创建。AudioSegment(...).append()在AudioSegment与+操作符一起添加对象时在内部使用。



默认情况下，使用100ms（0.1秒）交叉淡入淡出来消除爆裂声和爆裂声。



内容举报



返回顶部

```

1 from pydub import AudioSegment
2 sound1 = AudioSegment.from_file (" sound1.wav ")
3 sound2 = AudioSegment.from_file (" sound2.wav ")
4
5 # 默认100毫秒淡入淡出
6 相结合 = sound1.append ( SOUND2 )
7
8 # 5000 ms crossfade
9 combined_with_5_sec_crossfade = sound1.append ( sound2 , crossfade = 5000 )
10
11 # 无交叉淡入淡出
12 no_crossfade1 = sound1.append(sound2, crossfade=0)
13
14 # 无交叉淡入淡出
15 no_crossfade2 = sound1 + sound2

```

支持的关键词参数：

crossfade | 示例：3000 | default：（ 100整个持续时间AudioSegment ）当指定时，方法返回X毫秒内的帧数
AudioSegment

AudioSegment (...) .overlay ()

叠加其他AudioSegment到这一个AudioSegment。这么做的结果是AudioSegment将同时播放。如果叠加AudioSegment比这一个长，则结果将被截断（因此叠加声音的结束将被截断）。AudioSegment即使使用loop和times关键字参数，结果总是与此相同的长度。

由于AudioSegment对象是不可变的，你可以通过将较短的声音覆盖在较长的声音上，或通过创建一个AudioSegment具有适当持续时间的静音，并将两个声音覆盖到那个声音上来解决这个问题。



内容举报

返回顶部

```

1 from pydub import AudioSegment
2 sound1 = AudioSegment.from_file (" sound1.wav ")
3 sound2 = AudioSegment.from_file (" sound2.wav ")
4
5 played_together = sound1.overlay ( sound2 )
6
7 sound2_starts_after_delay = sound1.overlay ( sound2 , position = 5000 )
8
9 sound2_repeats_until_sound1_ends = sound1.overlay ( sound2 , loop = true )
10
11 sound2_plays_twice = sound1.overlay ( sound2 , times = 2 )
12
13 # 假设sound1为30秒长，sound2为5秒长：
14 sound2_plays_a_lot = sound1.overlay ( sound2 , times = 10000 )
15 len ( sound1 ) == len ( sound2_plays_a_lot )

```

支持的关键词参数：

position | 示例：3000 | default：（ 0开头AudioSegment ）覆盖AudioSegment将不会开始，直到X毫秒过去

loop | 示例：True | 默认：（ False整个持续时间AudioSegment ）覆盖AudioSegment将重复（从开始position）直到此结束AudioSegment

times | 示例：4 | 默认：（ 1整个持续时间AudioSegment ）覆盖AudioSegment将重复X次（从开始position），但仍将被截断到此长度AudioSegment

AudioSegment (...) .apply_gain (gain)

改变的幅度（一般，响度）AudioSegment。增益以dB为单位。此方法内部使用+。



内容举报

返回顶部

```

1 from pydub import AudioSegment
2 sound1 = AudioSegment.from_file (" sound1.wav ")
3
4 louder_via_method = sound1.apply_gain ( + 3.5 )
5 louder_via_operator = sound1 + 3.5
6
7 quieter_via_method = sound1.apply_gain ( - 5.7 )
8 quieter_via_operator = sound1 - 5.7

```



1

AudioSegment (...) .fade ()

更通用（更灵活）的淡入淡出方法。您可以指定start和end，或两者之一与持续时间（例如start和duration）。

```

1 from pydub import AudioSegment
2 sound1 = AudioSegment.from_file (" sound1.wav ")
3
4 fade_louder_for_3_seconds_in_middle = sound1.fade ( to_gain = + 6.0 , start = 7500 , duration = 3000 )
5
6 fade_quieter_between_2_and_3_seconds = sound1.fade ( to_gain = - 3.5 , start = 2000 , end = 3000 )
7
8 # 简单的方法是使用.fade_in ( ) 。注意：-120dB基本上就是静音了。
9 fade_in_the_hard_way = sound1.fade ( from_gain = - 120.0 , start = 0 , duration = 5000 )
10 fade_out_the_hard_way = sound1.fade ( to_gain = - 120.0 , end = 0 , duration = 5000 )

```

支持的关键字参数：



内容举报



返回顶部

to_gain| 示例：-3.0| 默认值：0（0dB，无变化）衰减结束时的变化。-6.0意味着衰落将从0dB（无变化）到-6dB，并且在衰落之后的所有内容将是-6dB。

from_gain| 示例：-3.0| 默认值：0（0dB，无变化）在淡入淡出开始时更改。-6.0意味着衰落（和之前的所有音频）将在-6dB将衰减到0dB - 衰落之后的音频的其余部分将在0dB（即，不变）。

start| 示例：7500| NO DEFAULT开始衰落的位置（以毫秒为单位）。5500意味着褪色将在5.5秒后开始。

end| 示例：4| NO DEFAULT覆盖AudioSegment将重复X次（从开始position），但仍将被截断到此长度

AudioSegment

duration| 示例：4| NO DEFAULT您可以使用start或end使用持续时间，而不是指定两者 - 为方便起见。

AudioSegment (...).fade_out()

淡出（静默）AudioSegment结尾。内部使用.fade()。

支持的关键字参数：

duration| 示例：5000| NO DEFAULT衰减应该持续多长时间（以毫秒为单位）。直接传递到.fade()内部

AudioSegment (...).fade_in()

淡入（从沉默）AudioSegment开头。内部使用.fade()。

支持的关键字参数：

duration| 示例：5000| NO DEFAULT衰减应该持续多长时间（以毫秒为单位）。直接传递到.fade()内部

AudioSegment (...).reverse()



内容举报

返回顶部

制作一个AudioSegment向后播放的副本。适用于Pink Floyd, screwing around和一些音频处理算法。

AudioSegment (...) .set_sample_width ()

AudioSegment使用指定的样本宽度（以字节为单位）创建此版本的等效版本。增加该值通常不会导致质量下降。减少它肯定会导致质量的损失。较高的样品宽度意味着更大的动态范围。

AudioSegment (...) .set_frame_rate ()

AudioSegment使用指定的帧速率（以Hz为单位）创建此版本的等效版本。增加该值通常不会导致质量下降。减少它肯定会导致质量的损失。较高的帧速率意味着较大的频率响应（可以表示较高的频率）。

AudioSegment (...) .set_channels ()

创建具有指定AudioSegment数量通道的等效版本（1是单声道，2是立体声）。从单声道转换为立体声不会导致任何可听到的变化。从立体声转换为单声道可能会导致质量损失（但只有左右通道不同）。

AudioSegment (...) .split_to_mono ()

将立体声AudioSegment分为两个，每个通道一个（左/右）。返回一个包含新AudioSegment对象的列表，左声道在索引0，右声道在索引1。

AudioSegment (...) .apply_gain_stereo ()



内容举报

返回顶部

```

1 from pydub import AudioSegment
2 sound1 = AudioSegment.from_file("sound1.wav")
3
4 stereo_balance_adjusted = sound1.apply_gain_stereo(-6, +2)

```

将增益应用于立体声的左右声道AudioSegment。如果AudioSegment是单声道，它将在应用增益之前转换为立体声。

1
两个增益参数以dB为单位。

AudioSegment (...) .pan ()

```

1 from pydub import AudioSegment
2 sound1 = AudioSegment.from_file ( " sound1.wav " )
3
4 # 向右平移15%
5 panned_right = sound1.pan ( + 0.15 )
6
7 # 向左平移50%
8 panned_left = sound1.pan ( - 0.50 )

```

取一个位置参数，平移量，应在-1.0（100%左）和+1.0（右100%）之间，

当pan_amount == 0.0时，左/右平衡不更改。

平移不会改变感知到的隆起，但是由于响度在一侧上减小，另一侧需要更大地补偿。

AudioSegment (...) .get_array_of_samples ()



以（数值）样本数组的形式返回原始音频数据。注意：如果音频具有多个通道，则每个通道的采样将被序列化 - 例如，立体声音频将是类似的[sample_1_L, sample_1_R, sample_2_L, sample_2_R, ...]。

这种方法主要用于实现音频效果和其他信号处理。

```
1 from pydub import AudioSegment
2 sound = AudioSegment.from_file ("sound1.wav")
3
4 samples = sound.get_array_of_samples ()
5
6 # 然后修改样本...
7
8 new_sound = sound._spawn ( samples )
```

注意，当使用numpy或scipy时，您需要在生成之前转换回数组：

```
1 import array
2 import numpy as np
3 from pydub import AudioSegment
4
5 sound = AudioSegment.from_file ("sound1.wav")
6 samples = sound.get_array_of_samples ()
7
8 shifted_samples = np.right_shift ( samples , 1 )
9
10 # 现在你必须转换回一个array.array
11 shifted_samples_array = array.array ( sound.array_type , shifted_samples )
12
13 new_sound = sound._spawn ( shifted_samples_array )
```



内容举报

返回顶部

AudioSegment (...) .get_dc_offset ()

返回介于-1.0和1.0之间的值，表示通道的直流偏移。这是使用audioop.avg()并通过样本最大值对结果进行归一化计算的。

支持的关键词参数：



¹ channel | 示例：2 | 默认值：1 选择左（1）或右（2）通道来计算DC偏移。如果段是单声道，则忽略此值。



AudioSegment (...) .remove_dc_offset ()



从通道中删除直流偏移。这是通过调用audioop.bias()实现的，所以小心溢出。



支持的关键词参数：



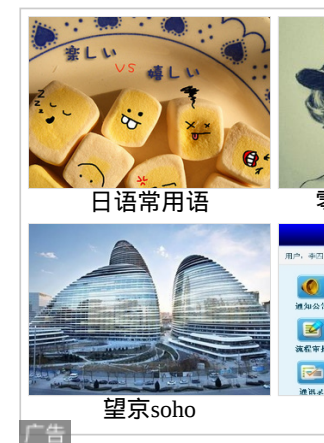
¹ channel | 示例：2 | 默认值：无选择左（1）或右（2）通道删除直流偏移。如果值为None，则从所有可用通道中删除。如果段是单声道，则忽略此值。



offset | 示例：-0.1 | 默认值：None 要从通道中删除的偏移量。如果为无，计算偏移。偏移值必须介于-1.0和1.0之间。

AudioSegment (...) .invert_phase ()

制作一个副本AudioSegment并反转信号的相位。可以产生反相波用于噪声抑制或消除。



内容举报

返回顶部

以上内容基本是机翻 对照原文效果更好
以及一下部分的API文档还没有完成

- Playback (pydub.playback)
- Signal Processing (compression, EQ, normalize, speed change - - pydub.effects, pydub.scipy_effects)
- Signal generators (Sine, Square, Sawtooth, Whitenoise, etc - pydub.generators)
- 👍 • Effect registration system (basically the pydub.utils.register__pydub__effect decorator)
 - 1 • Silence utilities (detect silence, split on silence, etc - pydub.silence)



版权声明：署名（BY）-非商业性使用（NC）-相同方式共享（SA）<https://creativecommons.org/licenses/by-nc-sa/3.0/cn/deed.zh>



发表你的评论

(http://my.csdn.net/weixin_35068028)



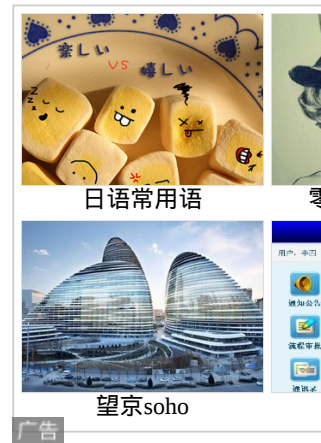
Python中pydub使用详解 (http://blog.csdn.net/gent__chen/article/details/52931728)

pydub是python中用户处理音频文件的一个库。 安装： 1、安装pip工具：sudo apt-get install python-pip 2、安装pydub：sudo pip install pydub



gent__chen (http://blog.csdn.net/gent__chen) 2016年10月26日 16:25 4281

python pydub 用法（2） (<http://blog.csdn.net/y734564892/article/details/77601017>)



内容举报



返回顶部

AudioSegment.silent() 创建一段没有声音的音频片段 from pydub import AudioSegment ten_second_silence = AudioSegment.silent(duration=10)



y734564892 (<http://blog.csdn.net/y734564892>) 2017年08月26日 14:07 582



【前端逆袭记】我是怎么从月薪4k到40k的！

谨以此篇文章献给我奋斗过的程序人生！我第一次编码是在我大一的时候....

(http://www.baidu.com/cb.php?c=lgF_pyfqHmknj0dP1f0lZ0qnfK9ujYzP1ndPWb10Aw-5Hc3rHnYnHb0TAq15HfLPWRznjb0T1YYnjfLuHR3n1RzPWmdnHb10AwY5HDdnHfdn1TLP1m0lgF_5y9YIZ0IQzq-uZR8mLPbUB48ugfEIAqspynElvNBnHqdIAdxTvqdThP-5yF_UvTkn0KzujYk0AFV5H00TZcqn0KdpyfqHRLPjnvnfKEpyfqHc4rj6kP0KWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqnHbknWc)



【Python】利用pydub库操作音频文件 (<http://blog.csdn.net/wr132/article/details/59133085>)



最近使用Python调用百度的REST API实现语音识别，但是百度要求音频文件的压缩方式只能是pcm（不压缩）、wav、opus、speex、amr，这里面也就wav还常见一点，但是一般设备录音得到...



wr132 (<http://blog.csdn.net/wr132>) 2017年03月01日 20:37 2320

音频信号中采样率和比特率以及采样编码的问题 (<http://blog.csdn.net/zjm750617105/article/d...>)

采样率和比特率：cmd: soxi aa.mp3 Input File : 'aa.mp3' Channels : 1 Sample Rate : 44100 Pr...



zjm750617105 (<http://blog.csdn.net/zjm750617105>) 2016年10月23日 15:40 1318

Ubuntu下Python音频播放处理库--安装总结 (<http://blog.csdn.net/piggymumu/article/details...>)



互相学习



内容举报



返回顶部

 piggymumu (<http://blog.csdn.net/piggymumu>) 2016年03月01日 22:24  824

 <p>1 数字制噪器 180.00/件 厂家直销 格雷特 MPX108-SM8 8系统 22年</p>	 <p>2 1.50/个 供应TOTX147PL, 光纤 连接器</p>	 <p>3 180.00/件 订做精图 网络箱 ONU 箱 综合箱 网络配线箱</p>
---	---	--



Python+OpenCV学习 (18) ---双目视觉深度信息恢复 (<http://blog.csdn.net/FireMicrocosm/>...

利用python学习OpenCV，个人感觉比较方便。函数的形式与C++基本相同，所以切换过来还是比较好的，对于像我这种对python不太熟练的人，使用python的集成开发环境PyCharm进行学习，...

 FireMicrocosm (<http://blog.csdn.net/FireMicrocosm>) 2015年09月20日 17:28  2970



Python中pydub使用详解 (http://blog.csdn.net/gent__chen/article/details/52931728)

pydub是python中用户处理音频文件的一个库。 安装： 1、安装pip工具：sudo apt-get install python-pip 2、安装pydub：sudo pip install pydub

 gent__chen (http://blog.csdn.net/gent__chen) 2016年10月26日 16:25  4281

python打包VideoCapture 捕获摄像头信息 (<http://blog.csdn.net/kkxgx/article/details/77492...>)

最近使用VideoCapture捕获摄像头信息，摸索了好久。首先要安装VideoCapture，将解压的内容分别拷贝到python安装的目录。我的电脑中如下所示：这样就可以使用VideoCapture...

 kkxgx (<http://blog.csdn.net/kkxgx>) 2012年07月15日 20:10  10470



jboss_esb简单介绍 (<http://download.csdn.net/download/karelwater/1011...>)




内容举报


返回顶部

2017年11月11日 11:49 212KB

下载

**粗糙集的简单介绍PPT** (<http://download.csdn.net/download/jieshuo3532/9...>)[/http://download.csdn.net/download/jieshuo3532/9...](http://download.csdn.net/download/jieshuo3532/9...)

2017年07月10日 16:59 656KB

下载

UML简单介绍(二十三)——基于UML的移动点餐系统分析 (<http://blog.csdn.net/ljtyzhr/article/...>)

1、点餐系统 点餐系统是一个非常常见的系统，在很多酒店也追随移动互联网的步伐，开发了一些移动点餐的系统。经过分析，我们划分了一下点餐系统的主要功能模块：点餐模式，...

ljtyzhr (<http://blog.csdn.net/ljtyzhr>) 2015年06月28日 17:27 2331**gulp教程简单介绍** (http://download.csdn.net/download/fe_super_7/98730...)[/http://download.csdn.net/download/fe_super_7/98730...](http://download.csdn.net/download/fe_super_7/98730...)

2017年06月17日 16:08 66KB

下载

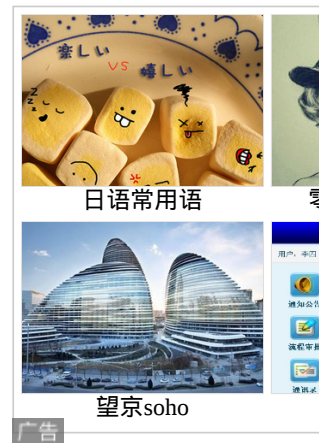
**IBM system z13简单介绍** (<http://download.csdn.net/download/ipadcsdn/...>)[/http://download.csdn.net/download/ipadcsdn/...](http://download.csdn.net/download/ipadcsdn/...)

2015年06月01日 21:16 824KB

下载

**nw.js node-webkit系列 (1) 简单介绍及应用** (<http://blog.csdn.net/zeping891103/article/det...>)

最近使用到了一种nwjs技术进行开发，实际上nwjs是基于nodejs的，它支持nodejs所有的api，主要用于跨平台轻量级桌面应用开发，运行环境包括32位和64位的Window、Linux和Mac...

zeping891103 (<http://blog.csdn.net/zeping891103>) 2016年02月23日 18:34 4705**(java) AOP代理方面的简单介绍** (<http://download.csdn.net/download/wa...>)

内容举报

返回顶部