

サウンド

はじめに

ここでは、Mathematicaを用いて音を作成する方法について述べる。音は波なので、三角関数などを用いれば作成できる。三角関数をグラフ表示するためにPlot関数を用いていたが、PlotをPlayに変更するだけで簡単に音を演奏することができる。ドレミの作成から和音の演奏などまで行ってみよう。

1. PlayとListPlay
2. うなり
3. 倍音と楽器の音色
4. ドレミと和音

1. PlayとListPlay

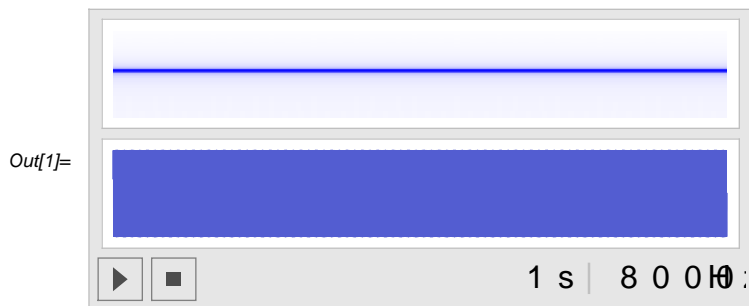
Mathematicaでは、グラフィックだけでなくサウンドも容易に作成することができる。もっとも、普通に用いるのはPlay関数である。Play関数を用いて周波数 f (Hz)の音を s 秒間演奏するためには以下のように入力する。

```
Play[Sin[2π f t],{t,0,s}]
```

 周波数 f (Hz)の音を s 秒間演奏する。

ここでは、例としてラの音を演奏することを考えよう。例えば、ラの音は周波数は440Hzである。ラの音を1分間(4分音符)だけ演奏するためには次のように入力する。

```
In[1]:= Play[Sin[2 Pi 440 t], {t, 0, 1}]
```



ステレオ効果を出すためには、チャンネル1、2の音の周波数を f_1, f_2 とすれば次のようになる。

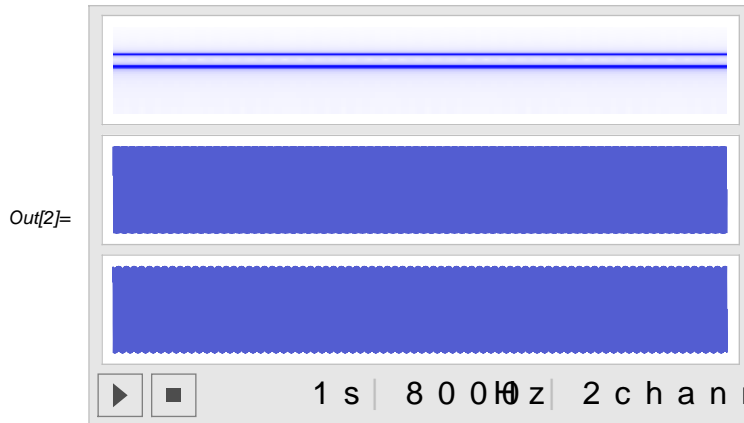
```
Play[{ Sin[2π f1 t], Sin[2π f2 t]},{t,0,s}]
```

チャンネル 1、2 の音の周波数がそれぞれ f_1, f_2 (Hz) の音を s 秒間演奏する。

つまり、チャンネル 1、2 の音の周波数を、ならべて {} でくくって指定すればよい。また、チャンネルがさらに多い場合は、同様にして関数を {} で複数くくって表示すればよい。

試みに、チャンネル 1 と 2 で、それぞれ 1 オクターブ異なるラの音を 1 秒間発生すると以下のようなになる。

```
In[2]:= Play[{Sin[2 Pi 440 t], Sin[2 Pi 880 t]}, {t, 0, 1}]
```



何らかのデータ（離散データ）をサウンドとして聴くためには、ListPlayコマンドを用いる。

```
ListPlay[{リスト}]
```

リストとして与えられた離散データからサウンドを生成する。

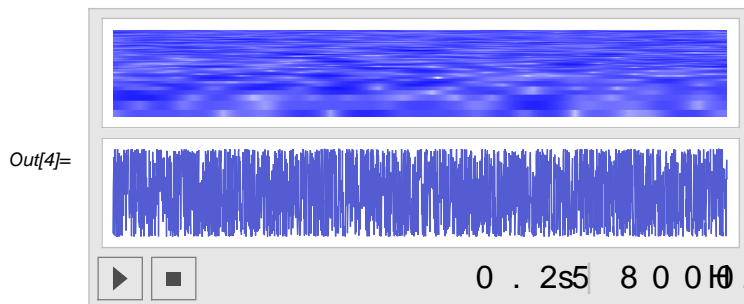
最初にデータを作成し、リストnlに定義する。

```
In[3]:= nl = Table[Sin[500 Pi n / 1000 + 10 RandomReal[]], {n, 2000}]
```

```
nl = Table[Sin[ $\frac{500 \pi n}{1000} + 10 \text{RandomReal}[]$ ], {n, 2000}]
```

ここで、Random[]は0から1の間の疑似乱数を発生する関数である。この場合、サンプル点を2000個発生することになる。そして、このサンプル点について音を発生するためにはListPlay関数を用いる。つまり、

```
In[4]:= ListPlay[nl]
```



一度作成した音をもう一度発生するには、Showコマンドを用いても良い。

```
In[4]:= s1 = ListPlay[nl]
```

```
In[5]:= Show[s1]
```

演習問題1.1

ドレミの周波数は、それぞれ 261.6, 293.6, 329.6 Hzです。これらの音を発生しなさい。

演習問題1.2

ドの音のサンプル点数を 100, 500, 1000, 2000, 4000, 8000 と変更して発生し、それらの違いについて述べなさい。

演習問題1.3

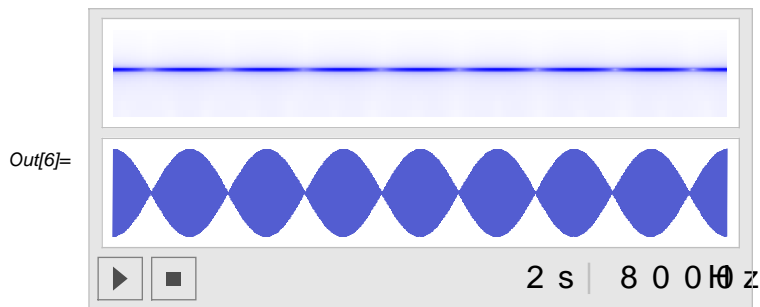
ミの音(329.6 Hz)と、その1オクターブと2オクターブ上のミの音を、それぞれチャンネル1, 2, 3として音を発生しなさい。

2. うなり

周波数が非常に近い波が重なると、振幅が時間的・空間的にも周期的に変化する現象を「うなり」という。波が可聴音であれば、音が大きくなったり、小さくなったりする。

最初に、440Hzの音に444Hzの音を重ねると以下のようなになる。

```
In[6]:= Play[Sin[2 Pi 440 t] + Sin[2 Pi 444 t], {t, 0, 2}]
```

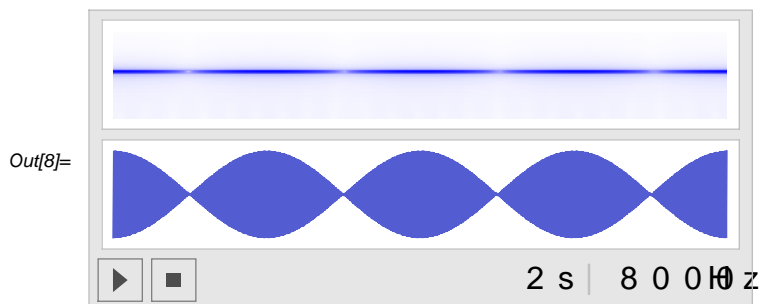


また、440Hzの音に442Hzの音を重ねると以下のようなになる。先の場合に比べて、すこし振幅の変化がゆっくりしているのがわかるでしょうか。

```
In[7]:= f = Sin[2 Pi 440 t] + Sin[2 Pi 442 t]
```

```
Out[7]= Sin[880 π t] + Sin[884 π t]
```

```
In[8]:= Play[f, {t, 0, 2}]
```



演習問題2.1

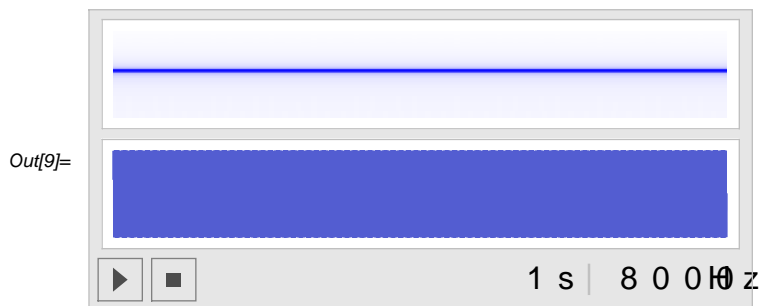
レの音に対して、+2,4,10Hzの音を重ねた音を生成し、うなりの発生を確認しなさい。

3. 倍音と楽器の音色

■ 基本音と倍音

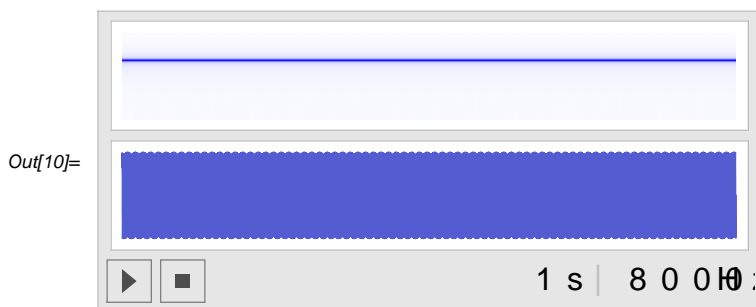
ある音が1オクターブあがる場合、その周波数は2倍になる。従って、1オクターブ下がる場合、周波数は半分となる。試みとして、1オクターブ上のラは880Hzとなる。まず、基本音の場合は次のようになる。

```
In[9]:= Play[Sin[2 Pi 440 t], {t, 0, 1}]
```



次に、1オクターブ高い音は次のようになる。

```
In[10]:= Play[Sin[2 Pi 880 t], {t, 0, 1}]
```



■ 音色

楽器の場合、音色には基本音と倍音が合わさっている。同じラの音であっても、楽器の場合、周波数440Hzの基本音に、それと数オクターブことなる音が加わっている。例えば、周波数440Hzの基本音のラと、それに1オクターブ高い音が重なったラの音を演奏するためには次のように入力する。

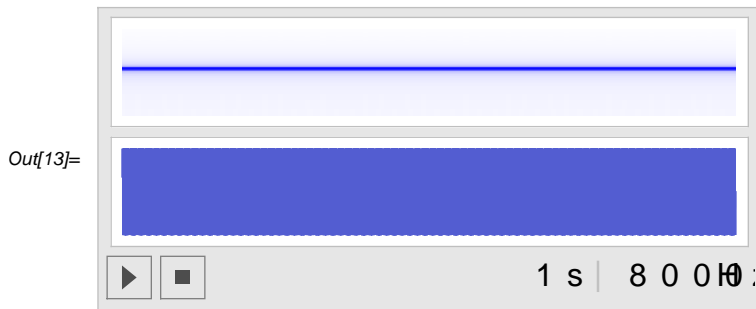
```
In[11]:= f1 = Sin[2 Pi 440 t]
```

```
Out[11]= Sin[880 π t]
```

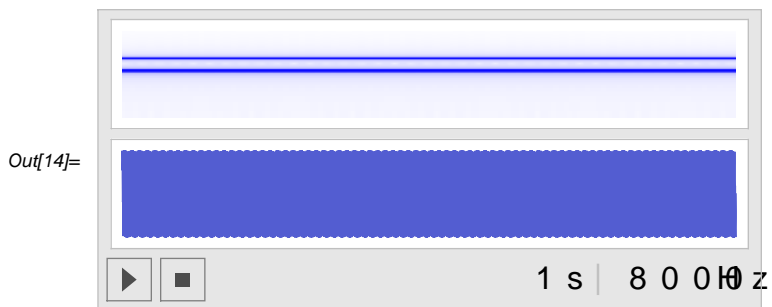
```
In[12]:= f2 = Sin[2 Pi 440 t] + Sin[2 Pi 880 t]
```

```
Out[12]= Sin[880 π t] + Sin[1760 π t]
```

```
In[13]:= Play[f1, {t, 0, 1}]
```

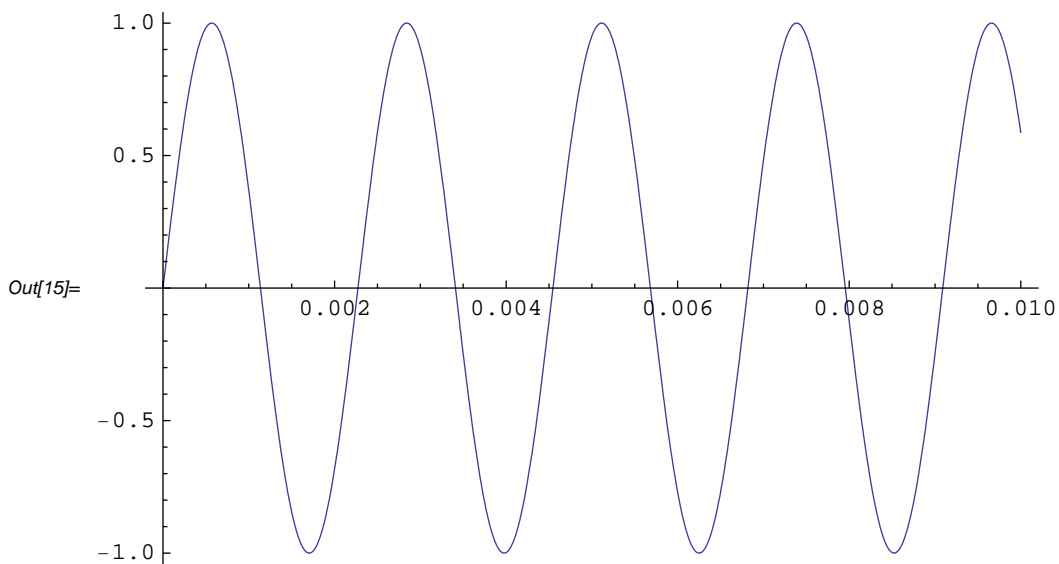


```
In[14]:= Play[f2, {t, 0, 1}]
```

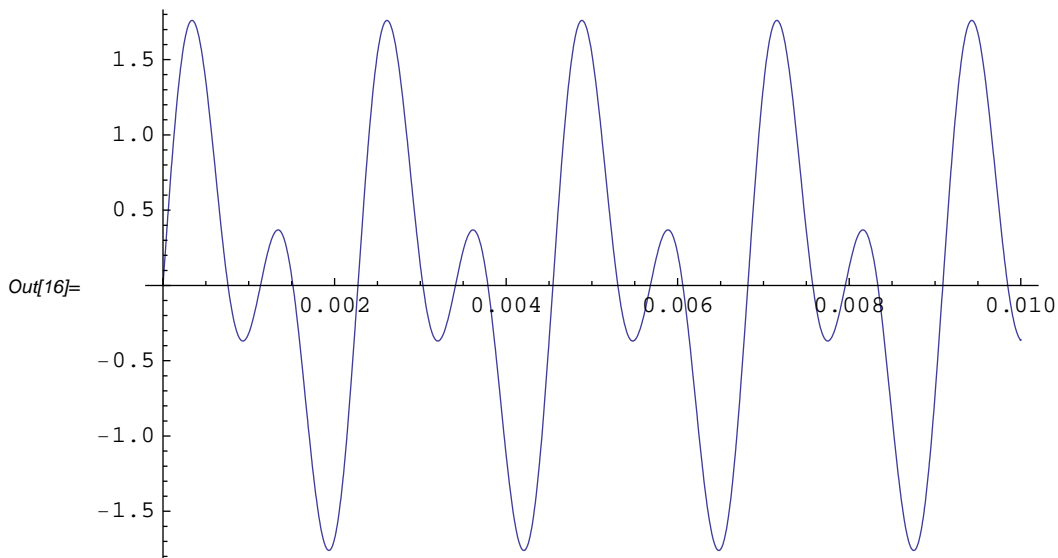


ここで音の波形を $0 \leq t \leq 0.01$ でグラフ化してみると、それぞれ次のようになる。

```
In[15]:= Plot[f1, {t, 0, .01}]
```



```
In[16]:= Plot[f2, {t, 0, .01}]
```



演習問題3.1

ミの音について、基本音とそれより1オクターブ高い倍音をそれぞれ個別に発生しなさい。また、基本音と、基本音にそれより1オクターブ、2オクターブ高い音を重ねた音を発生させて聞き比べなさい。

演習問題3.2

演習問題5で発生した、ミの基本音と、基本音にそれより1オクターブ、2オクターブ高い音を重ねた音の波形を $0 \leq t \leq 0.01$ でグラフ化しなさい。

4. ドレミと和音

■ ドレミファソラシドを演奏する。

ドレミファソラシドの周波数は、それぞれ 261.6, 293.6, 329.6, 349.2, 392, 440, 493.8, 523.3 Hzなので、最初にそれをリスト定義する。

```
In[17]:= {do, re, mi, fa, so, ra, shi, do2} =
          {261.6, 293.6, 329.6, 349.2, 392, 440, 493.8, 523.3}
```

```
Out[17]= {261.6, 293.6, 329.6, 349.2, 392, 440, 493.8, 523.3}
```

きちんと代入されているか、確認してみると次のようになる。

```
In[18]:= do
```

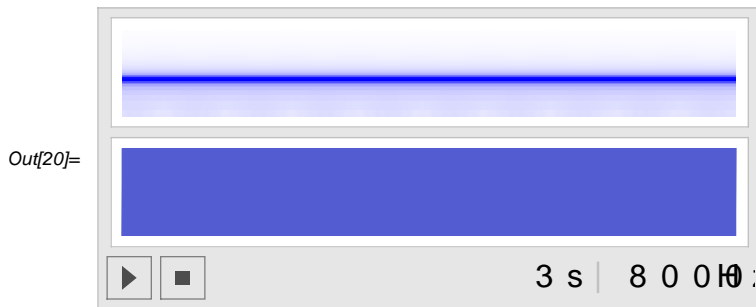
```
Out[18]= 261.6
```

```
In[19]:= fa
```

```
Out[19]= 349.2
```

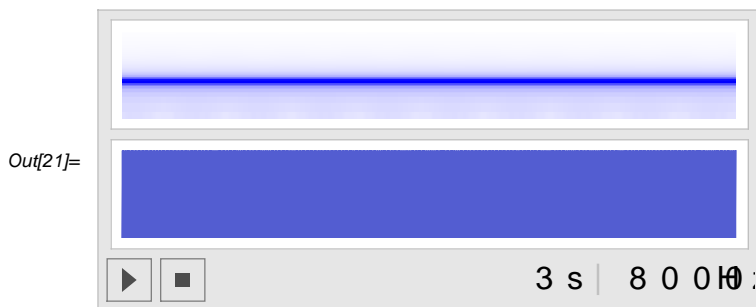
そして、先の周波数を用いてドを演奏すると、次のようになる。

```
In[20]:= Play[Sin[2 Pi do t], {t, 0, 3}]
```



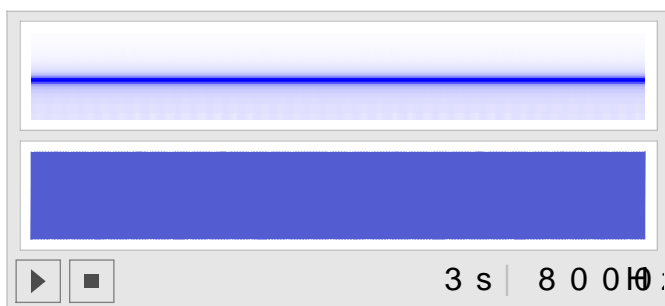
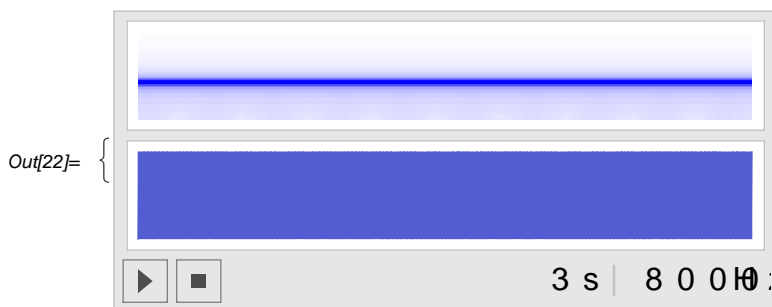
いちいち入力するのは面倒なので、Map関数と準関数の定義を用いることを考える。まず、上記の例を準関数を用いる形で書き換えると次のようになる。

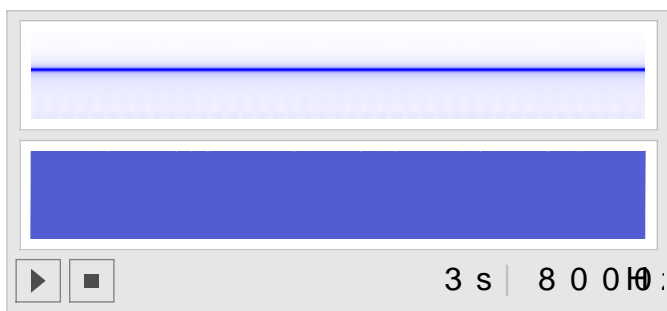
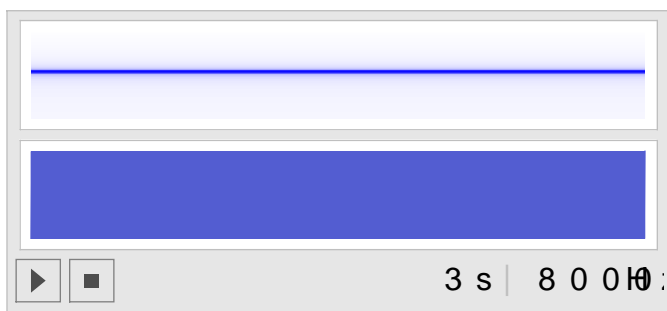
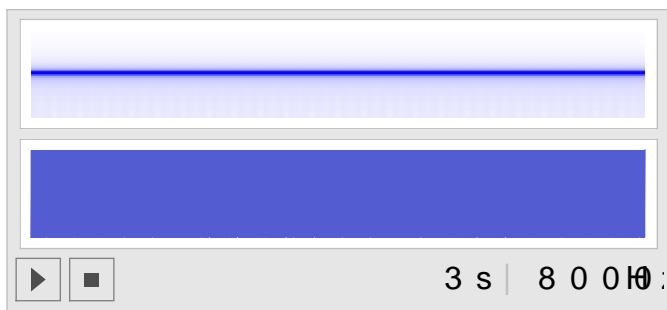
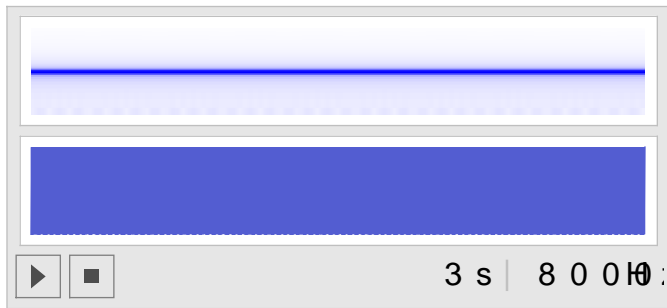
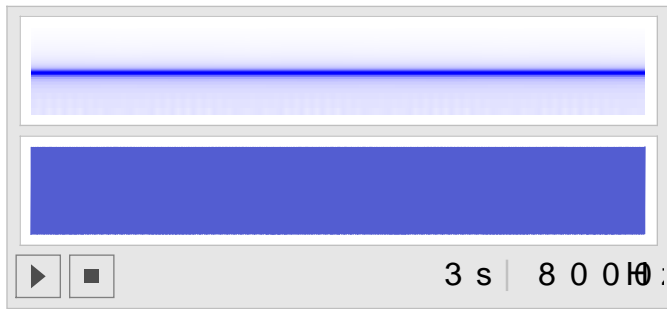
```
In[21]:= Play[Sin[2 Pi # t], {t, 0, 3}] &[do]
```



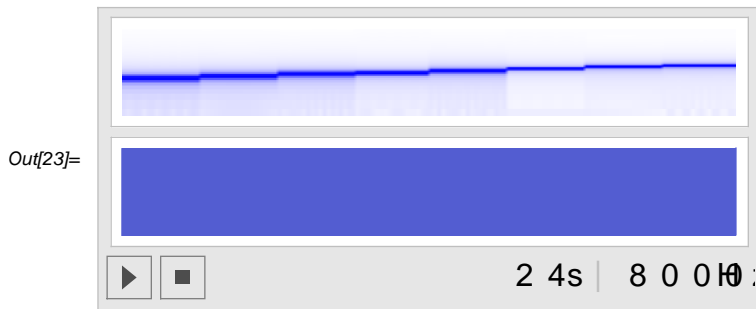
続いてMap関数を用いてドレミファソラシドを演奏すると、次のようになる。

```
In[22]:= Map[Play[Sin[2 Pi t #], {t, 0, 3}] &, {do, re, mi, fa, so, ra, shi, do2}]
```





In[23]:= **Sound** [%]



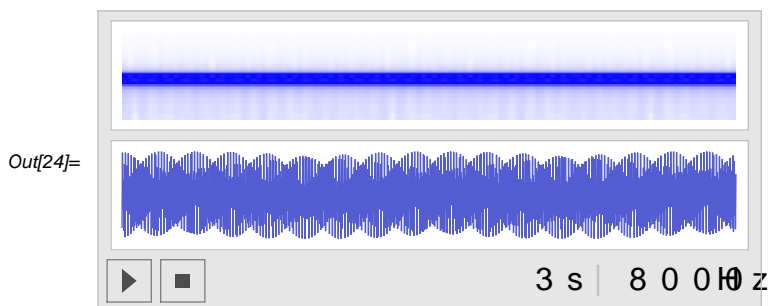
演習問題4.1

先の関係式や定義を利用して、1オクターブ高いドレミファソラシドを演奏しなさい。

■ 和音を演奏する。

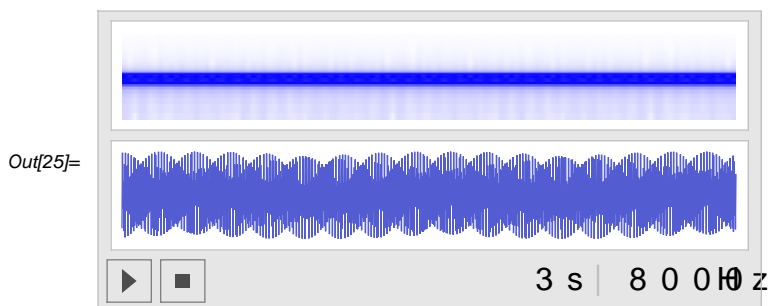
ドミソを3秒間演奏するためには、次のように入力する。

In[24]:= **Play**[**Sin**[2 Pi t do] + **Sin**[2 Pi t mi] + **Sin**[2 Pi t so], {t, 0, 3}]



せっかくなので、少し高度ですがApplyを用いて書き換えてみましょう。かえって、難しくなったように思うかもしれませんが、複雑なものではこちらのほうが便利なきももあります。

In[25]:= **Play**[**Apply**[**Plus**, **Map**[**Sin**[2 Pi # t] &, {do, mi, so}]], {t, 0, 3}]



演習問題4.2

和音のうち、ドファラとシレソを演奏しなさい。

5. 作曲してみよう

「さくら さくら」の出だしのところは「ララシ」である．このところをMathematicaで演奏してみよう．

```
In[26]:= Play[Sin[2 Pi t ra], {t, 0, 0.5}];
         Play[Sin[2 Pi t ra], {t, 0, 0.5}];
         Play[Sin[2 Pi t shi], {t, 0, 1.0}];
```

何だか、おかしくなかつただろうか。じつは、上記の記述では、さいしょの「ララ」の間に切れ目がないので、ひとつの「ラ」となってしまう、全体が「ラシ」と聞こえているのです。そこで、音の切れ目として、

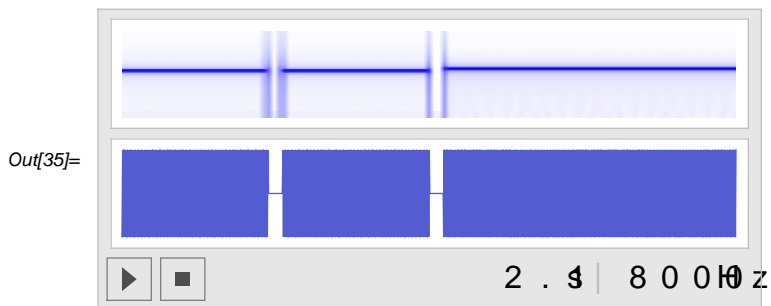
```
Play[Sin[2 Pi t 0], {t, 0, 0.05}]
```

をいれると、すこしすっきりするのではないだろうか。

```
In[29]:= Play[Sin[2 Pi t ra], {t, 0, 0.5}];
         Play[Sin[2 Pi t 0], {t, 0, 0.05}];
         Play[Sin[2 Pi t ra], {t, 0, 0.5}];
         Play[Sin[2 Pi t 0], {t, 0, 0.05}];
         Play[Sin[2 Pi t shi], {t, 0, 1.0}];
```

一つの音楽としてまとめて定義し、再びプレイするには以下のように入力するとよい。

```
In[35]:= Sound[{Play[Sin[2 Pi t ra], {t, 0, 0.5}],
               Play[Sin[2 Pi t 0], {t, 0, 0.05}],
               Play[Sin[2 Pi t ra], {t, 0, 0.5}],
               Play[Sin[2 Pi t 0], {t, 0, 0.05}],
               Play[Sin[2 Pi t shi], {t, 0, 1.0}]}]
```



シャープとフラット

シャープとフラットは元の周波数から $\sqrt[12]{2}$ を足したり、引いたりすればよい。

シャープは次のように

```
{dos, res, mis, fas, sos, ras, shis, do2s} =
  {do, re, mi, fa, so, ra, shi, do2} +  $\sqrt[12]{2}$ 
```

フラットは次のようになる .

```
{dof, ref, mif, faf, sof, raf, shif, do2f} =  
{do, re, mi, fa, so, ra, shi, do2} -  $\sqrt[12]{2}$ 
```

演習問題9

楽譜を参考にして , 「さくらさくら」を最後まで演奏してみなさい.

演習問題10

3つの楽器についてサウンドを作成し,必要とあれば友人とともにそれらを合わせて合奏してみよ.

6. サウンドのエクスポート

サウンドをファイルとして出力するには以下のように入力する.

```
Export["c:\playlist.wav", playlist, "WAV"]
```

この結果,フォルダc:のところにpolaylist.wav というファイルが作成される.