

ミュージック・ブロックスのプログラミング案内

ミュージック・ブロックスを学び、プログラミングするための完全なガイド

ミュージック・ブロックスは子供のための音楽とコードが合体されているプログラミング環境。子供さん達はこのツールで音楽、数学、またコードの基本的な色々が楽しく発見することができます。ミュージック・ブロックスはタートルブロックから作られて、ピッチとリズムのツールもあります。

タートルブロックの案内はミュージック・ブロックスより基本的なコンセプトがありません。この案内は音楽の基本があり、ミュージック・ブロックスの実例があります。

目次

1 初めに

2 音の関係

2.1 音価のブロック

2.2 ピッチのブロック

2.3 和音

2.4 休符

2.5 ドラム

3 音楽でプログラミング

3.1 チャンク

3.2 音楽的の転化

3.2.1 音符動きのブロック

3.2.2 シャープ(嬰)とフラット(変)とフラット(変)

3.2.3 移調をアジャストのブロック

3.2.4 付点音符

3.2.5 演算で音価をかけ割ること

3.2.6 回繰り返し音符

3.2.7 スイング・リズムとタイの音符

3.2.8 音量、クレシェンド、スタッカート、スラーのブロック

3.2.9 音程と相対的な音量の関係

3.2.10 絶対音程

3.2.11 音符転回

3.2.12 逆に

3.2.13 音色と調の設定

3.2.14 ビブラート

3.3 声部

3.4 グラフィックス

3.5 拍子

3.6 相互作用

3.7 ウィジェット

3.7.1 ステータス・モニター

3.7.2 音符のチャンクを作ること

3.7.2.1 ピッチ・タイム行列

3.7.2.2 リズムのブロック

3.7.2.3 タブルのこと

3.7.2.4 「タブル」というのは？

3.7.2.5 Using Individual Notes in the Matrix

3.7.3 リズム・ブロックでリズムを

3.7.4 音楽の音階とモード

3.7.5 ピッチ・ドラム・グラフ

3.7.6 音程の関係のことをピッチスライダーで発見

3.7.7 ピッチ・スライダーで何のピッチでも発見

3.7.8 テンポを変えること

3.8 ミュージック・ブロックスから以遠

この案内の中の例はコードのリンクもあります。そのすぐ楽しめるコードのリンクはライブで再生と書いております。

1. 初めに

[目次に戻す](#) | [次のトピック \(2. 音符の音を鳴らすのに\)](#)

ミュージック・ブロックスはブラウザで実行するために作られています。クローム・ブラウザで一番テストされていますがファイヤーフォックスも実行できます。github.io (ギットハープ) (<https://musicblocks.sugarlabs.org>)のウェブサイトから実行でき、ミュージック・ブロックスのソース・コードもダウンロードして自分のパソコンで実行できます。

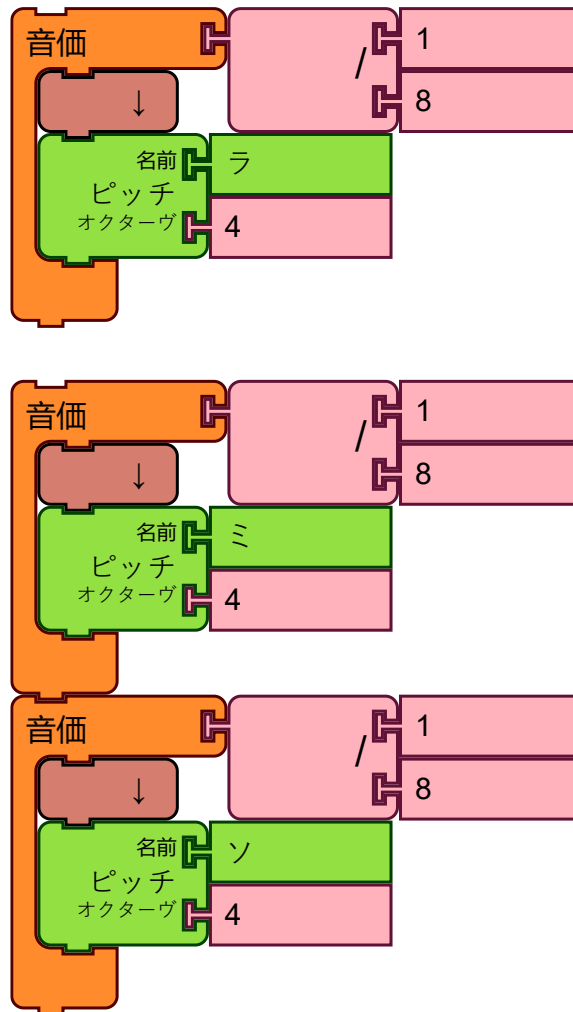
この案内よりミュージック・ブロックスの細かいを楽しみたかったら、[ミュージック・ブロックスの基本](http://github.com/sugarlabs/musicblocks/tree/master/documentation) (<http://github.com/sugarlabs/musicblocks/tree/master/documentation>)をどうぞ、読んでください。ミュージック・ブロックスの元のタートル・ブロックスの細かいを楽しみたかったら、[タートル・ブロックスの基本](http://github.com/sugarlabs/turtleblocksjs/tree/master/documentation) (<http://github.com/sugarlabs/turtleblocksjs/tree/master/documentation>)をどうぞ読んでください。

[1 音の関係](#) [前のトピック \(1. 初めに\)](#) | [目次に戻す](#) | [次のトピック \(3. 音楽でプログラミング\)](#)

ミュージック・ブロックスは音楽の基本の色々があります。例えば、[ピッチ](#), [リズム](#), [音量](#), [音色](#)とシンセのツールもあります。

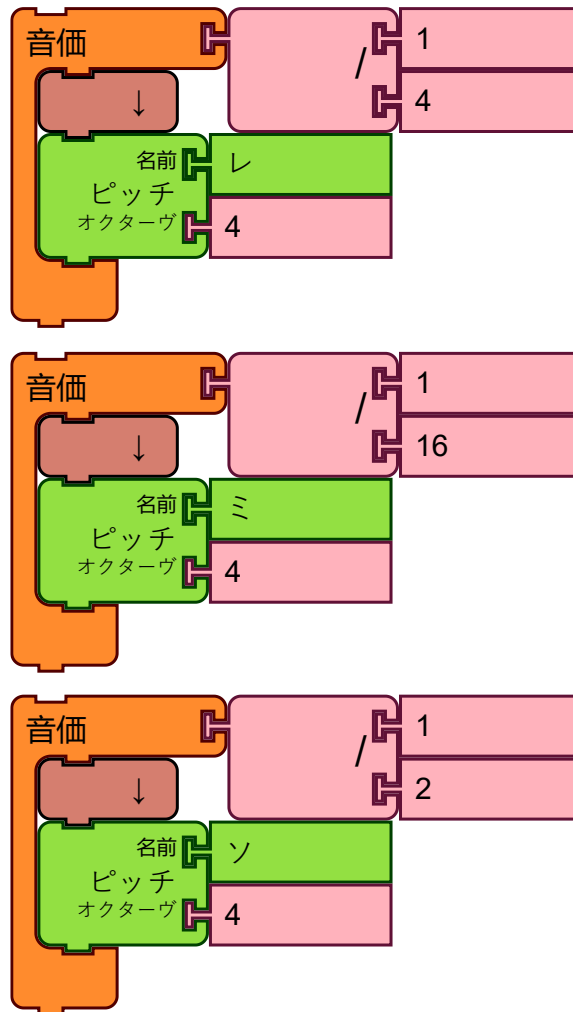
2.1 音価のブロック

ミュージック・ブロックスの一番基本的なブロックは音価ブロックです。音価ブロックの中に[ピッチブロック](#)が入れることができます。音価ブロックはピッチの長さがどのぐらいか決めます。



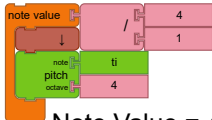

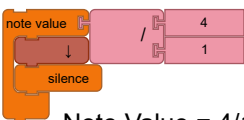

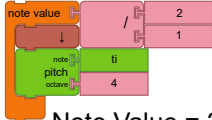

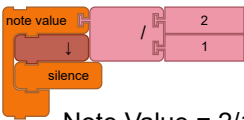

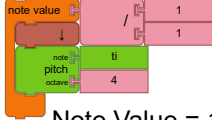
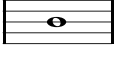
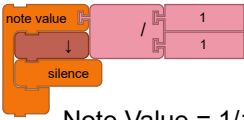
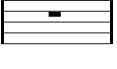


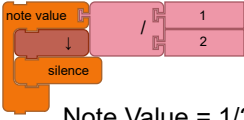



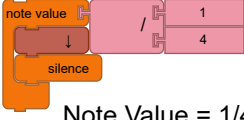



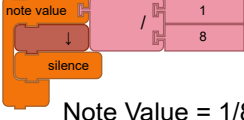
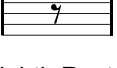
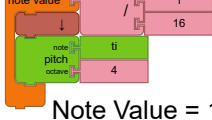

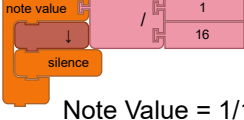
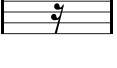


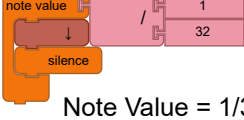
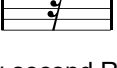
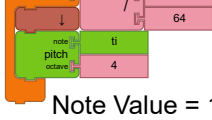

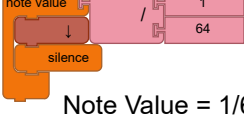
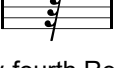

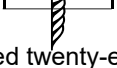
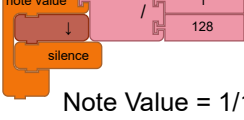
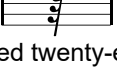
上の例に、一つの（分離した）音価ブロックがあります。そのブロックに1/8の数値の数字ブロックがつながっています。1/8の数値は音符の音価です。その1/8の音価と言うのは八分音符とも言います。

その下に、二つの分理的に鳴らされる音符があります。両方とも'1/8'音符で、全部で音価の合計が'1/4'音符の同じ長さです。



この例に、違う音価があります。上から、四分音符の'1/4'、十六音付の'1/16'、半音符の'1/2'です。

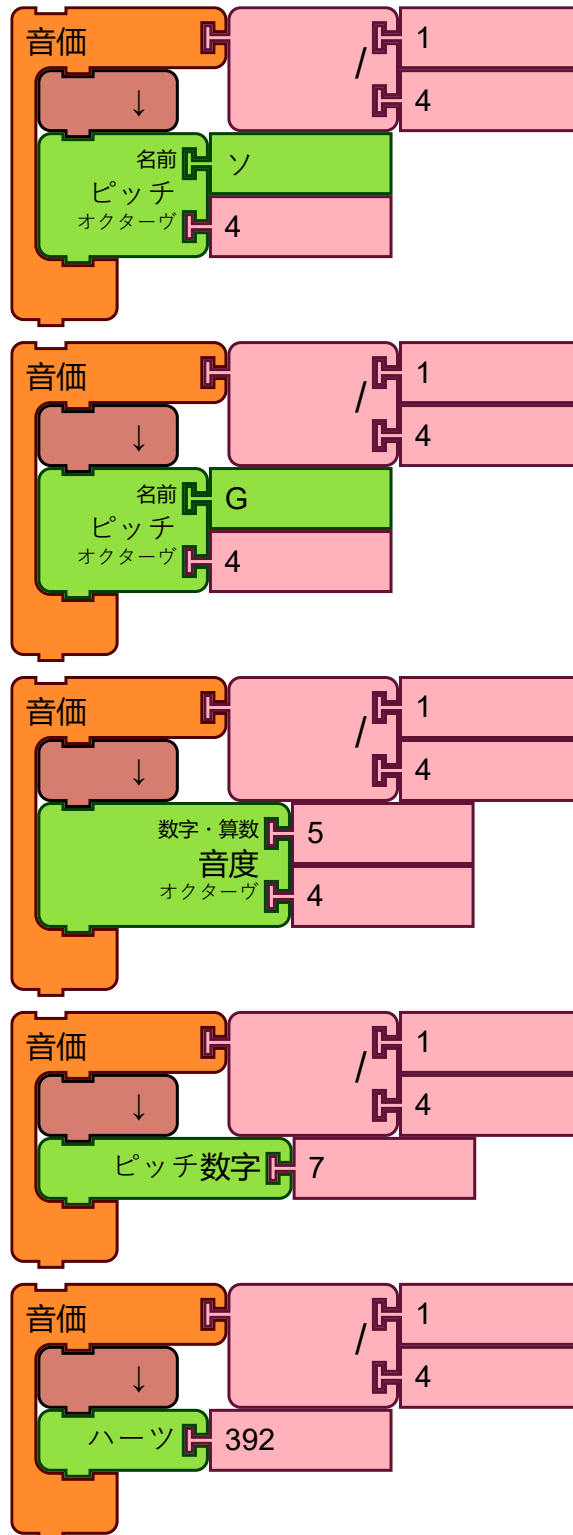
音価ブロックの数値を数字・計算のブロックでいろいろな算数ができますよ。

Note Value Blocks	Western Notation (Notes)	Silence Blocks	Western Notation (Rests)
 <p>Note Value = 4/1</p>	 <p>Longa Note</p>	 <p>Note Value = 4/1</p>	 <p>Longa Rest</p>
 <p>Note Value = 2/1</p>	 <p>Breve Note</p>	 <p>Note Value = 2/1</p>	 <p>Breve Rest</p>
 <p>Note Value = 1/1</p>	 <p>Whole Note</p>	 <p>Note Value = 1/1</p>	 <p>Whole Rest</p>
 <p>Note Value = 1/2</p>	 <p>Half Note</p>	 <p>Note Value = 1/2</p>	 <p>Half Rest</p>
 <p>Note Value = 1/4</p>	 <p>Quarter Note</p>	 <p>Note Value = 1/4</p>	 <p>Quarter Rest</p>
 <p>Note Value = 1/8</p>	 <p>Eighth Note</p>	 <p>Note Value = 1/8</p>	 <p>Eighth Rest</p>
 <p>Note Value = 1/16</p>	 <p>Sixteenth Note</p>	 <p>Note Value = 1/16</p>	 <p>Sixteenth Rest</p>
 <p>Note Value = 1/32</p>	 <p>Thirty-second Note</p>	 <p>Note Value = 1/32</p>	 <p>Thirty-second Rest</p>
 <p>Note Value = 1/64</p>	 <p>Sixty-fourth Note</p>	 <p>Note Value = 1/64</p>	 <p>Sixty-fourth Rest</p>
 <p>Note Value = 1/128</p>	 <p>Hundred twenty-eighth Note</p>	 <p>Note Value = 1/128</p>	 <p>Hundred twenty-eighth Rest</p>

上の図表を使って音価とその音価のブロックを閲覧してください。

2.2 ピッチのブロック

ピッチブロックは音価ブロックの中に使われています。ピッチブロックはピッチの名前とピッチのオクターヴを決めます。ピッチの名前とオクターヴの数値を組み合わせ、音符の振動（音波の振動）を決めます。



ピッチの名前の選ばれるブロックが色々あります。次に例えが説明してあります。

上のピッチブロックがソルフェージュブロックで決められています。そのピッチ・ブロックは'ソル'と'4オクターヴ'のインプットがあります。ソルフェージュの名前が「ド、レ、ミ、ファ、ソ、ラ、シ」から選ぶことができます。

その次のピッチ・ブロックの選ばれているピッチがピッチ・アルファベットブロックで決められています。そのピッチ・ブロックのインプットは'G'と'4オクターヴ'です。ピッチ・アルファベットの名前が「C D E F G A B」から選ぶことができます。

その次のブロックが音度ブロックで例の選ばれている数値が音階(スケールかモード)の五度でオクターヴが4の意味です。C == 1, D == 2, ... (<<=== is this still correct for our newer method of scale degree??? Please check!)

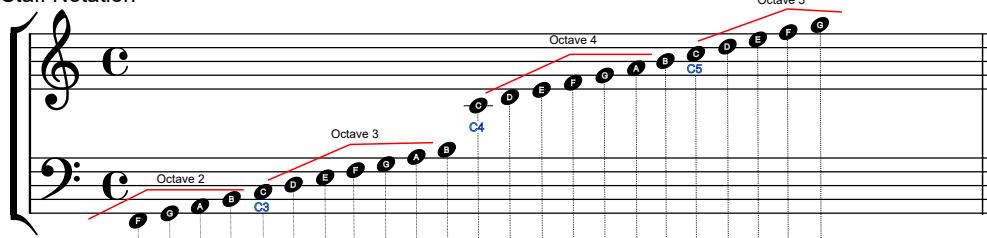
その次のブロックはピッチ数字ブロックでピッチが選ばれています。7のインプットで4オクターヴのC音符から7半音のピッチの意味です。ピッチ数字のゼロはどの絶対ピッチかピッチ数字のゼロを設定ブロックで決めることができます。

一番下にあるブロックのピッチはハーツブロックで決められています。ハーツ・ブロックは数字・算数の色々のブロックと一緒に使います。例のハーツ・ブロックの数値は392で音符が392ハーツのピッチで鳴らします。

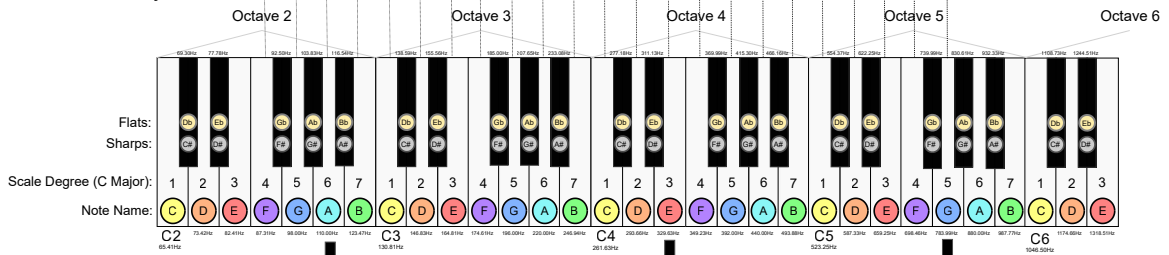
ハーツを使うとオクターヴのインプットがむしします。オクターヴの数値のインプットは全数の必要があります。

ピッチの名前は文事ブロックでもインプットができますよ。

Staff Notation



Keyboard



Using the *Hertz Block*
1 Hertz = 1 Hz = 1 Cycle per second

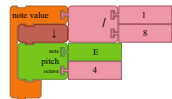


Using the *Pitch Block*

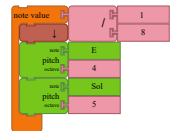


Using the *Solfege Block*

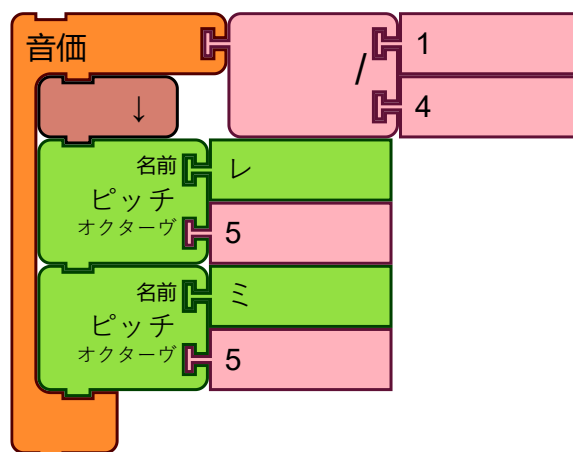
Use Individually to create a *Note*



Combine them to create a *Chord*

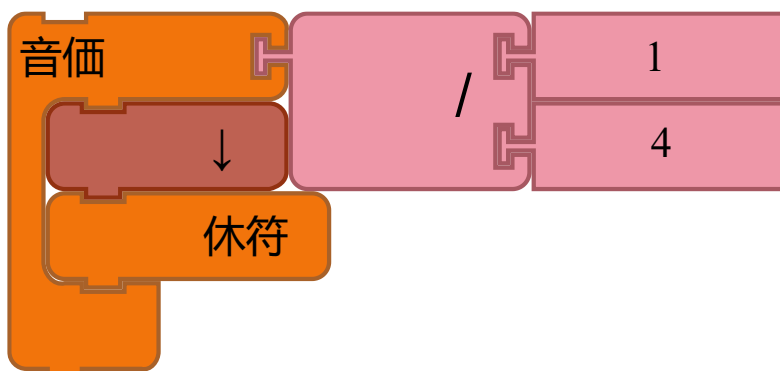


2.3 和音



和音(一つよりの一緒に鳴らされているピッチ)は一つの音価ブロックの中にピッチブロックを、上の例のように、一つ以上入れます。

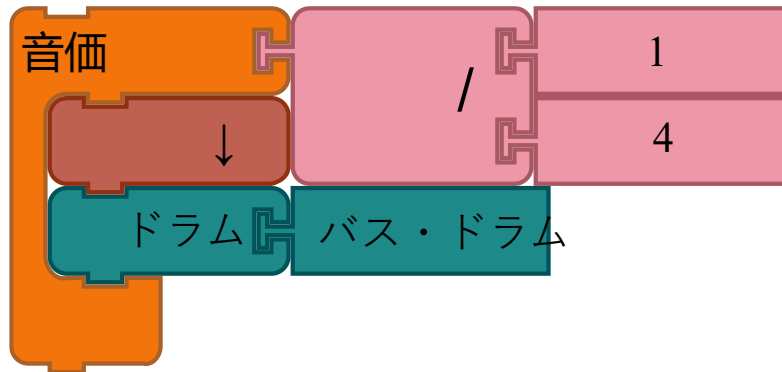
2.4 休符



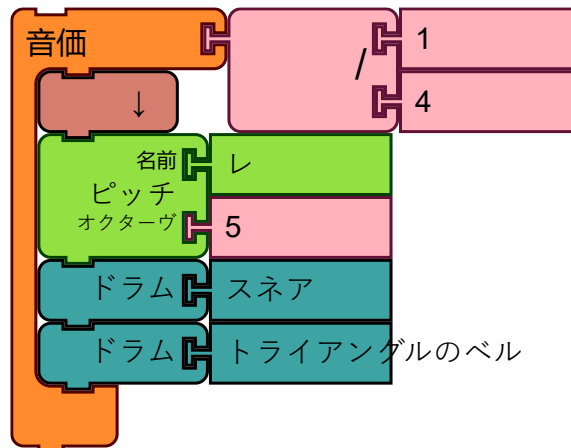
休符ブロックは音価ブロックの中に入れるとその音価の長さで入れられているネズミが音鳴らしを休みます。

ピッチブロックを出すと自動的に休符が現れます。

2.5 ドラム



ピッチブロックのにも使える場合と同じでドラム・ブロックがピッチ・タイム行列か音価ブロックの中にも使えます。今、24個ぐらいのドラムの音の中から選ぶことができます。デフォルトのドラムがバス・ドラムです。



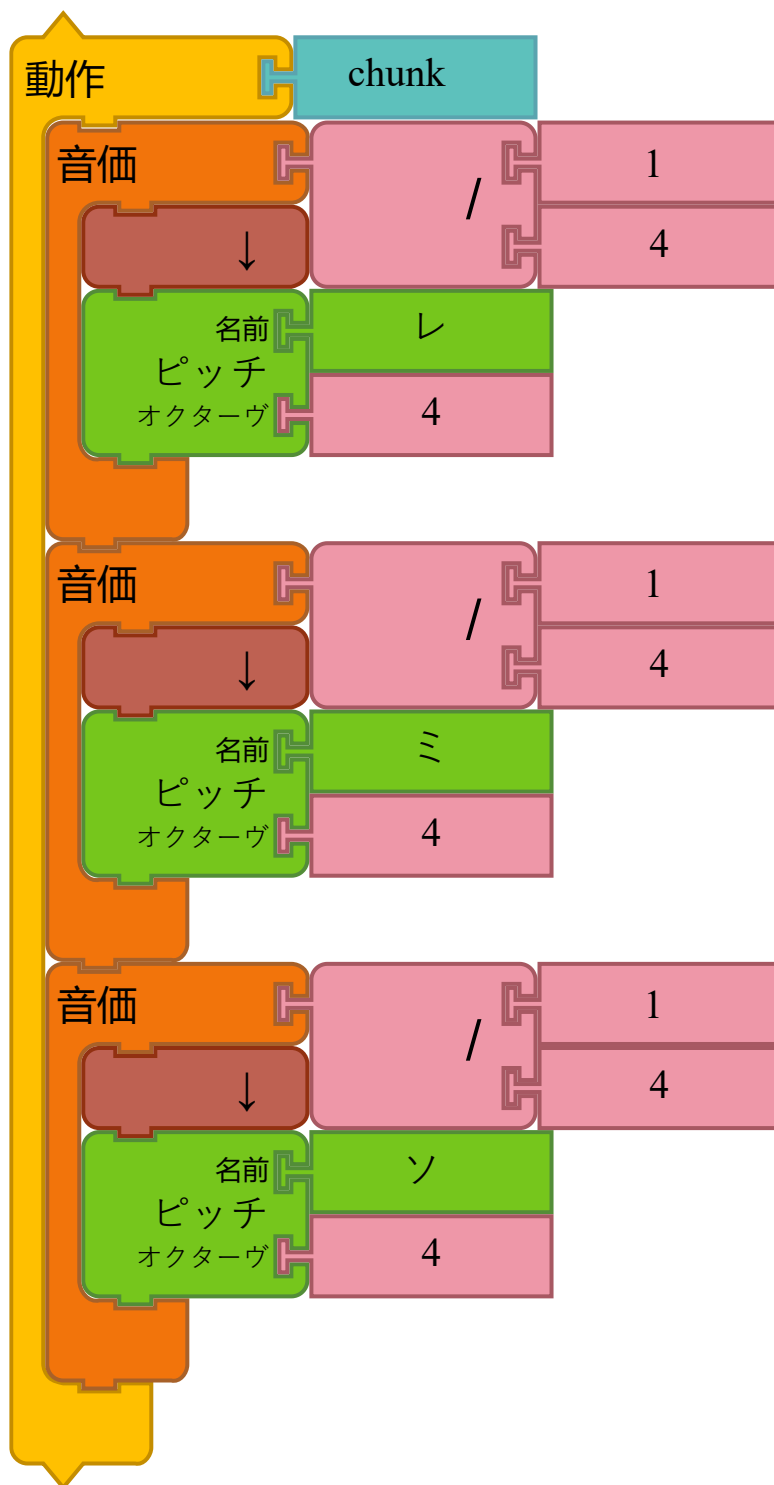
この上の例のように和音一つよりのドラムブロックと一緒に使えます。一つの音価ブロックの中に一つよりのドラム・ブロックとピッチブロックとお好み組み合わせせて音の楽しみができます。

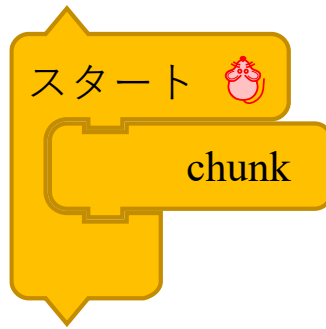
3. 音楽でプログラミング

[前のトピック \(2. 音の関係\)](#) | [目次に戻る](#) | [次のトピック \(4. ウィジェット\)](#)

このセクションのトピックはチャンクで動作ブロックを使って音楽とプログラミングができます。チャンクが自分で動作ブロックを使って作ることも、ピッチ・タイム行列を使って作ることもできます。

3.1 チャンク





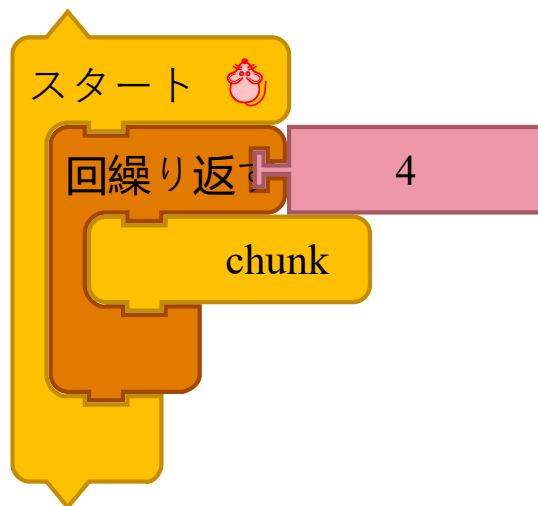
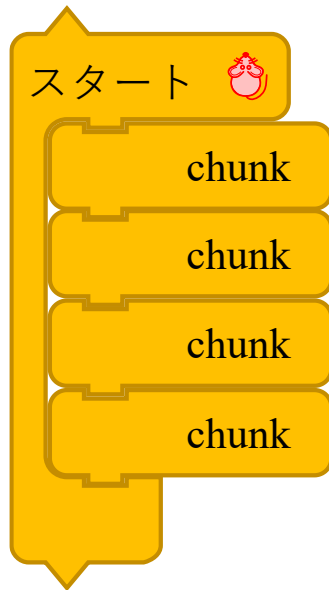
新しい**動作**ブロックをパレットから引く時、自動的に新しいブロックが作られます。その新しいブロックが**動作**パレットの一番上に現れます。新しいブロックがパレットから引かれたブロックのコードを**絶対的に**読んで**実行**します。ですから新しいブロックのコード・スタックをクリックしても引かれたブロックをクリックしても**効果が同じ**です。パレットから引かれるデフォルトの**名前**が **チャンク**, **チャンク1**, **チャンク2** ... です。そのデフォルトの**名前**が**好きに変える**ことができます。いくつもパレットからお好み引いて使えます。

動作の色々のブロックが入れられているブロックを読んで**実行**するためです。**動作**ブロックはいつも呼ばれた時にしか**実行**しません。例えば、**スタート**・ブロックの中にあるのが**実行**のボタンが押された時にしか**実行**しません。自分のコードと**音楽**を整理するために**すぐ便利で約に立ちます**。

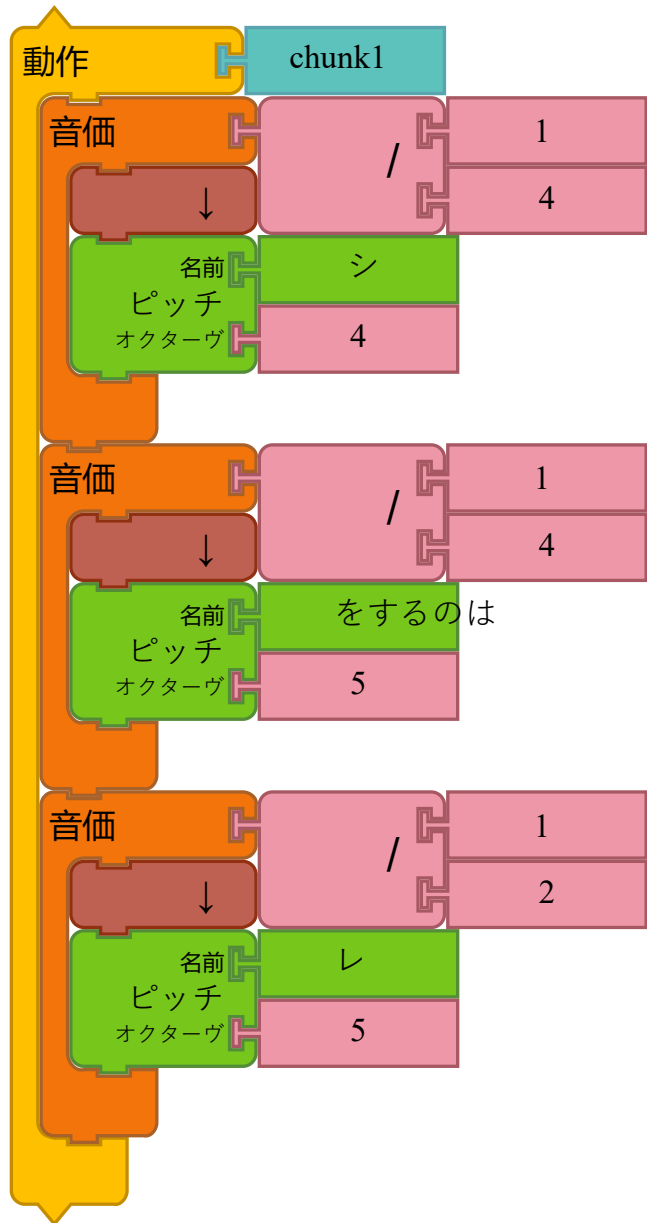
スタートブロックは**動作**ブロックの一つの特別な種類です。**実行**のボタンをクリックするとすべての**スタート**・ブロックの中に入っているコードが**実行**されます。**スタート**・ブロックはプロジェクトの**初め**でしょうか。

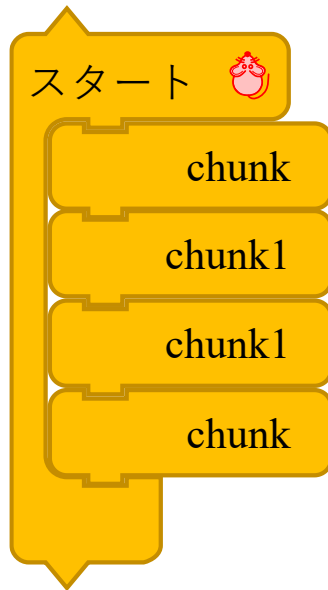
自分のプロジェクトを**実行**するために、いろんな**方法**がありまして画面の上左の**実行**ボタンを押すと**三つの実行の早さ**のオプションがあります。一般押しと早く**実行**、少し長く押しとゆっくりなペースで**実行**、もと長く押しと**音楽**はゆっくりに動く。

上の例に**チャンク**ブロックは**スタート**ブロックの中で**実行**のボタンを押すと**スタート**ブロックの中に入ってるコード（この例で**チャンク**ブロック）が**読まれて実行**します。この例のブロックを**変えた**かったら、**スタート**ブロックにあるものを**変える**だけです。

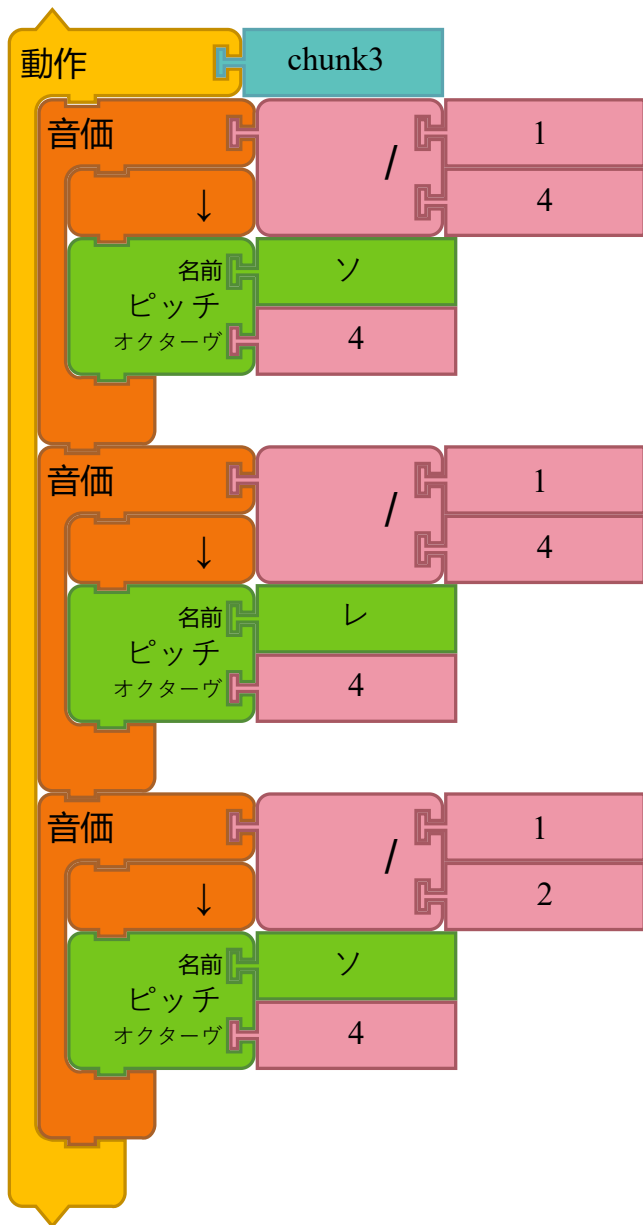
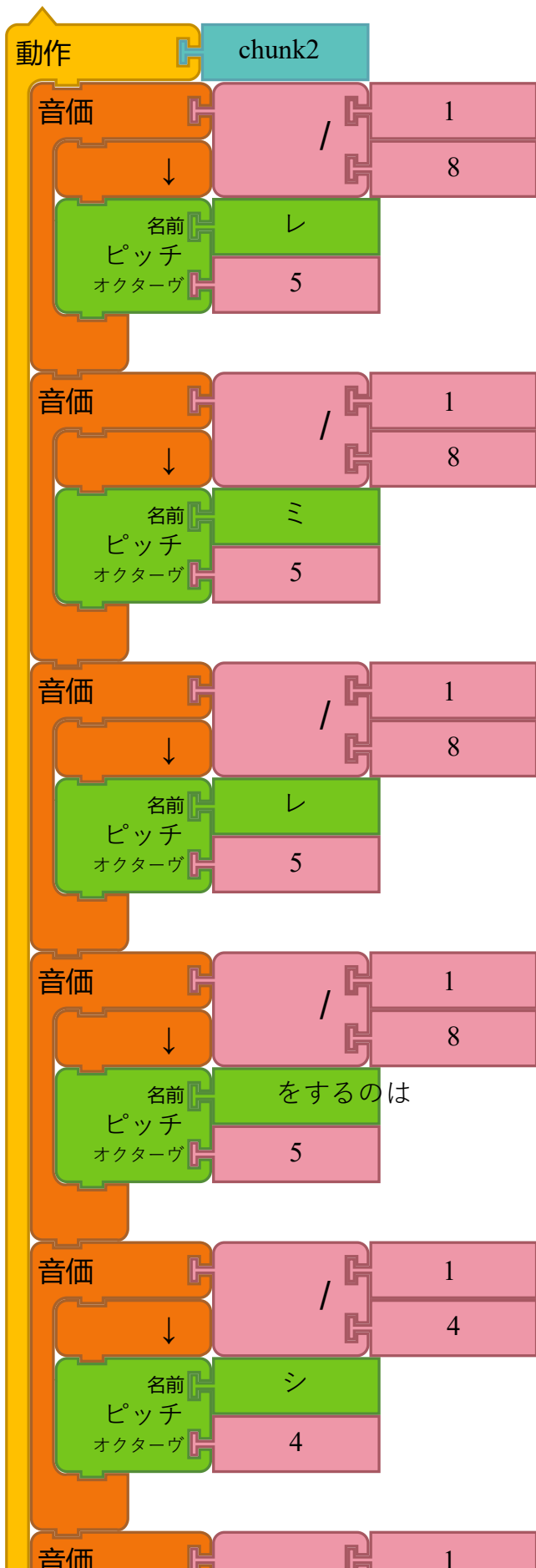


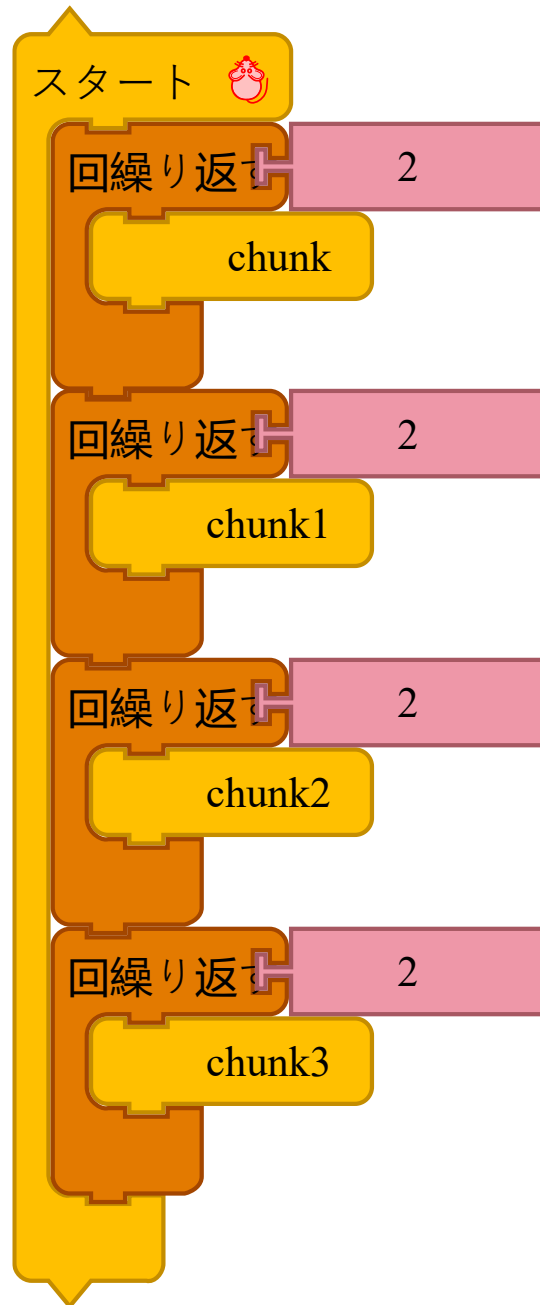
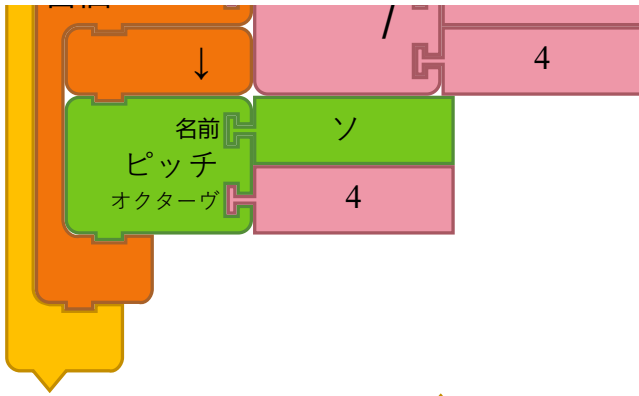
繰り返すチャンクを作る方法が色々あります。一つ以上のチャンクブロックを一つ一つ使うか繰り返すブロックを使う方法もあります。





チャンクの順も変えて効果が変わります。上の例で最初に"チャンク"を弾いて、次に"チャンク1"二回で、最後に"チャンク"をまた弾きます。



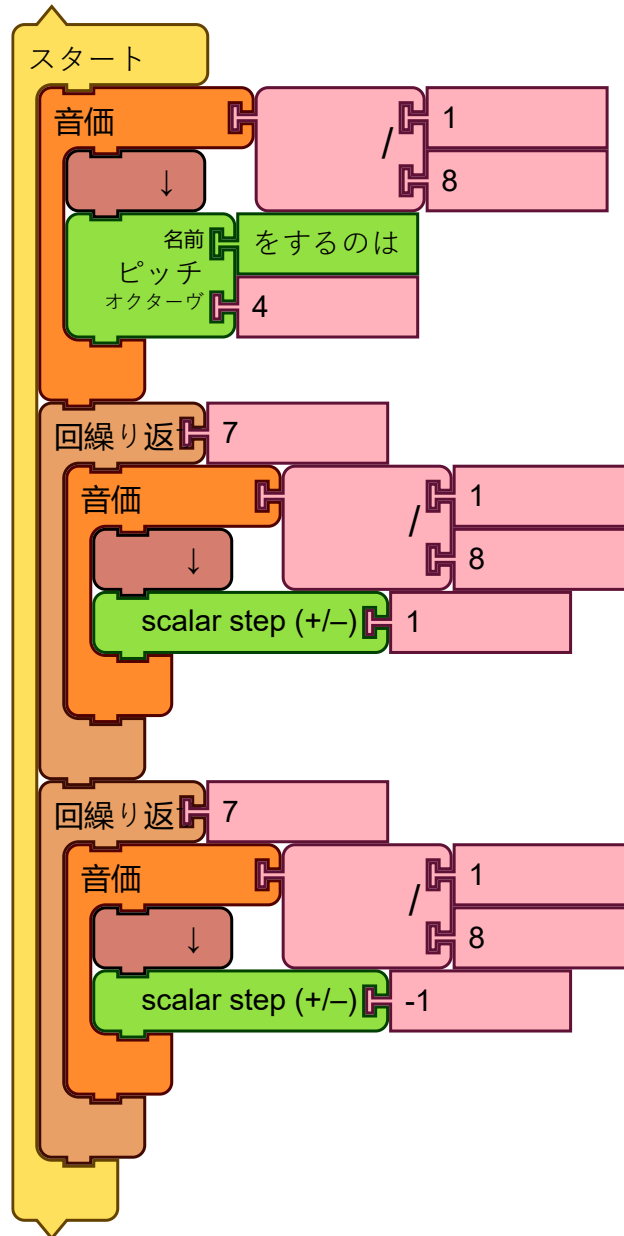


チャンクを組み合わせて音楽が作れます！上のメロディーご存知ですか？（ミュージック・ブロックのコードを読んでチャレンジしてみてください）

3.2 音楽的の転化

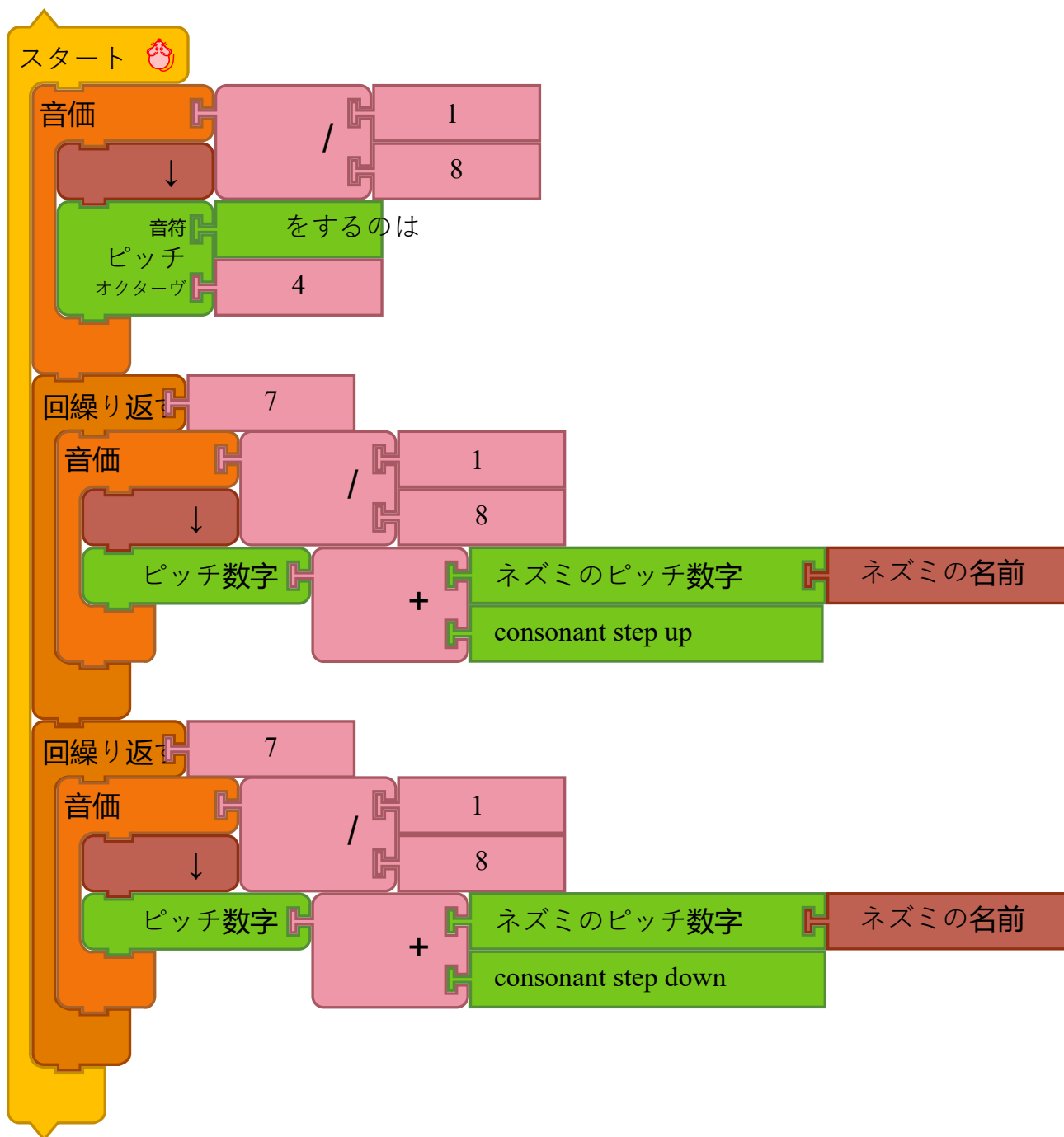
ピッチとリズムを転化する方法がいっぱいあります。次に読んで、やってみましょう。

3.2.1 音符動きのブロック



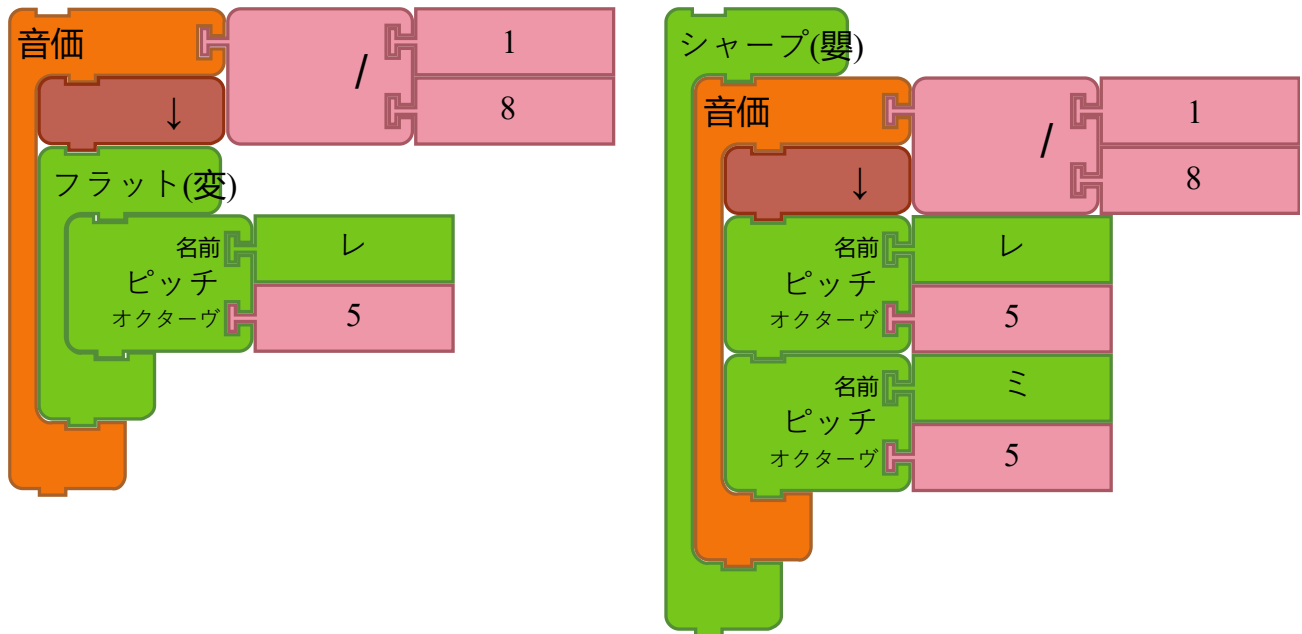
ステップピッチブロックはメロディーの音符を音階的に上/下に動かすことができます。上の例にステップピッチブロックが回繰り返すブロックの中でその入ってるコードが7回繰り返して、音階の音程で音符が上に上がって、下に下ろし動きます。

ライブで再生 (<https://musicblocks.sugarlabs.org/index.html?id=1523032034365533&run=True>)



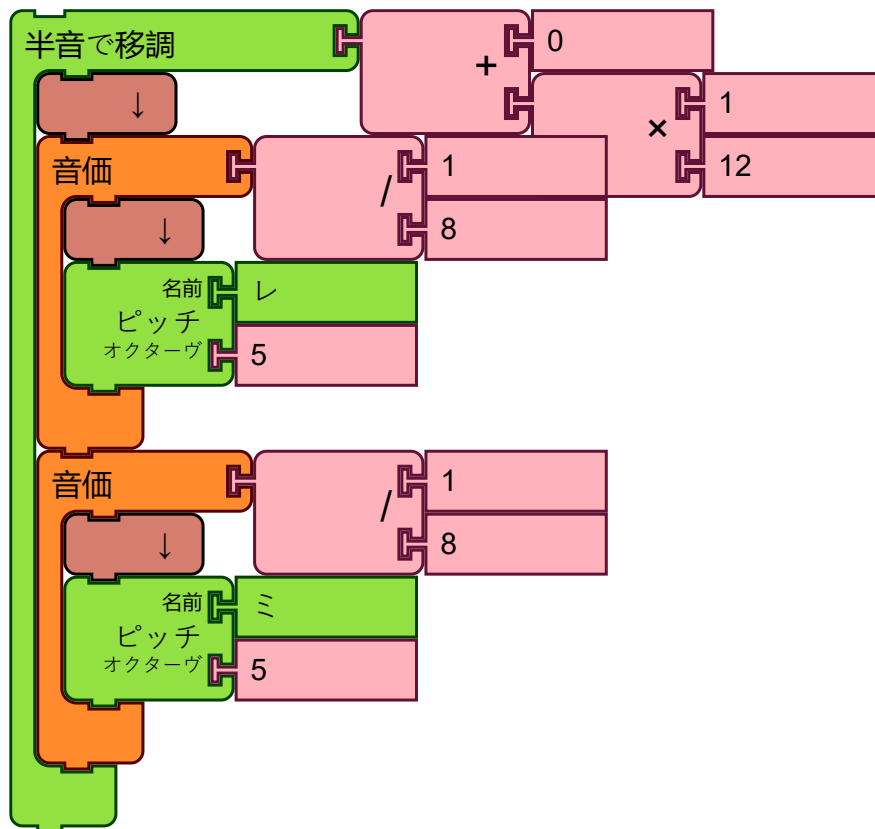
Another way to move up and down notes in a scale is to use the *Consonant Step Up* and *Consonant Step Down* blocks. These blocks calculate the number of half-steps to the next note in the current mode. (You can read more about [音楽の音階とモード](#) below.) Note that the *Mouse, ピッチ, 数字・算数* block returns the *ピッチ* number of the most recent note played.

3.2.2 シャープ(嬰)とフラット(変)

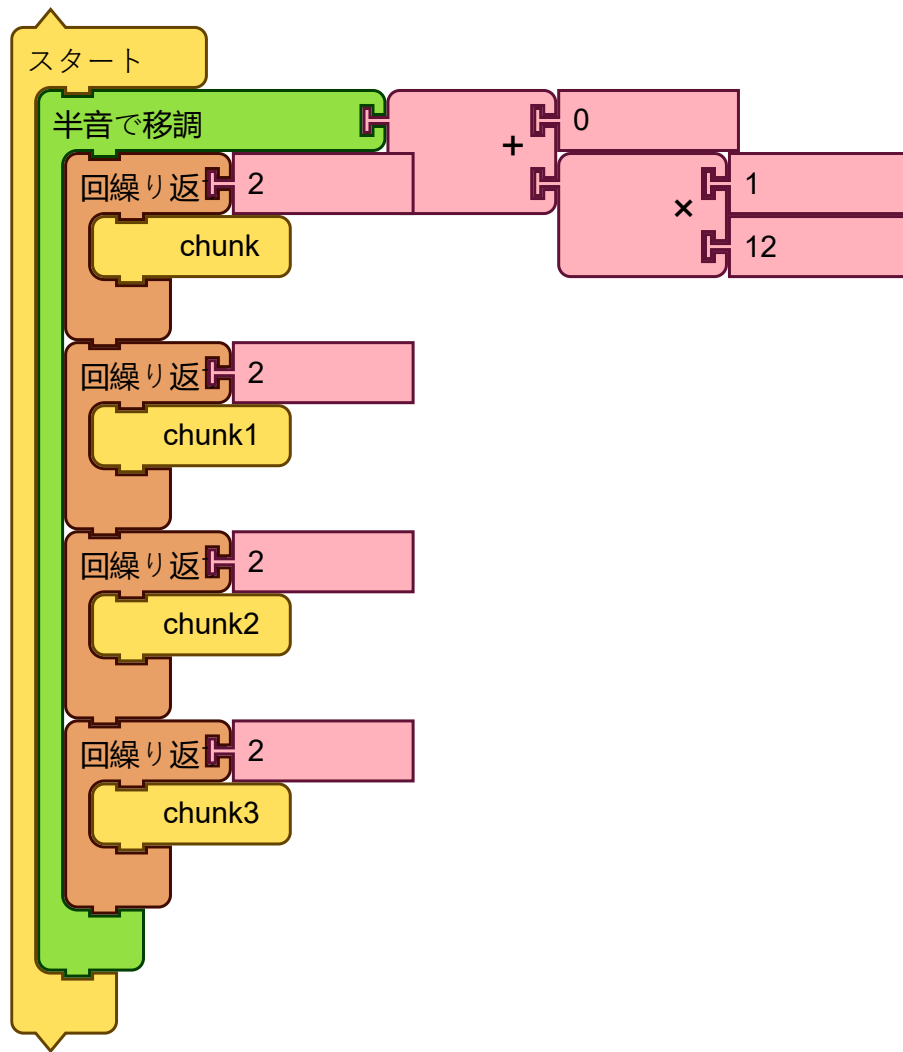


♯は日本語で嬰 (えい)、♭は変 (へん) です。シャープ (嬰) とフラット (変) のブロックはピッチブロックか音価ブロックかチャンクを巻いてピッチを転化することができます。一つのシャープ (嬰) ブロックを使ってピッチが半音上で変えます。一つのフラット (変) ブロックを使ってピッチが半音下で変えます。左の例ピッチブロックが'ミ'でフラット (変) の訳でピッチが半音下変えます(ミ♭になります)。右の例、二つのピッチブロックも (和音) が両方のピッチも半音上変えます。

3.2.3 移調をアジャスト

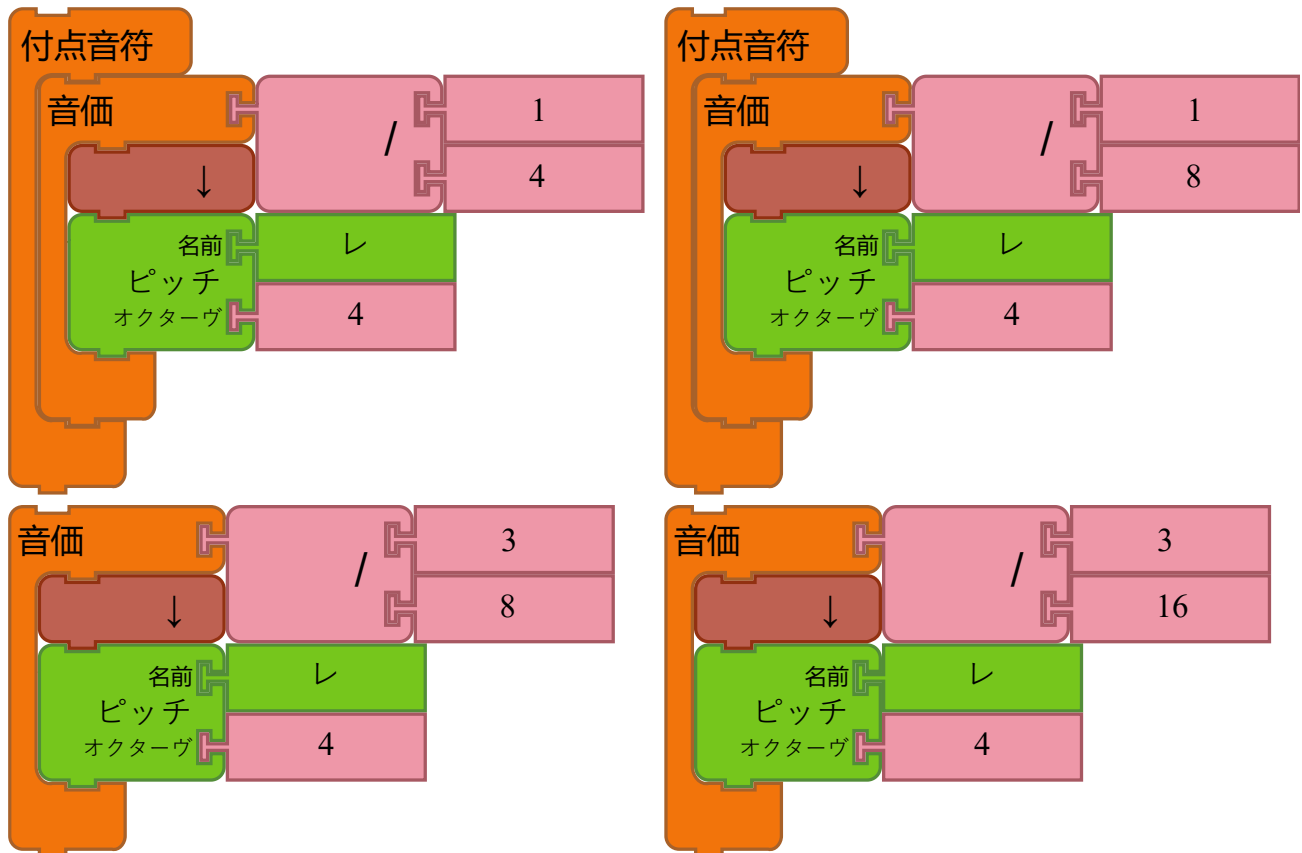


移調をアジャストブロックは巻かされているピッチ・ブロックで選ばれているピッチを半音の数値で変えることができます。整数のインプットはピッチを高く転調し、陰性のインプットはピッチを低く転調します。インプットイは全数の必要があります。オクターヴを転調するために、12のインプットは1オクターヴ上に転調し、-12のインプットは1オクターヴ下に転調します。



上の例にこの前の例のメロディーを転調をアジャスト・ブロックで1オクターヴに転化します。

3.2.4 付点音符 (ドット)



"付点音符"のリズムの音符が付点音符ブロックでできます。付点音符の音符はその入っている音価を50%で増します(50%で増します)。例えば、付点音符の4分音符が三つの8分音符の長さで鳴らします。計算は $1/4 + 1/8 = 3/8$ 。付点音符の8分音符が3/16の音価で ($1/8 + 1/16$)鳴らします。

付点音符を使う代わりに音価の数値を変えて付点音符と同じ長さもできますよ。例えば、4分音符の符点音符の長さが欲しかったら、 $1/4$ の代わりに $3/8$ のインプットをして4分音符の付点音符と同じ長さで鳴らします。


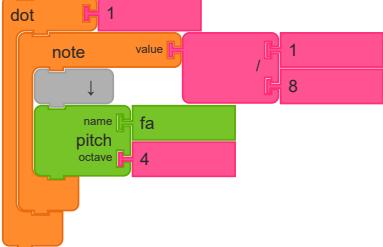
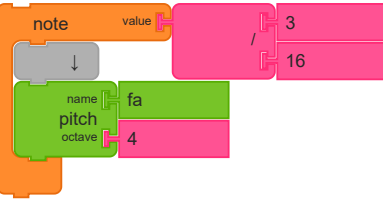
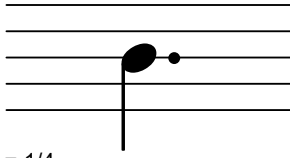
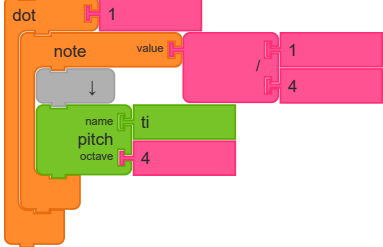
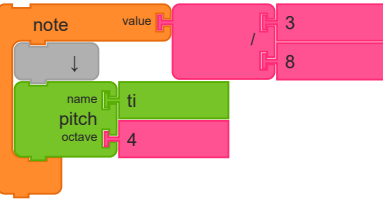
Using Dotted Notes

The dot increases the value of a note by half of its value.

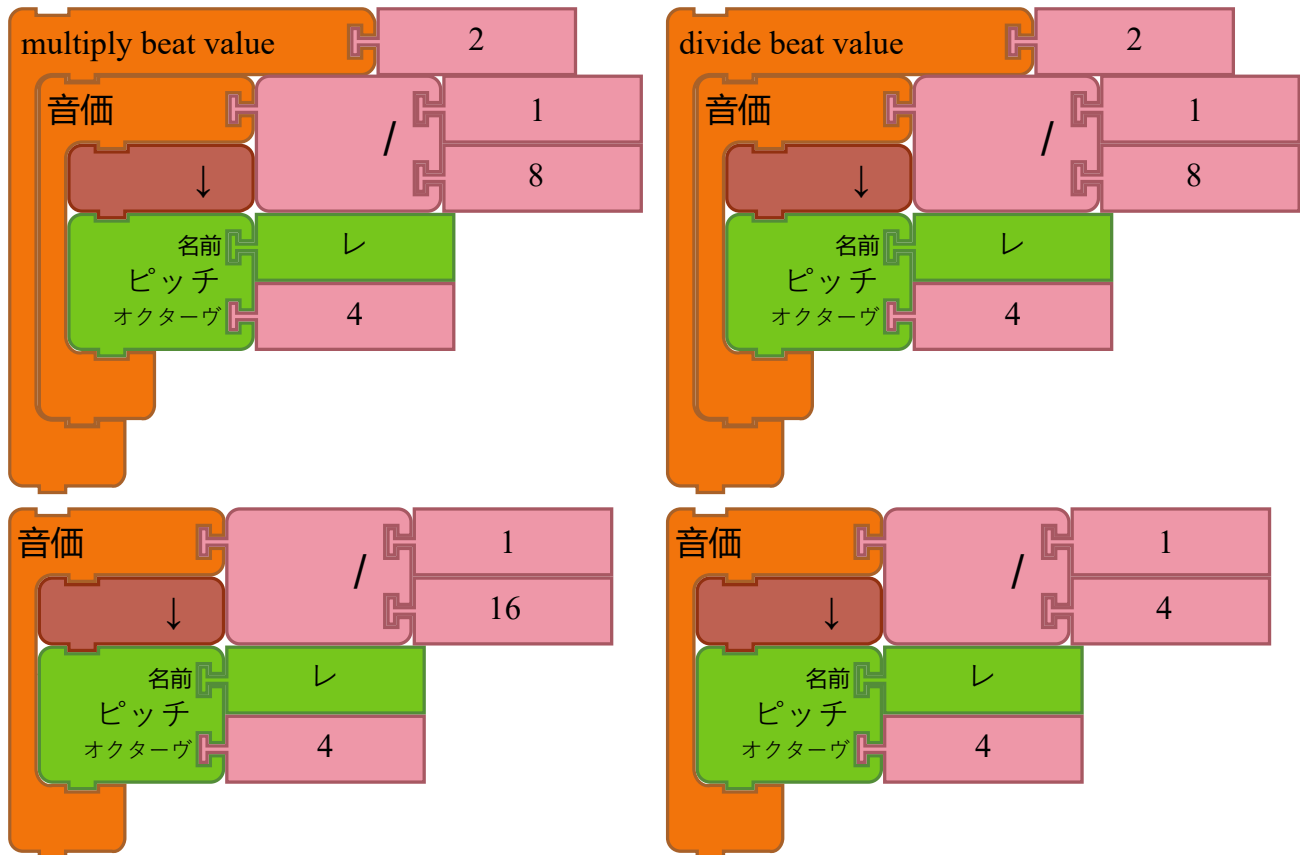
x= value of note

$$\text{Formula: } x + \frac{x}{2} = \text{value of dotted note}$$

Examples:

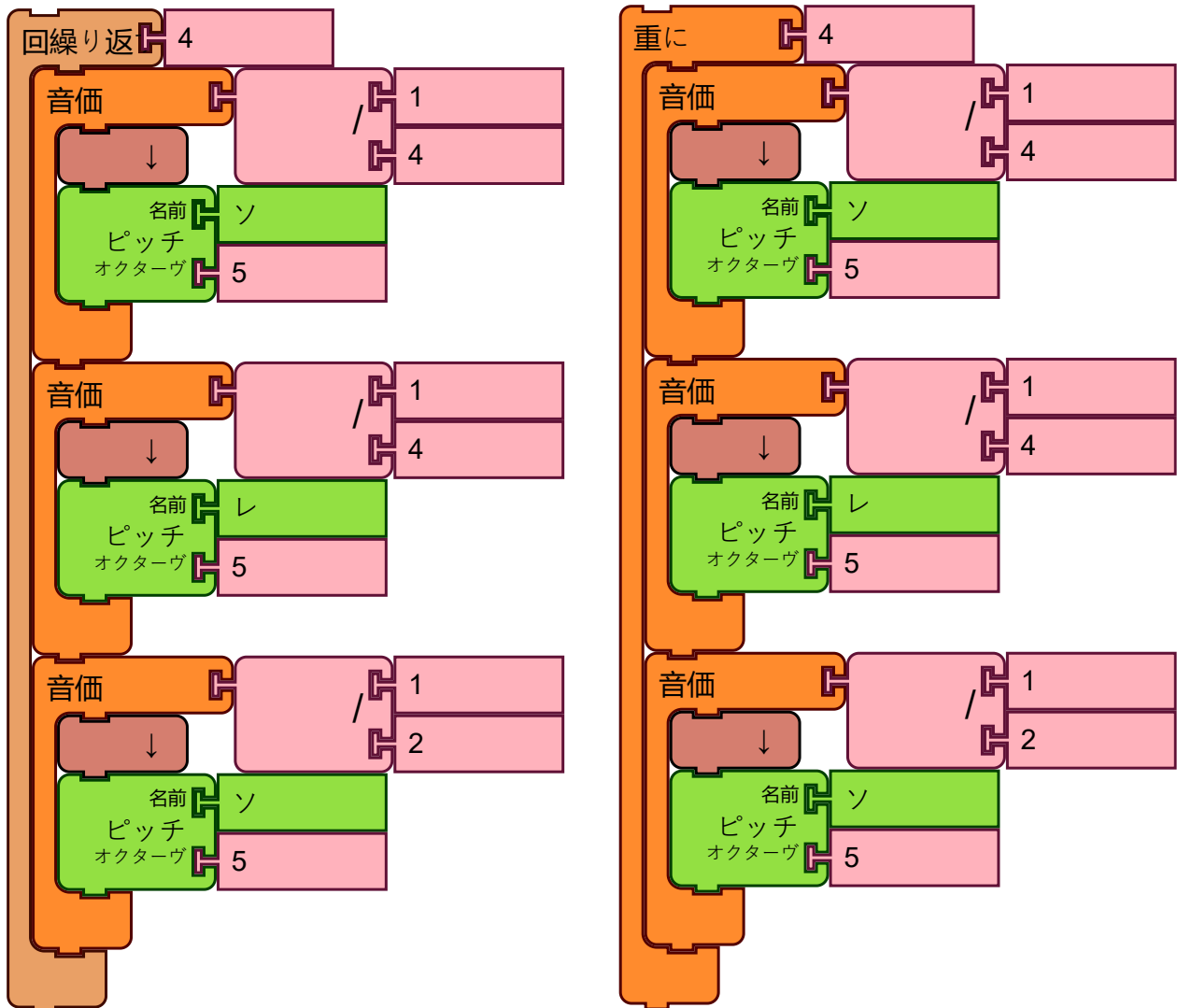
Western Notation	Music Blocks Notation with dot	Music Block Notation without dot
 <p>For x = 1/8,</p> $\frac{1}{8} + \frac{1}{(8 \cdot 2)} = \frac{1}{8} + \frac{1}{16} = \frac{2}{16} + \frac{1}{16} = \frac{3}{16}$		
 <p>For x = 1/4,</p> $\frac{1}{4} + \frac{1}{(4 \cdot 2)} = \frac{1}{4} + \frac{1}{8} = \frac{2}{8} + \frac{1}{8} = \frac{3}{8}$		

3.2.5 演算で音価をかけ割ること



You can also multiply (or divide) the beat value, which will speed up or slowdown the notes. Multiplying the beat value of an 1/8 note by 2 is the equivalent of playing a 1/16 note. Dividing the beat value of an 1/8 note by '2' is the equivalent of playing a 1/4 note.

3.2.6 回繰り返り音符

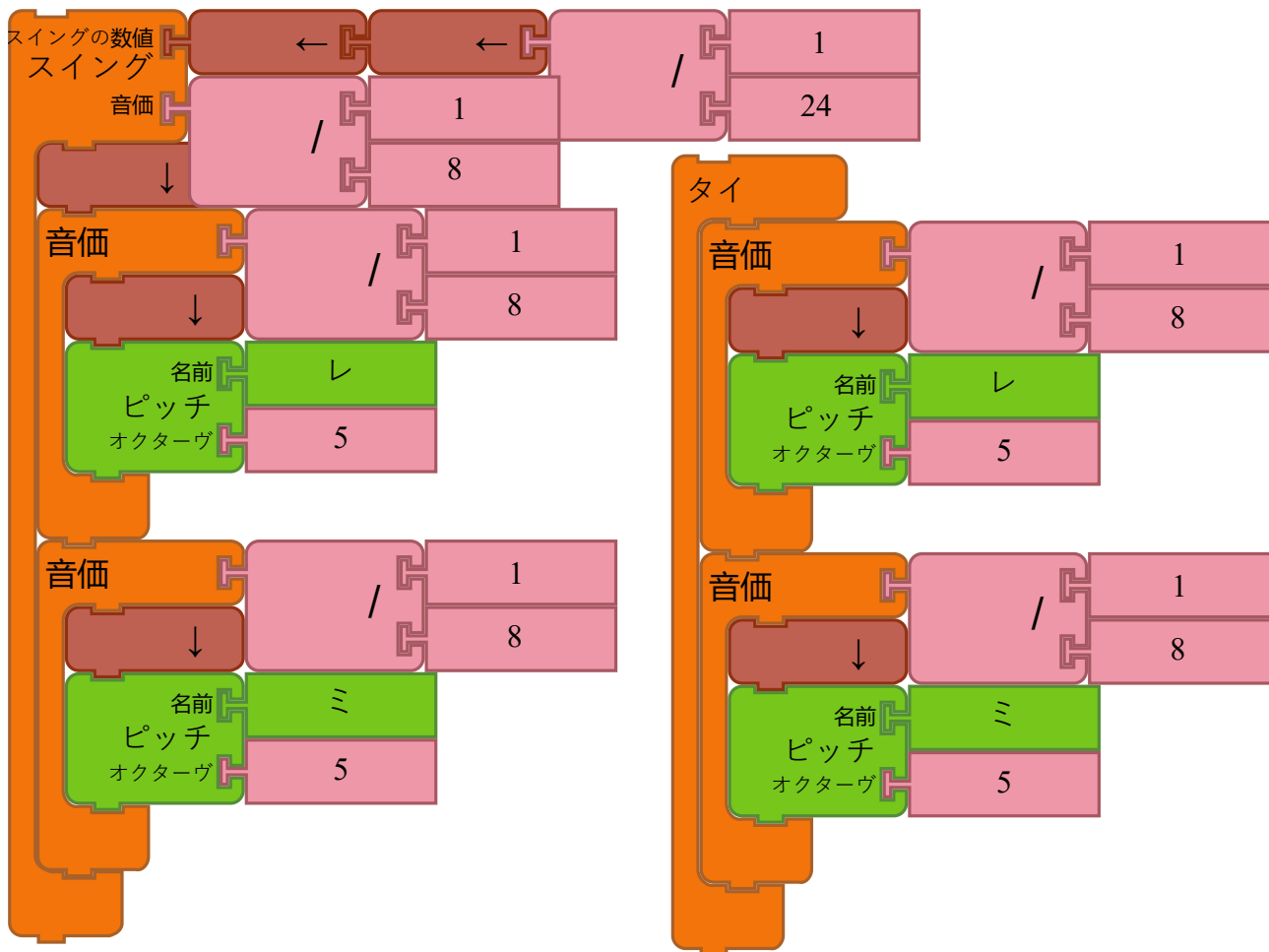


音符を回繰り返す方法が色々あります。回繰り返すブロックははっている音符をインプットの数値で繰り返します。

重にブロックは入っている音符をすべてインプットの数値で重します。

左の例に結果がソ, レ, ソ, ソ, レ, ソ, ソ, レ, ソ, ソ, レ, ソで;右の例の結果がソ, ソ, ソ, ソ, レ, レ, レ, レ, ソ, ソ, ソ, ソです。

3.2.7 スイング・リズムとタイの音符



スイングブロック音符対ずつで使えます。その音符の対の先が音価をスイングのインプット数値で増して、対の次の音符の数値からスイング・ブロックの数値を引きます。

対の音符の音価は同じじゃないのスイング・ブロックは効果がありません。

上の例でレ 5 は $1/6$ の音価で鳴らして、ミ 5 は $1/12$ の音価で鳴らします。 ($1/8 + 1/24 === 1/6$ and $1/8 - 1/24 === 1/12$)。

対の音符の音価の合計は変わっていませんよ。

タイも音符対ずつで使えます。タイはそのタイ・ブロックの中に入っている音符の音価をすべて組み合わせてタイされている音価の合計で鳴らします。

タイ・ブロックに入っている音符のピッチは同じじゃないと効果がありません。タイ・ブロックに入っている音符の音価は違ってOKです。

Using Notes with Ties


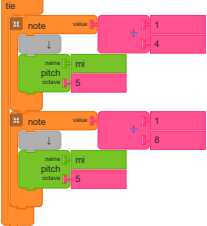
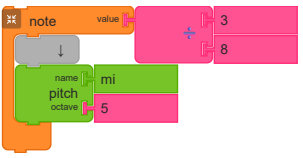

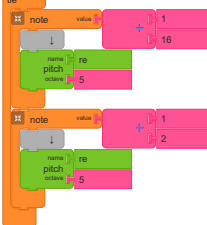
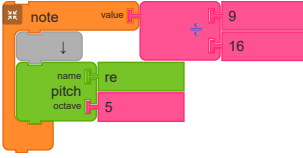
A tie connects two notes of the same pitch* and indicates that they are to be played as the sum of the two notes.

x= value of note 1

y= value of note 2

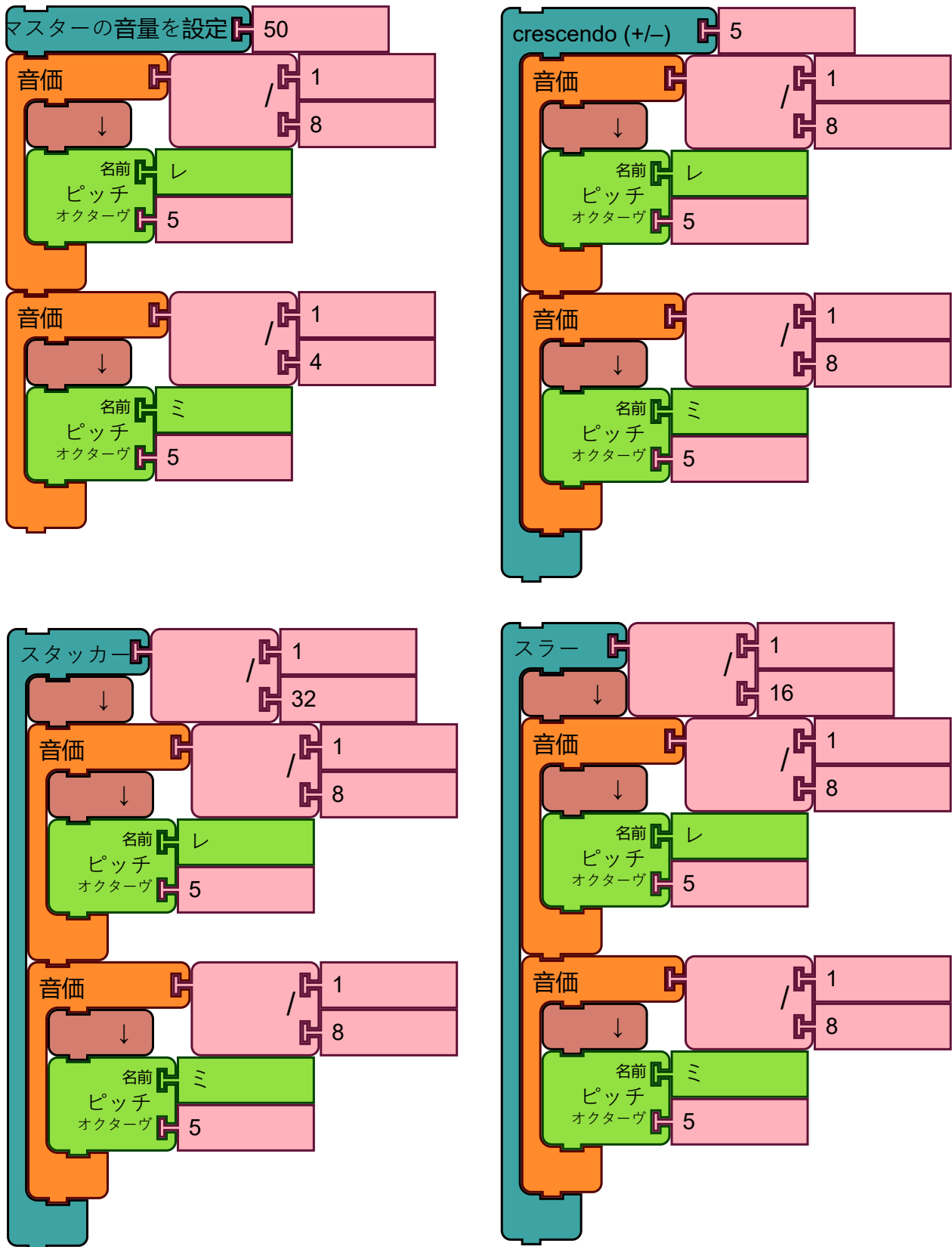
Formula: $x + y =$ total value of notes contained within tie

Examples:

Western Notation	Music Blocks Notation with tie	Music Block Notation without tie
 $\frac{1}{4} + \frac{1}{8} = \frac{3}{8}$ <p>Find common denominator: $x = \frac{1}{4} \quad 2 * \frac{1}{4} = \frac{2}{8} + \frac{1}{8} = \frac{3}{8}$ $y = \frac{1}{8}$</p>		
 $\frac{1}{16} + \frac{1}{2} = \frac{9}{16}$ <p>Find common denominator: $x = \frac{1}{16} \quad 8 * \frac{1}{16} = \frac{8}{16} + \frac{1}{16} = \frac{9}{16}$ $y = \frac{1}{2}$</p>		

* Ties affect rhythm, not pitch. For tie to work, both pitches must be exactly the same. If not, it will be considered a slur.

3.2.8 音量、クレシェンド、スタッカート、スラーのブロックの関係



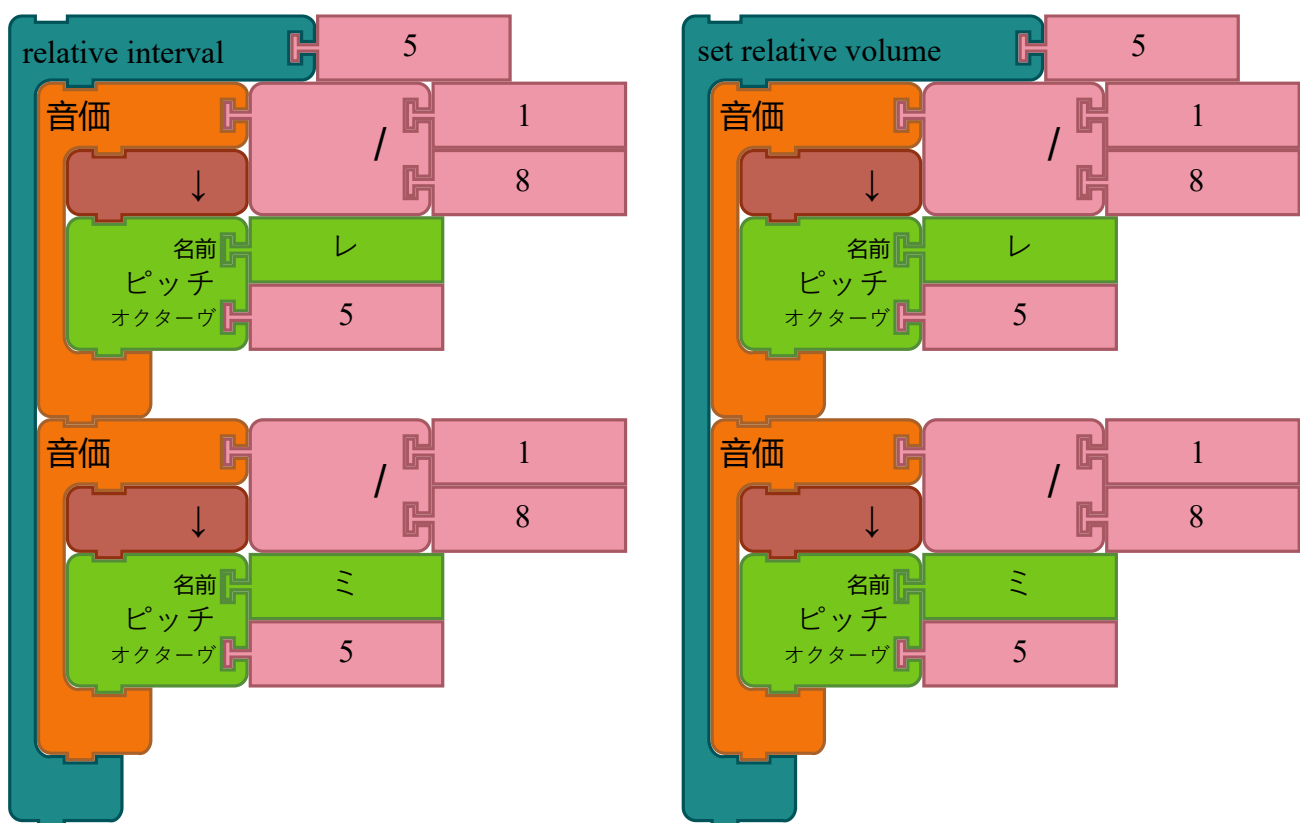
音量を設定ブロックは音符の鳴らす音量を変えます。デフォルトが50で最低限が0(絶対静か)で最大限が100(一番大きい音量)です。

クレッシェンドブロックは巻かされている音符の音量を小さく大きくします。例えば、5の数値のクレッシェンドブロックに三つの音符ブロックが入っていたら、最後の音符の音量が最小の音符の音量より15%大きいです。

スタッカートブロックは音符の音価を変えなく音符の鳴らす長さをスタッカートの数値で短くします。

スラーブロックは音符の音価を変えなく音符の鳴らす長さをスラーの数値で長くします。音楽でレガートとも言います。

3.2.9 音程と相対的な音量の関係



相対的な音程は入っている音符から音程を計算してそのピッチも同時に鳴らします。例えば、音符から五度上の音も欲しかったら、このブロックを使います。上の例には、レに'ラ'をミに'シ'を同時に鳴らします。

相対的な音量を設定ブロックは入っている音符の音量をブロックの数値で足すことか引くことをします。例えば、100の数値は現在の音量を倍にします。

絶対音程

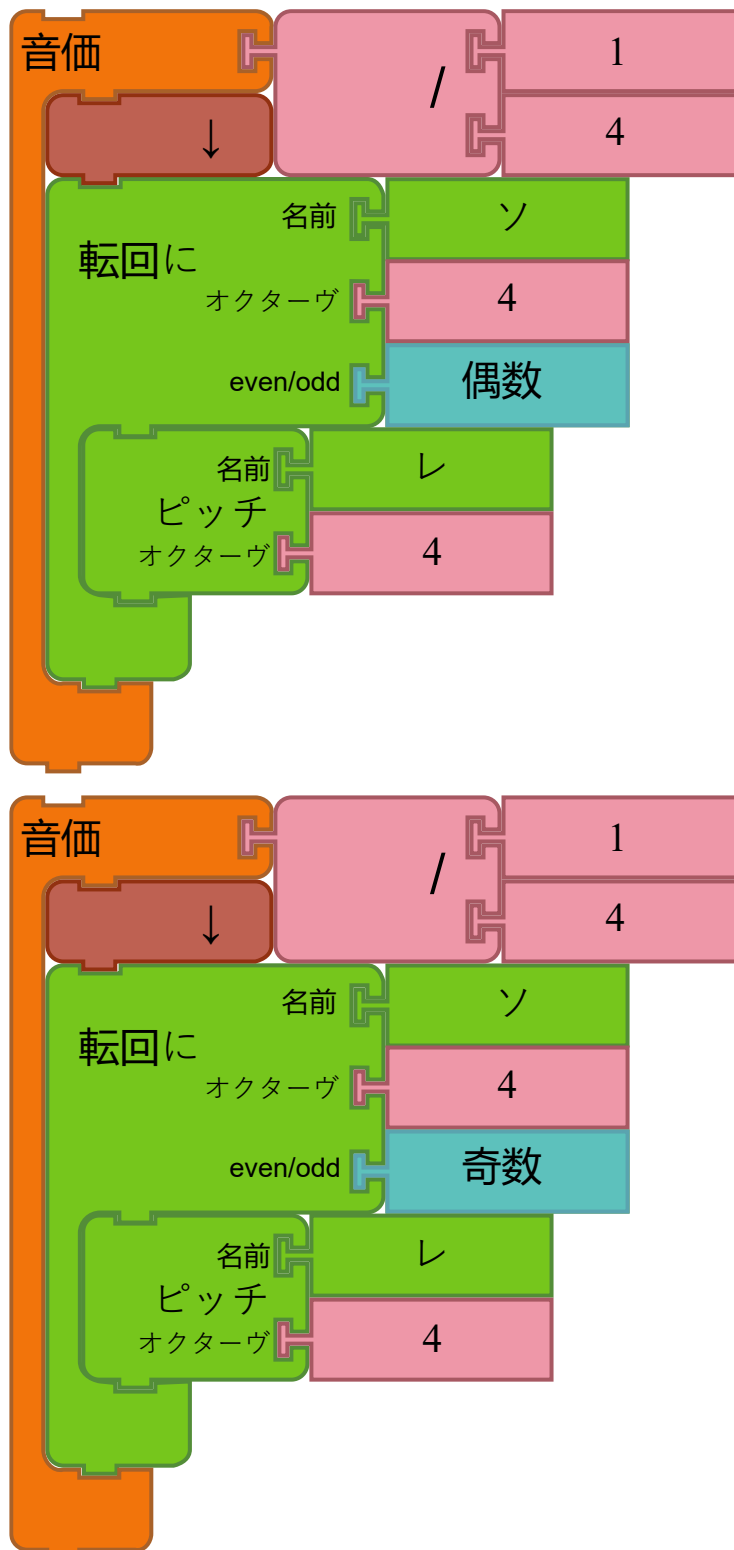
 alt tag

増ブロックが絶対音程を計算します。例えば、「像5程度」(<===check this!!) はインプットのピッチふあら像5程度のピッチの音符をインプットのピッチと一緒に鳴らします。同様に、短のブロックはインプットのピッチから絶対の音程を計算して一緒に鳴らします。例えば、短のブロックを使えばインプットのピッチから短のピッチも一緒に鳴らします。他の絶対音程のブロックは完全のも、減のも、長のもあります。

上の像5音程の例で二つの音符のインプットから四つの音符が鳴らして、D_5とA_5和音の音符も一緒に鳴らします。その後、E_5とC_5のピッチの音符も和音として鳴らします。短3音程の例にはオクターヴの転化ともまず、D_5とF_5の音符が鳴らして、次にE_5とG_6の和音が鳴らします。

(how to say double stop?? Also correct in English. A chord is usually 3 notes or more)

3.2.11 音符転回

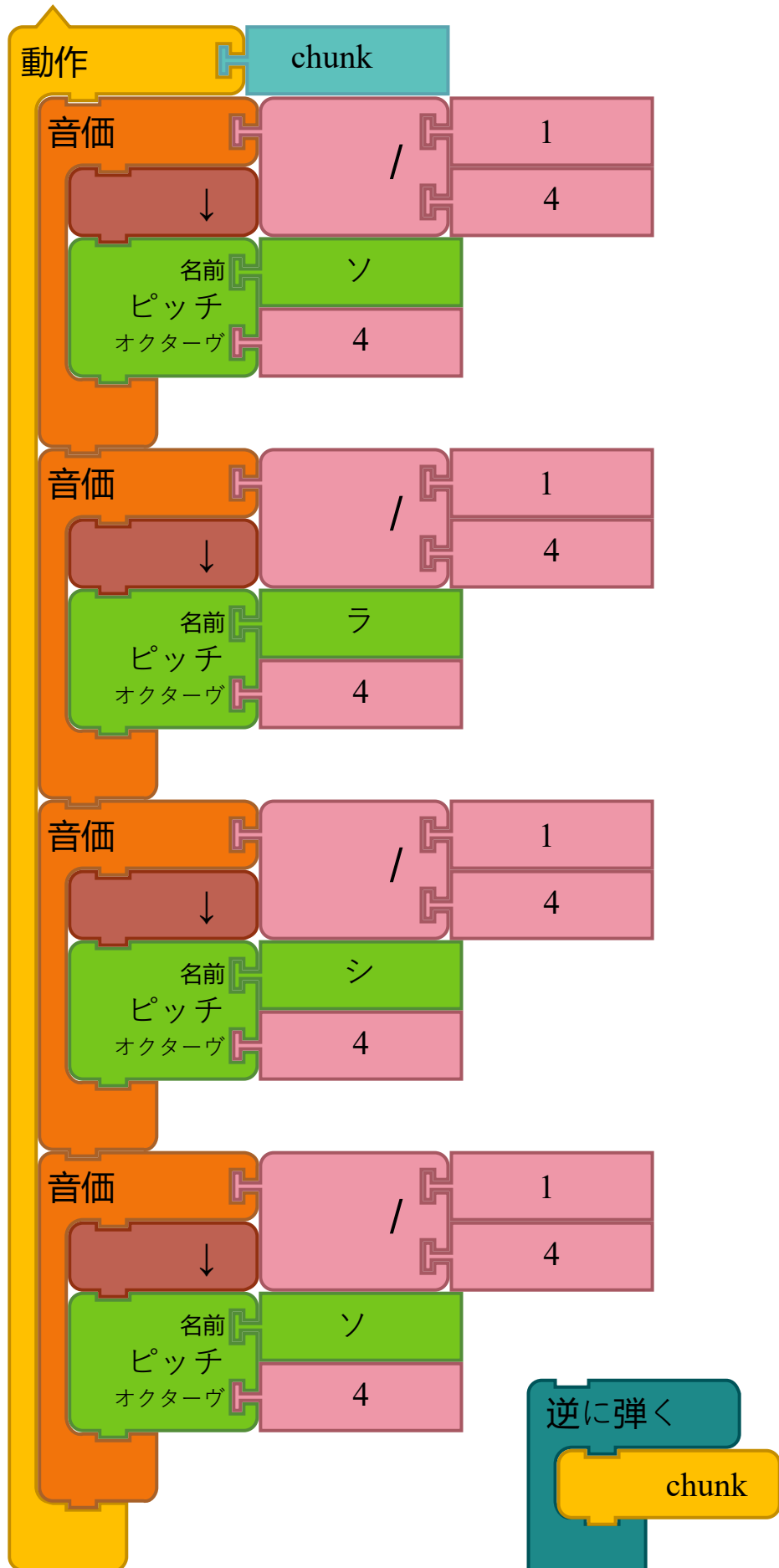


転回のブロックはmelodyの音程的にそのインプットの音符のピッチを転回します。転回ブロックで二つの種類があります。奇数が選ばれていると転回ブロックのピッチのインプットの回りで転回します。偶数が選ばれていると転回ブロックのピッチのインプットから半音の半分上のピッチからを回して転回します。

(take another look at English as well -- we should have a picture/chart for this as well)

上の転回 (偶数) の例で、G₄ を回して転回して D₄ のピッチのインプットが入っていて G₄ のピッチが結果で鳴らします。上の転回 (奇数) の例で D₄ のピッチが G₄ と G_{#4} の真ん中のピッチの回りで転回されて C_{#5} のピッチの結果が出ます。

3.2.12 逆に

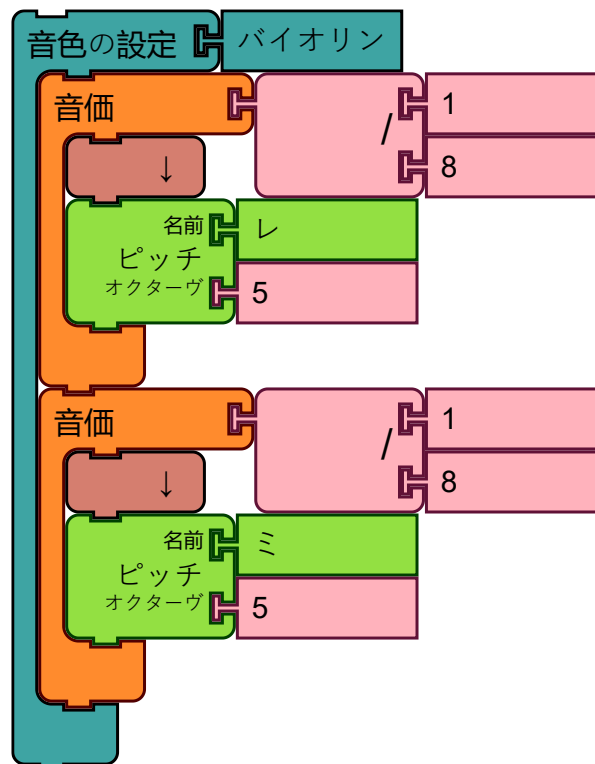


逆に再生ブロックは入られている音符ブロックをすべて逆に再生します。上の例に In the example above, the notes in チャンクに入っている音符がソ, シ, ラ, ソの順番で再生します(下から上のように逆に再生します)。

ライブで再生 (<https://musicblocks.sugarlabs.org/index.html?id=1522885752309944&run=True>)

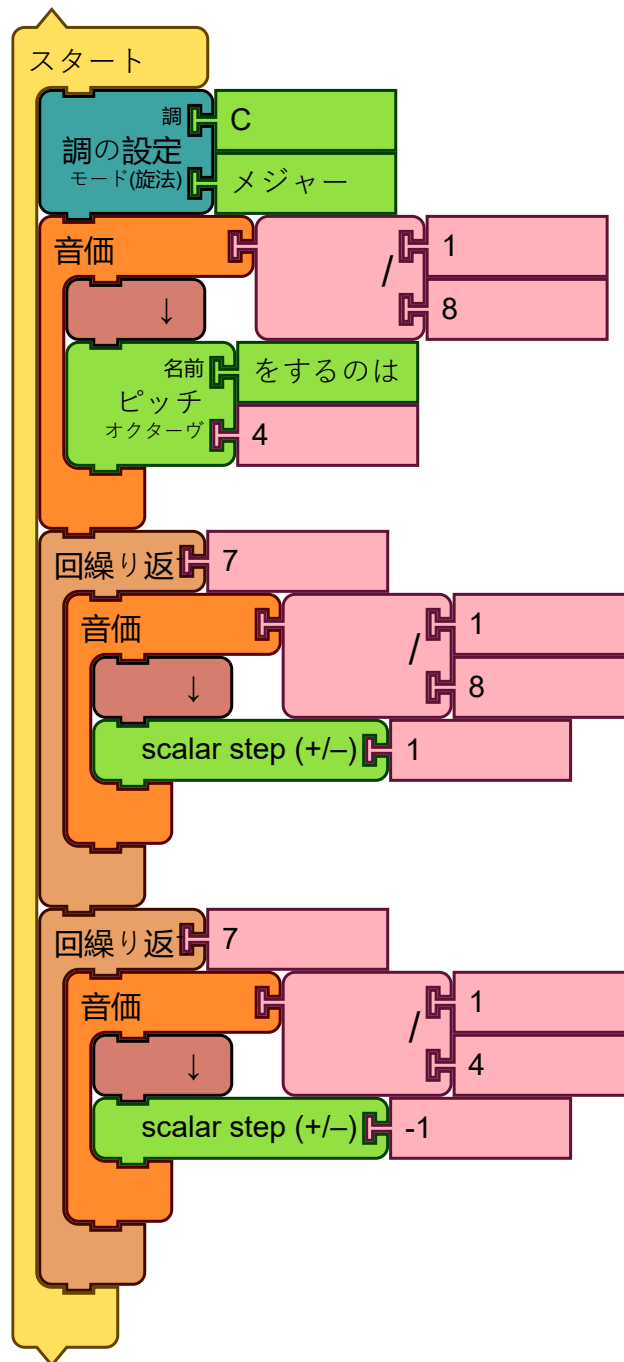
逆に再生ブロックの中に入っているのすべてのブロックが逆に再生するので、論理のブロック(「もし」、「なら」、「でなければ」のそれぞれのブロック)を使う場合は注意! には気をつけてください。

3.2.13 音色と調の設定

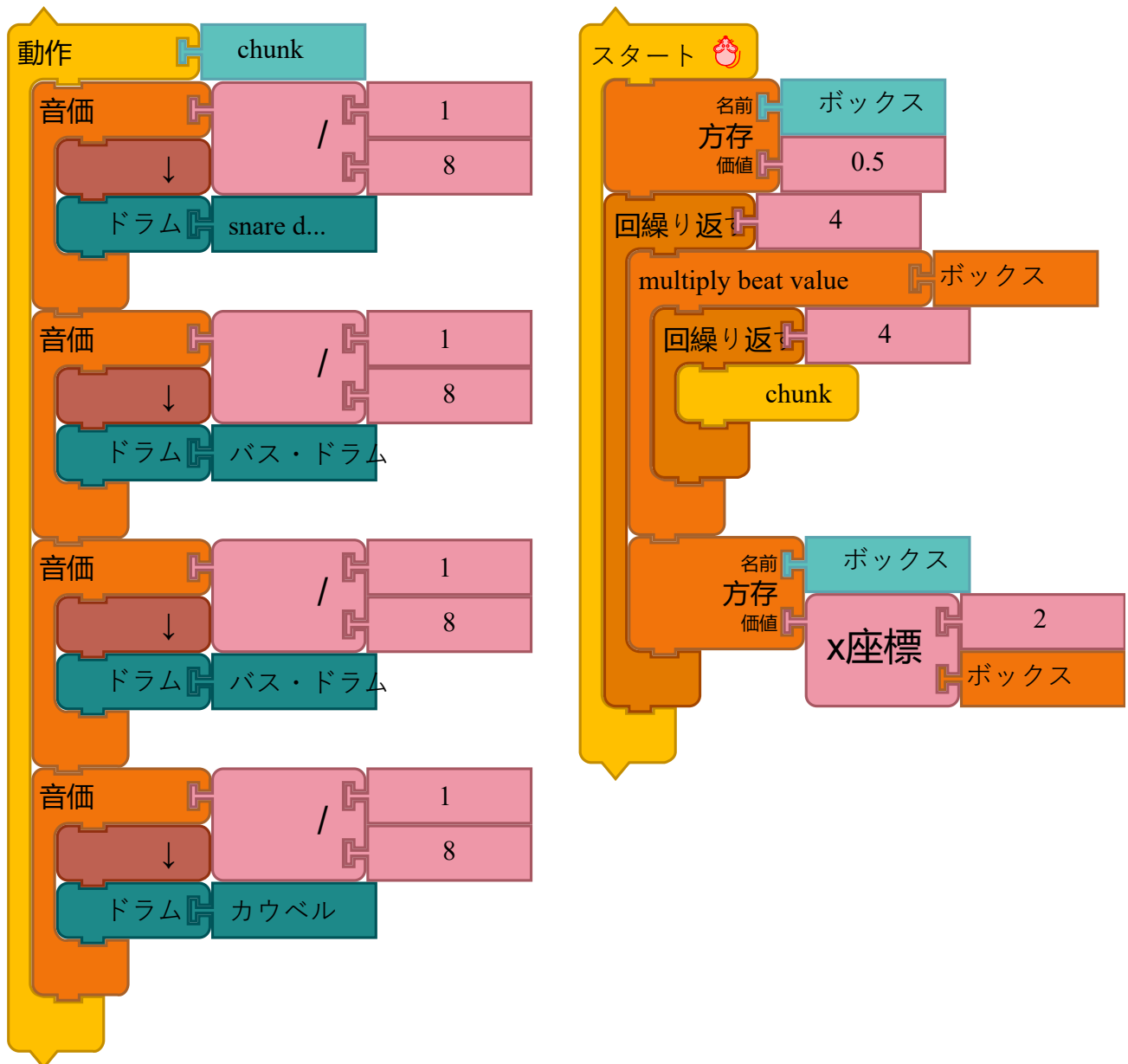


(needs to be changed in English version as well: Set synth and set keys has been moved, right???)

シンセを設定ブロックで入られているスタート(声部)のシンセを決めます。例えばバイオリンとチェロの音色のシンセを選ぶことができます。



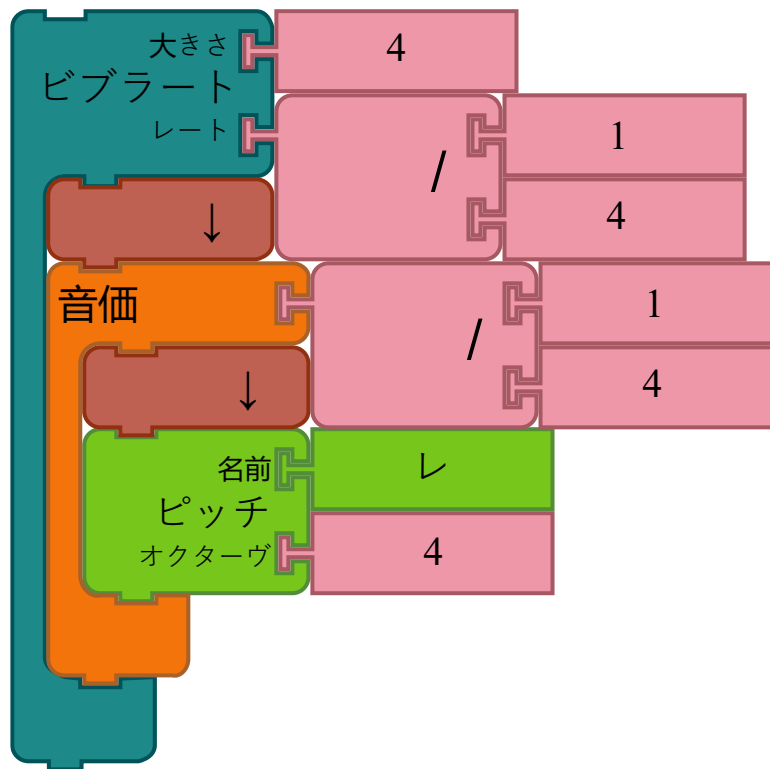
The **音階/モードを設定** block will change the key and mode of the mapping between solfege, e.g., ド, レ, ミ, to note names, e.g., C, D, E, when in C Major. 音階とモード include 長 and 短, Chromatic, and a number of more exotic 音階とモード, such as Bebop, Geez, Maqam, etc. This block allows users to access "movable ド" within Music Blocks, where the mapping of solfege to particular ピッチ changes depending on the user's specified tonality.



In the above example, the sequence of ドラム beats is increased over time.

ライブで再生 (http://sugarlabs.github.io/musicblocks/?file=MusicBlocks_drumexample.tb)

3.2.14 ビブラート



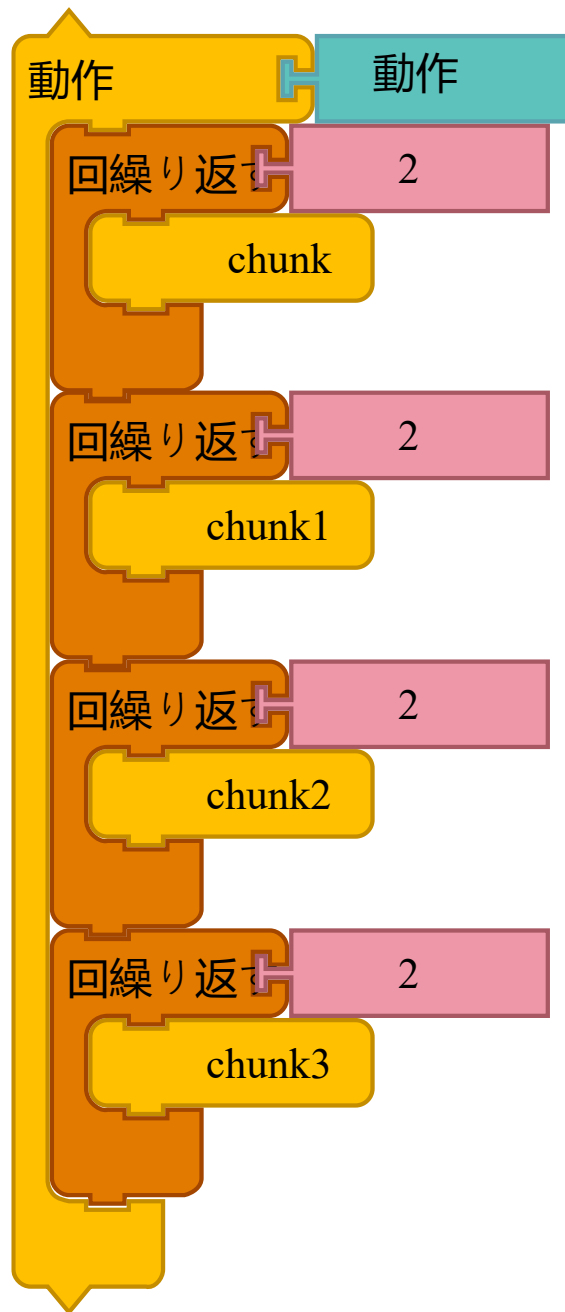
ビブラートブロックは入られているピッチの高音を早く小さく上げたり下げたりします。

「強度」のインプットの数値は1から100(セント)までです。100セントは半音と同じですよ。レートのインプットがそのビブラートされている音符の音価でどのレートで行われます。

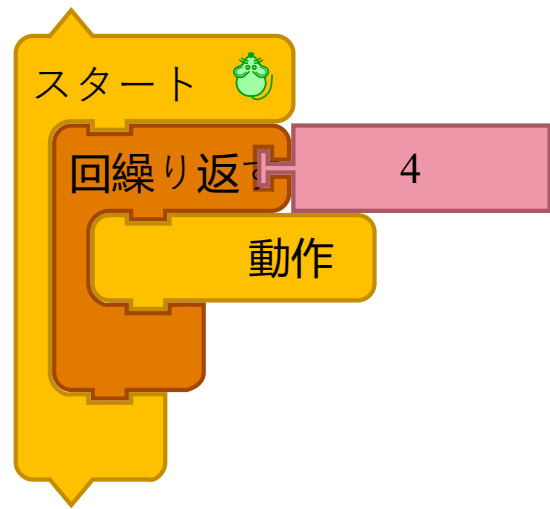
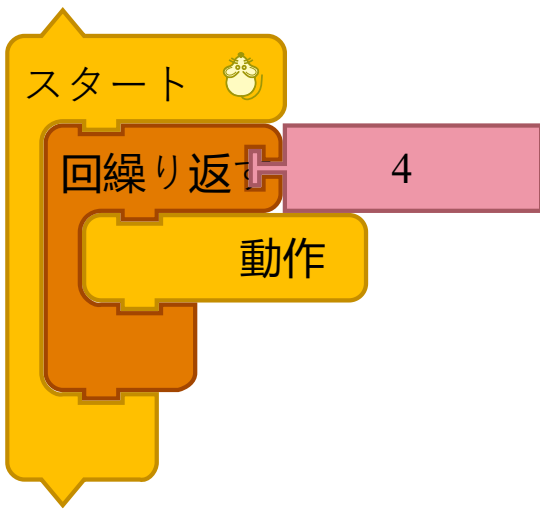
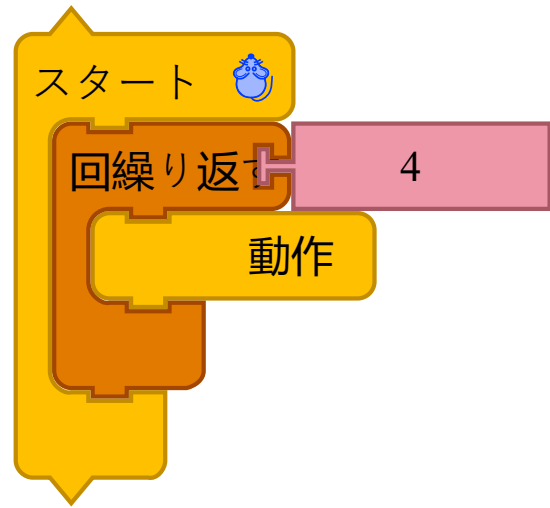
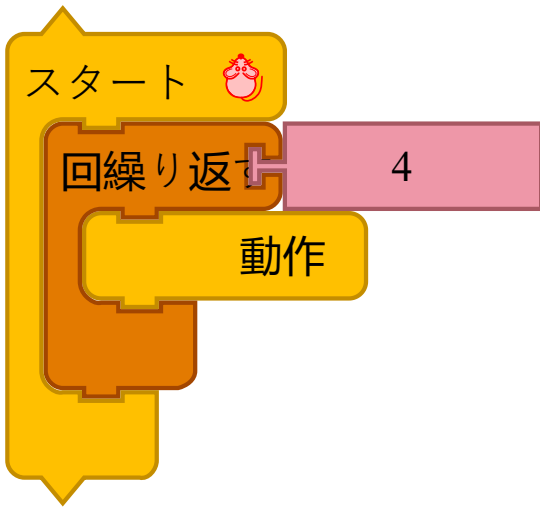
(I need to fix the Japanese, but we also need to fix the English as well)

3.3 声部

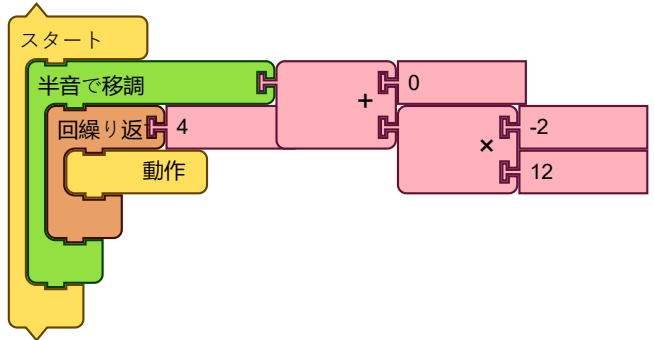
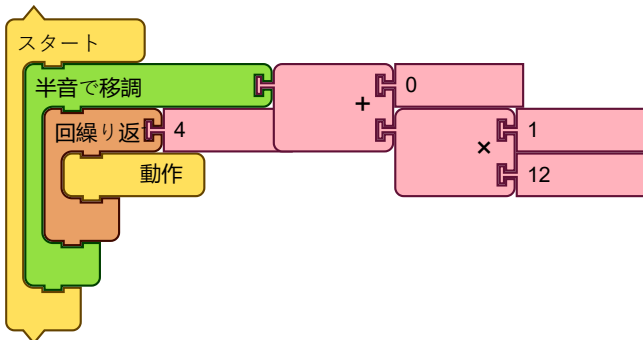
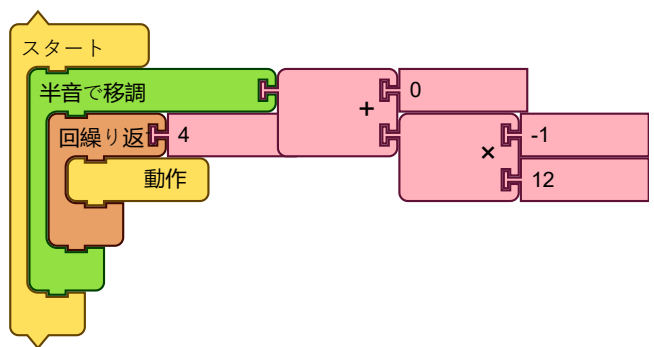
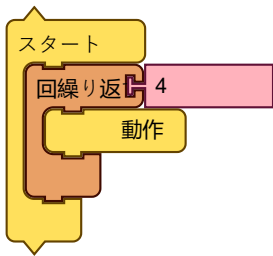
ミュージック・ブロックスで「再生」のボタンを押すとそれぞれのスタートブロックは格声部のように再生します。(「再生」のボタンを押す時、すべてのスタートブロックのコードが同時に再生します。)



この前の例のmelodyをもし、新しい声部とするため別のスタートブロックに入れて...

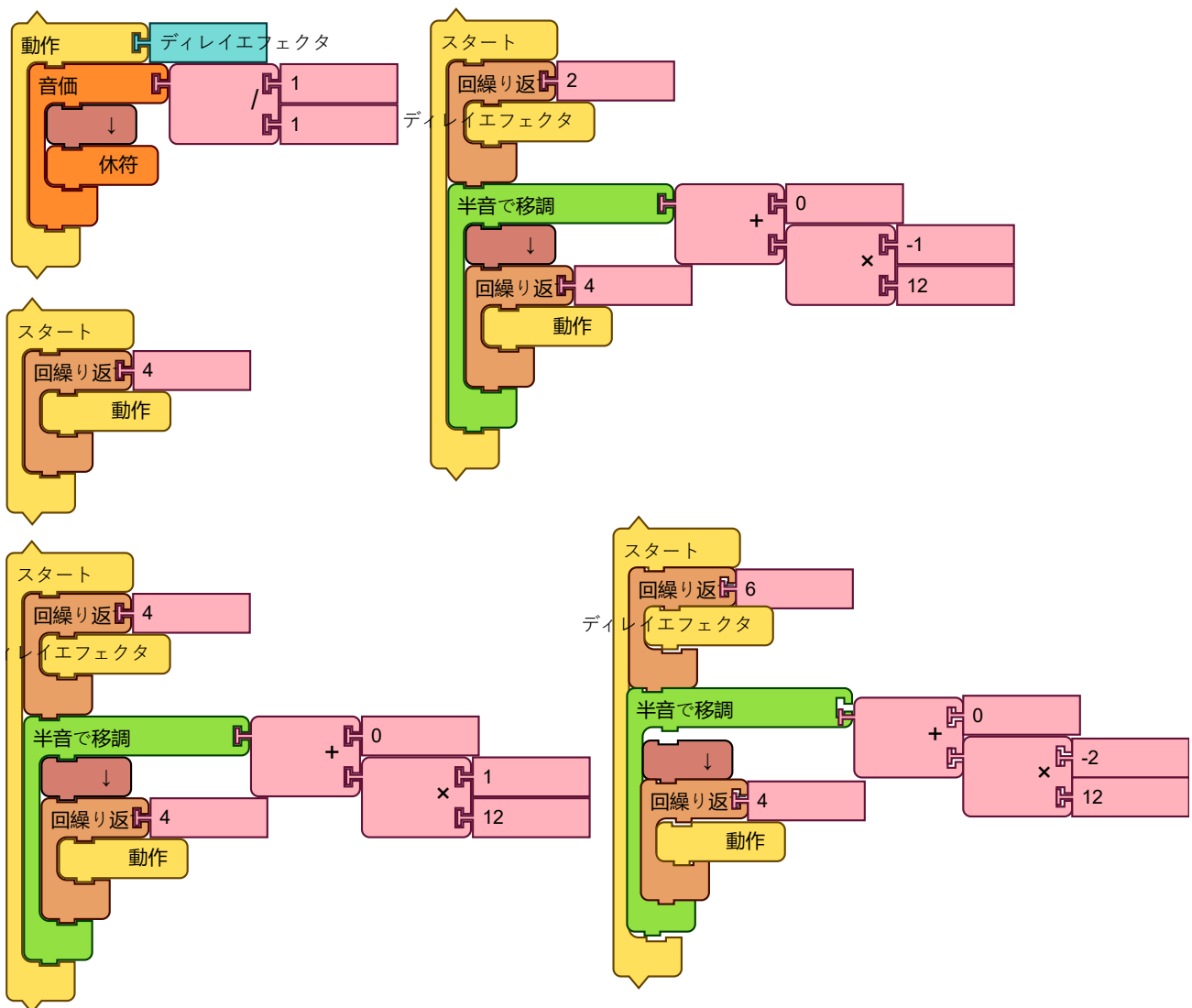


...複合的なスタートブロックで再生ができますよ！



もしオクターヴを上にか、下にか**転化**すればもともとおもしろくなりますよ！

(Add to English and Japanese -- It is interesting b/c of the differentiation in sound)

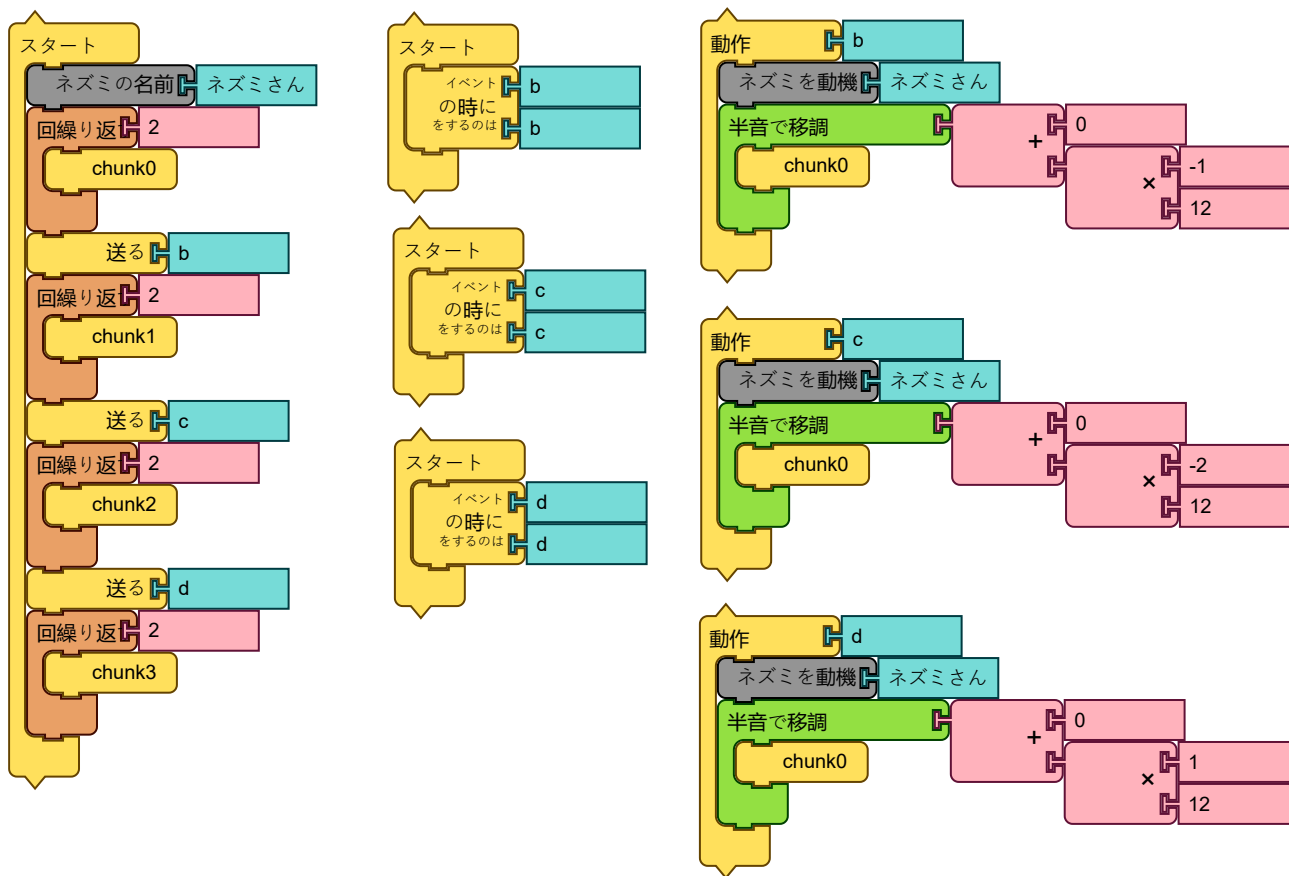


時刻的にそのそれぞれの声部をずれば最高におもしろいではありませんか？

(mention "round", which I guess in Japanese is "canon")

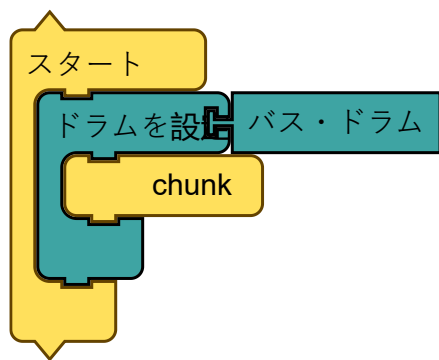
格声部を同じmelodyをこんな風に時刻的にずれるのは「ラウンド」、また「カノン」と言います。

ライブで再生 (<https://musicblocks.sugarlabs.org/index.html?id=1523026536194324&run=True>)



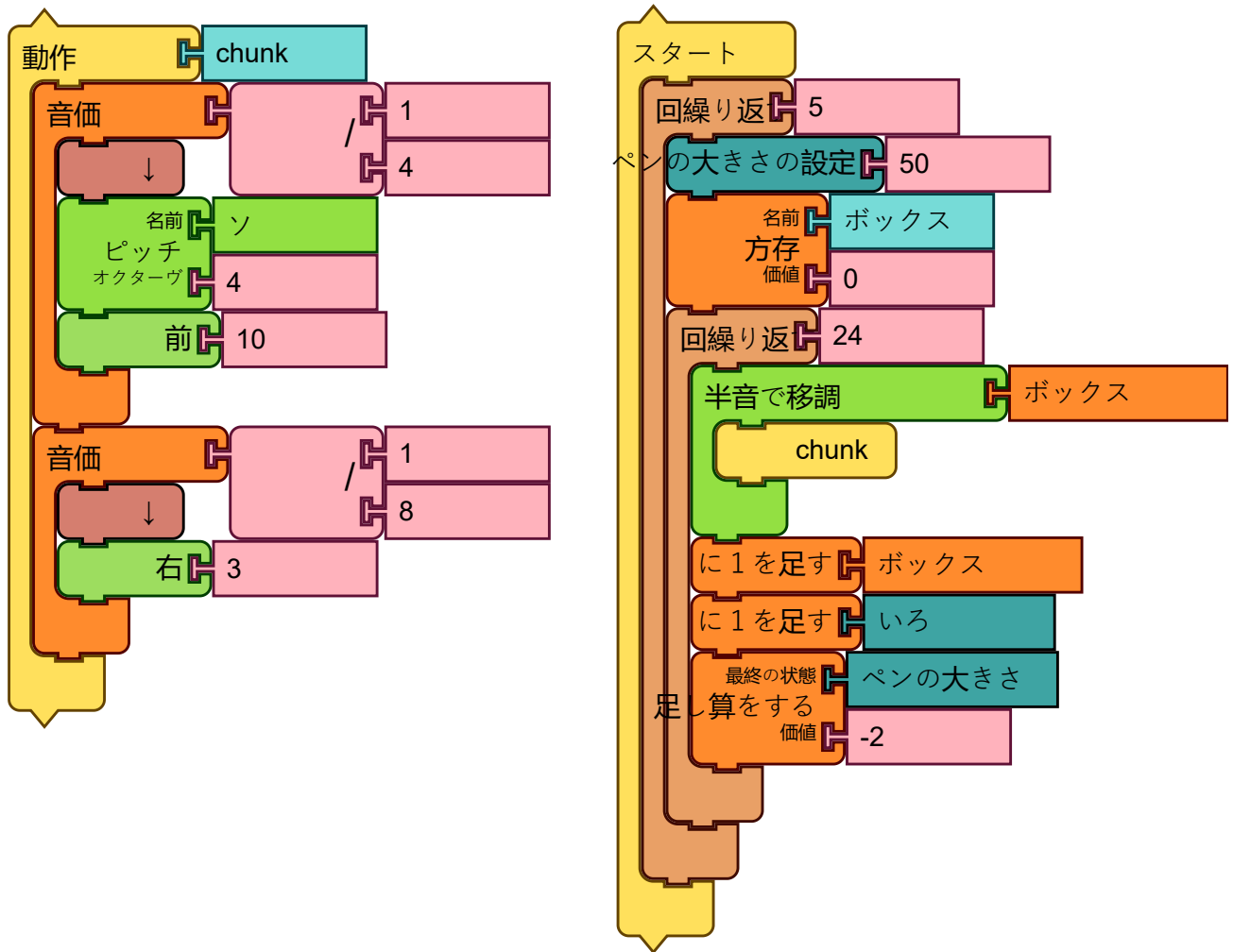
(what is meant by "preprogrammed delay"? ...oh, I see. Grammar needs improvement.)

声部の時刻をずれるためもう一つのやりかたが、送るブロックを使って声部の「最小時刻」をそのブロックが送ります。上の例には、melodyの決められた部分が再生されたら、イベントがそのブロックから送られて格声部ができます。ネズミを動機ブロックを使ってすべての声部が同じクロックに合わせて再生します。



ドラムのトラックを作るため、"ドラム"の特別なスタートブロックがあります。ドラム・スタートに入っている、すべてのピッチブロックが、ドラムに変えて打ち鳴らします・c2がドラムのデフォルトです。(=< I do not understand what is meant by default sample.)上の例にチャンクの中に入ってるピッチがバス・ドラムの音として打ち鳴らします。

3.4 グラフィックスと一緒に



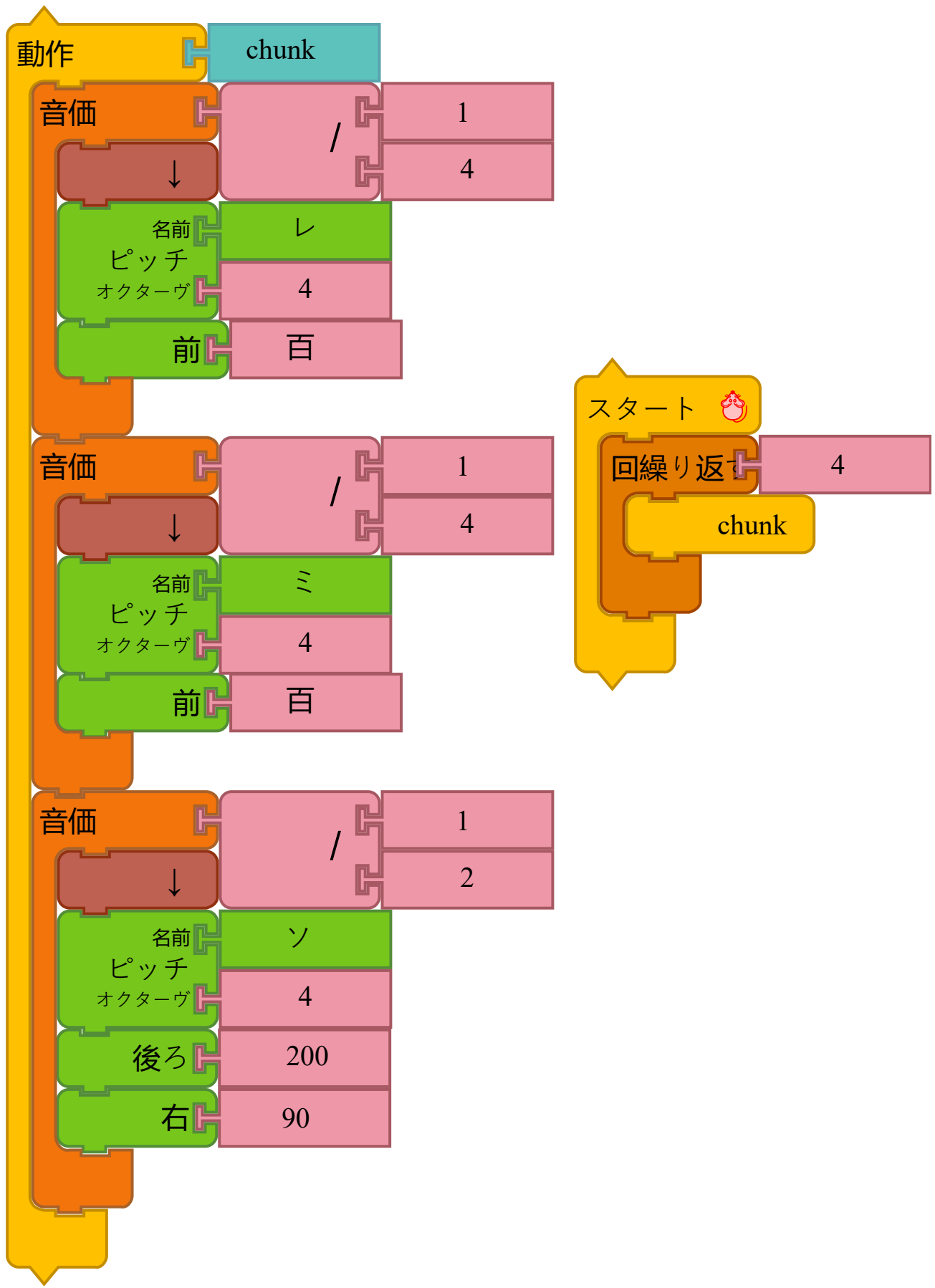


ネズミ・グラフィックス(もともとタートル・グラフィックス)は音楽と一緒に組み合わせることができます。ネズミのパレットからのグラフィックスブロック(前と後ろのブロック)を音価ブロックの間に入れるとグラフィックスの動きが音楽と同時にします。

この例でネズミのSpriteが4分音価ずつ、音符が鳴らすと同時に前に動きます。それから8分音価の音符と一緒に、右に向かって回ります。

回繰り返すの回数で、ピッチが半音で高く転化して、ペンの大きさが少し大きくなります。

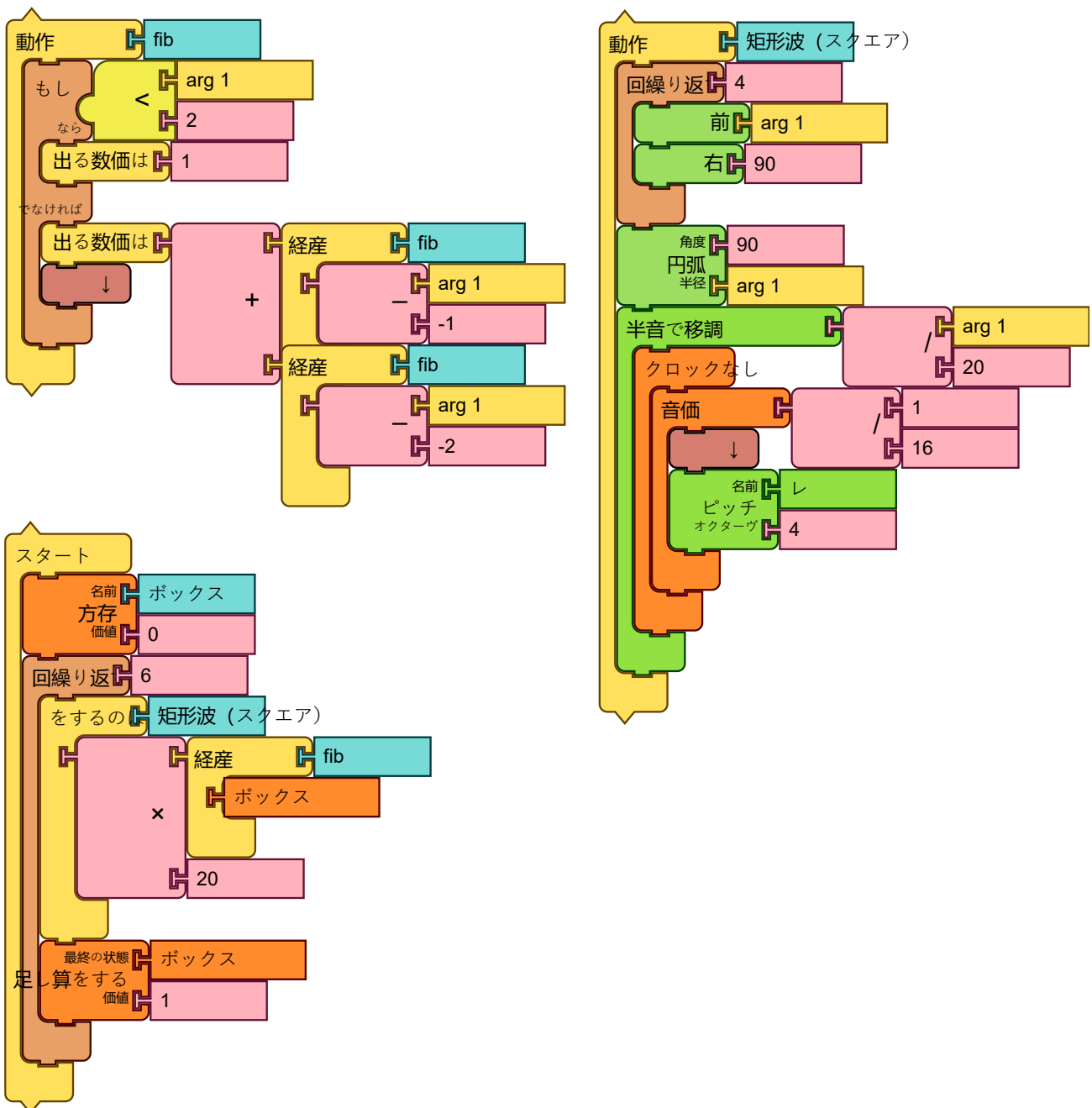
ライブで再生 (<https://musicblocks.sugarlabs.org/index.html?id=1518563680307291&run=True>)



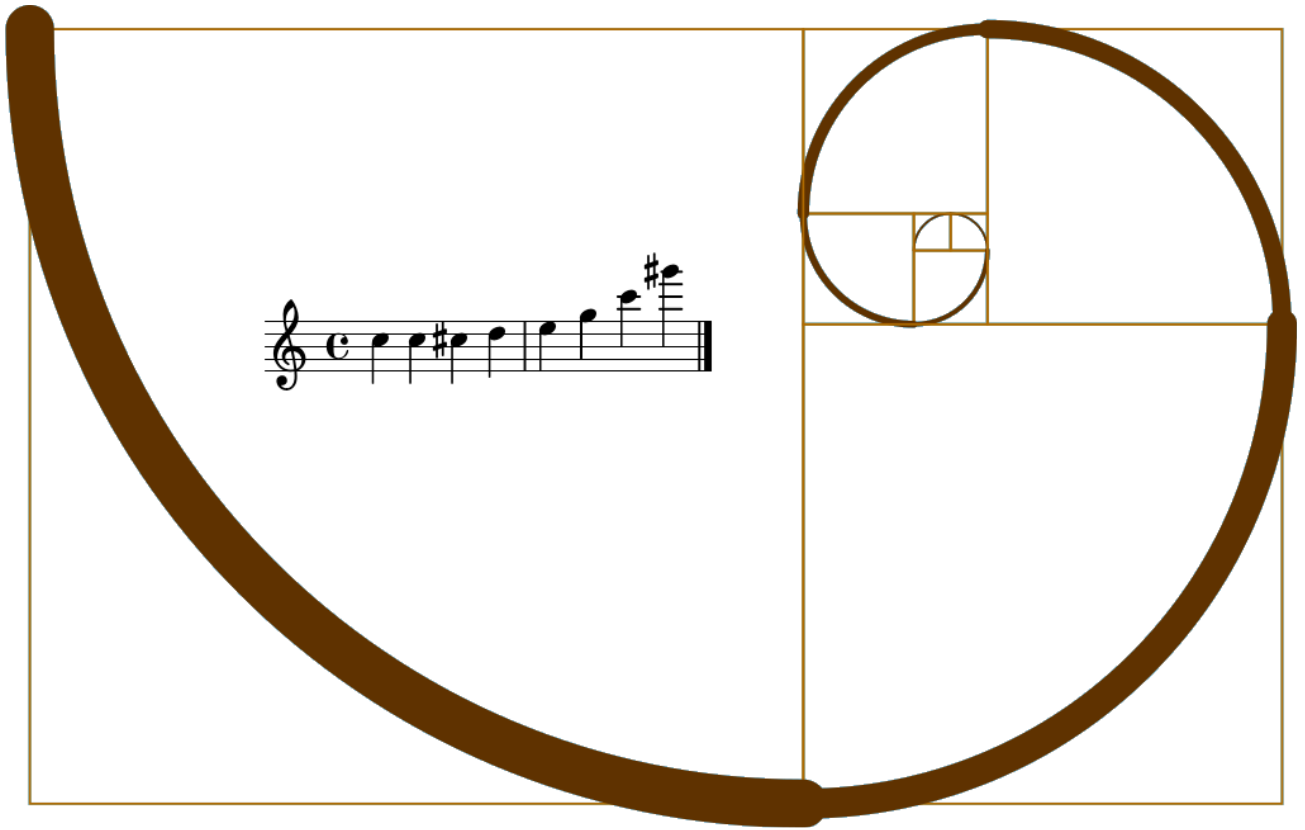
グラフィックスと音楽と一緒に合わせるため、もう一つのやり方がグラフィックスのコードを音価ブロックに入れるとその音価の期間で動作が行います。

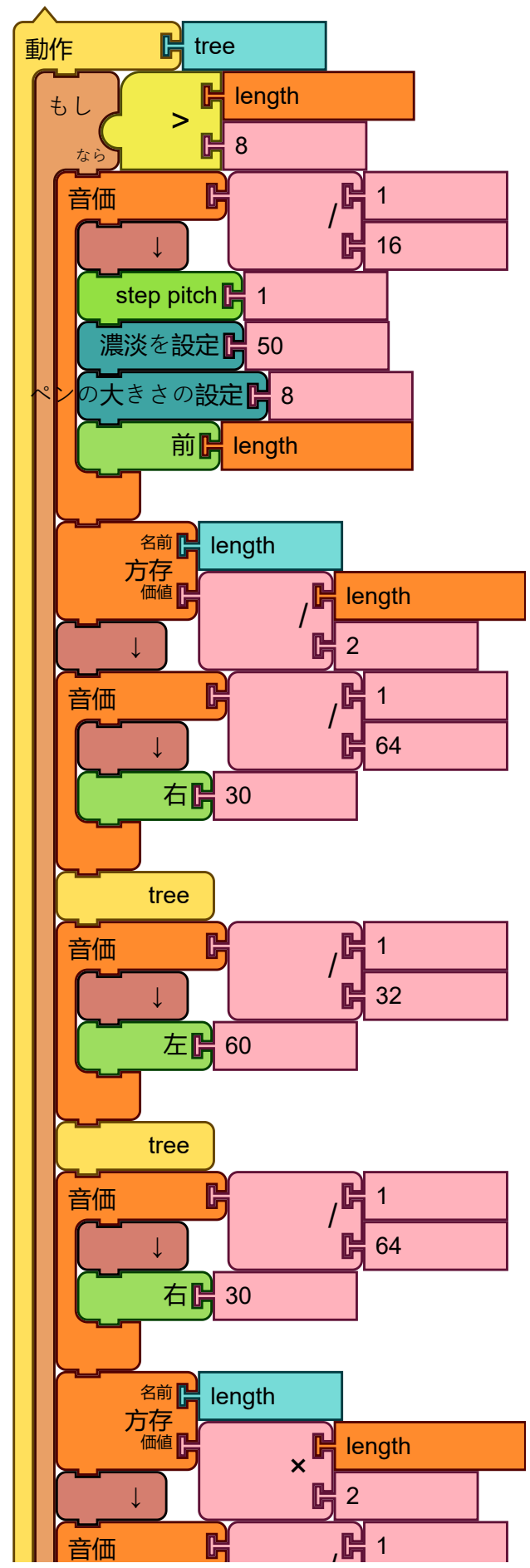
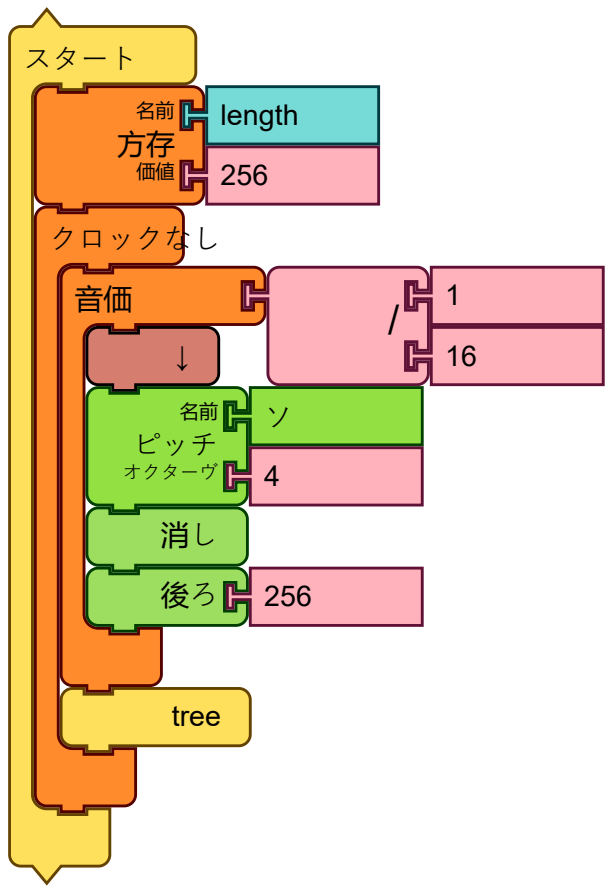
ネズミのSpriteがダンスのように動くでしょう。(added a comment that it is "dance-like"; need to change instances of turtle to mouse in English version)

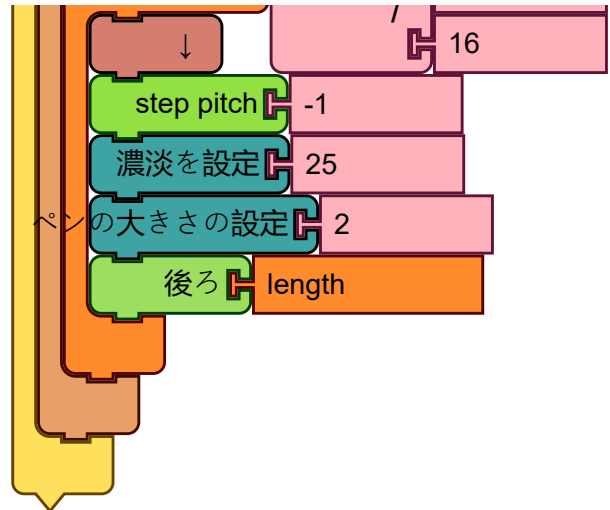
ライブで再生 (<https://musicblocks.sugarlabs.org/index.html?id=1523106271018484&run=True>)



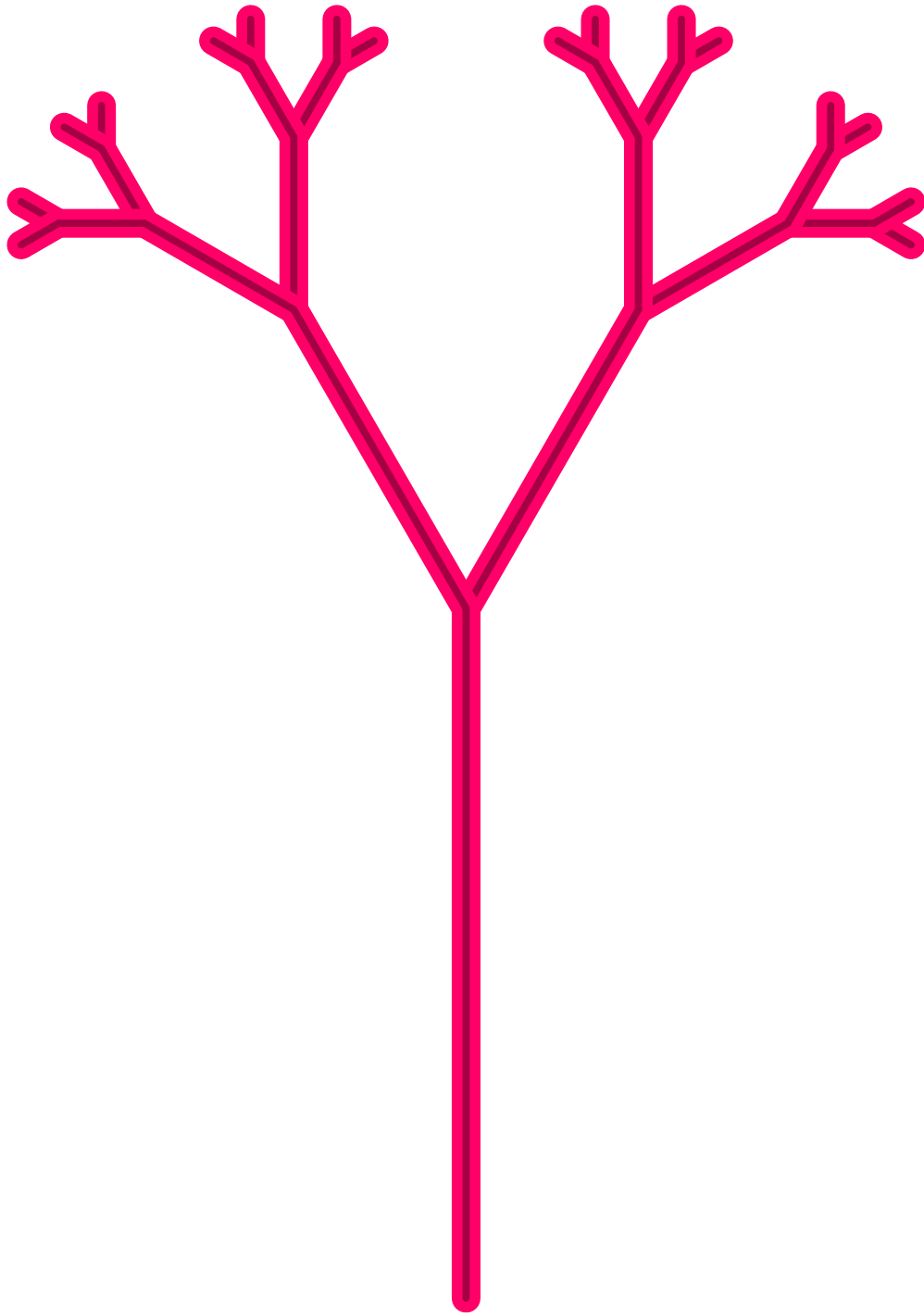
上の例に、グラフィックスと音楽のコンピュータの操作がもと複雑の訳で音楽のタイミングをずらさないため、クロックなしブロックがグラフィックスと音楽の動きを区別します。"クロックなし"ブロックはそれぞれの動作の順番をリズムより優先します。







グラフィックスを音符に入れるのもう一つの例です。木の形をリズムに合わせてグラフィックスをネズミさんが書きます。木の枝、上ほど、ネズミが動くとピッチの高音も高くなります。



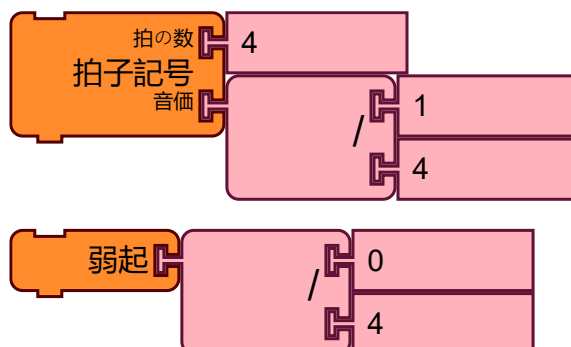
[ライブで再生 \(https://musicblocks.sugarlabs.org/index.html?id=1523029986215035&run=True\)](https://musicblocks.sugarlabs.org/index.html?id=1523029986215035&run=True)

3.5 拍子

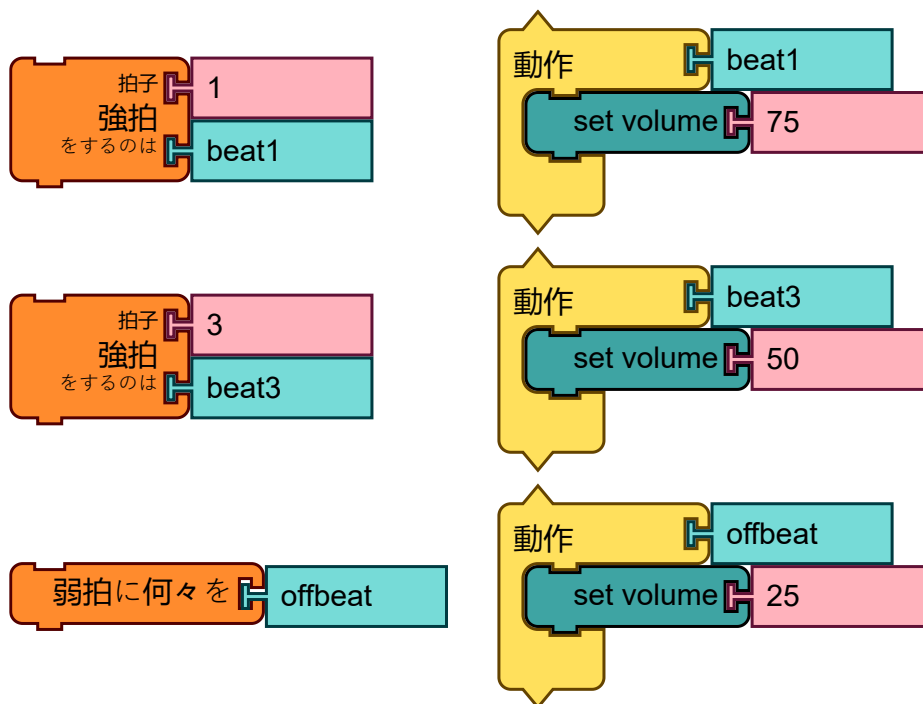
拍子ブロックで音楽のビートを決めます。拍子ブロックは拍子記号ブロックと一緒に使ってデフォルトが4/4です。

弱起ブロックが初めの全拍子記号の前の拍子音価を決めるためです。弱起と言うのは「ピッカップ」とも読んでいます。

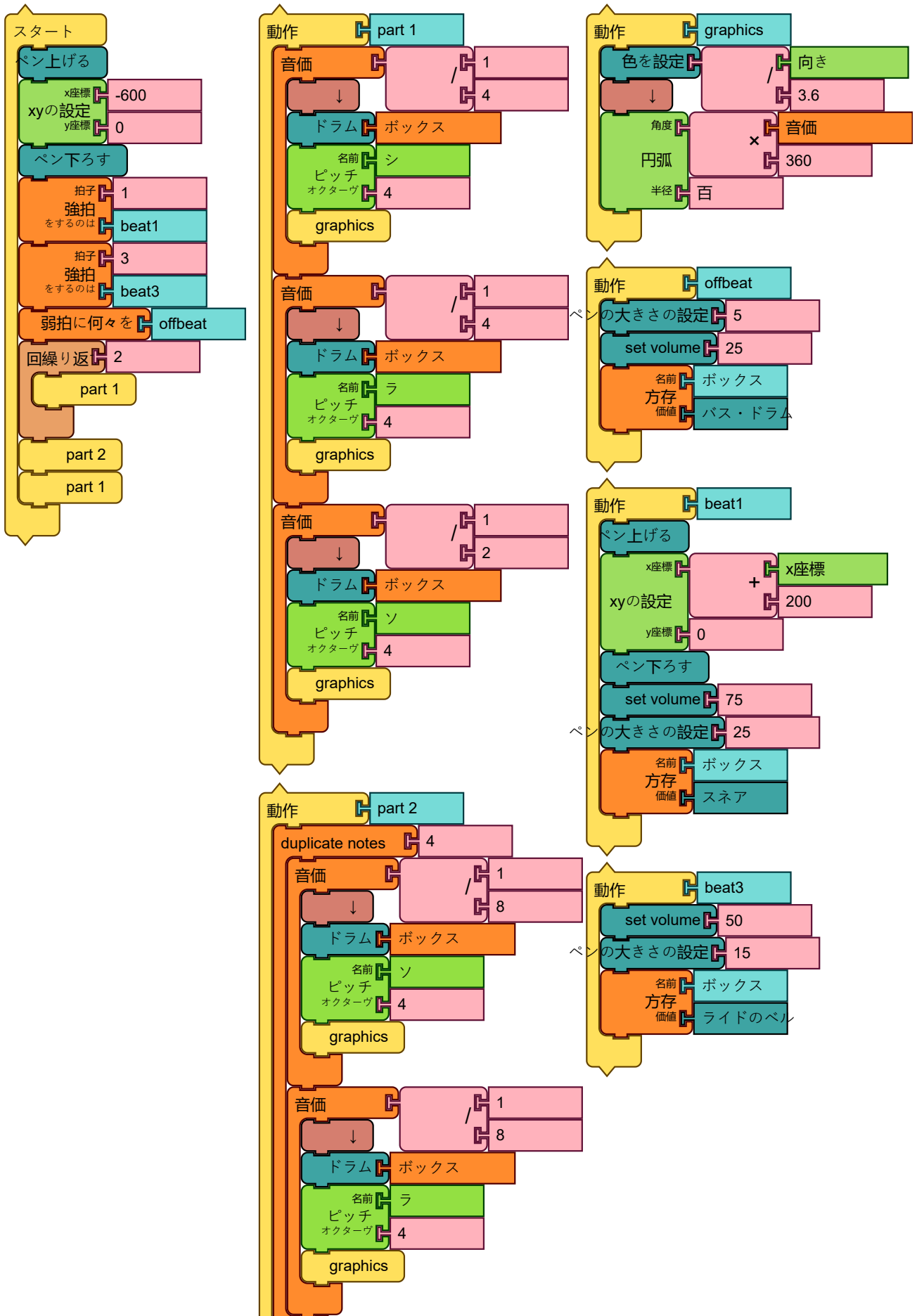
(Need better definition for pickup in English version)



拍子を指定するの役に立つ時があります。例えば、下の例で音符の音量が1と3の拍子に増して、それぞれ残りの(弱)拍子の音量が小さくします。

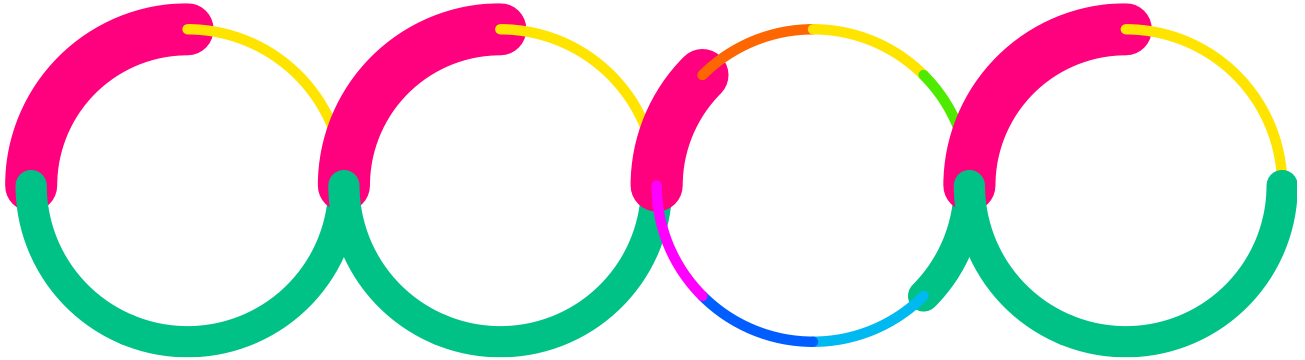


拍子に何々をする動作ブロックと弱拍に何々をする動作ブロックでそれぞれの拍子に行う動作を指定することができます。(Note that the action is run before any blocks inside the note block associated with the 拍子 are run.)





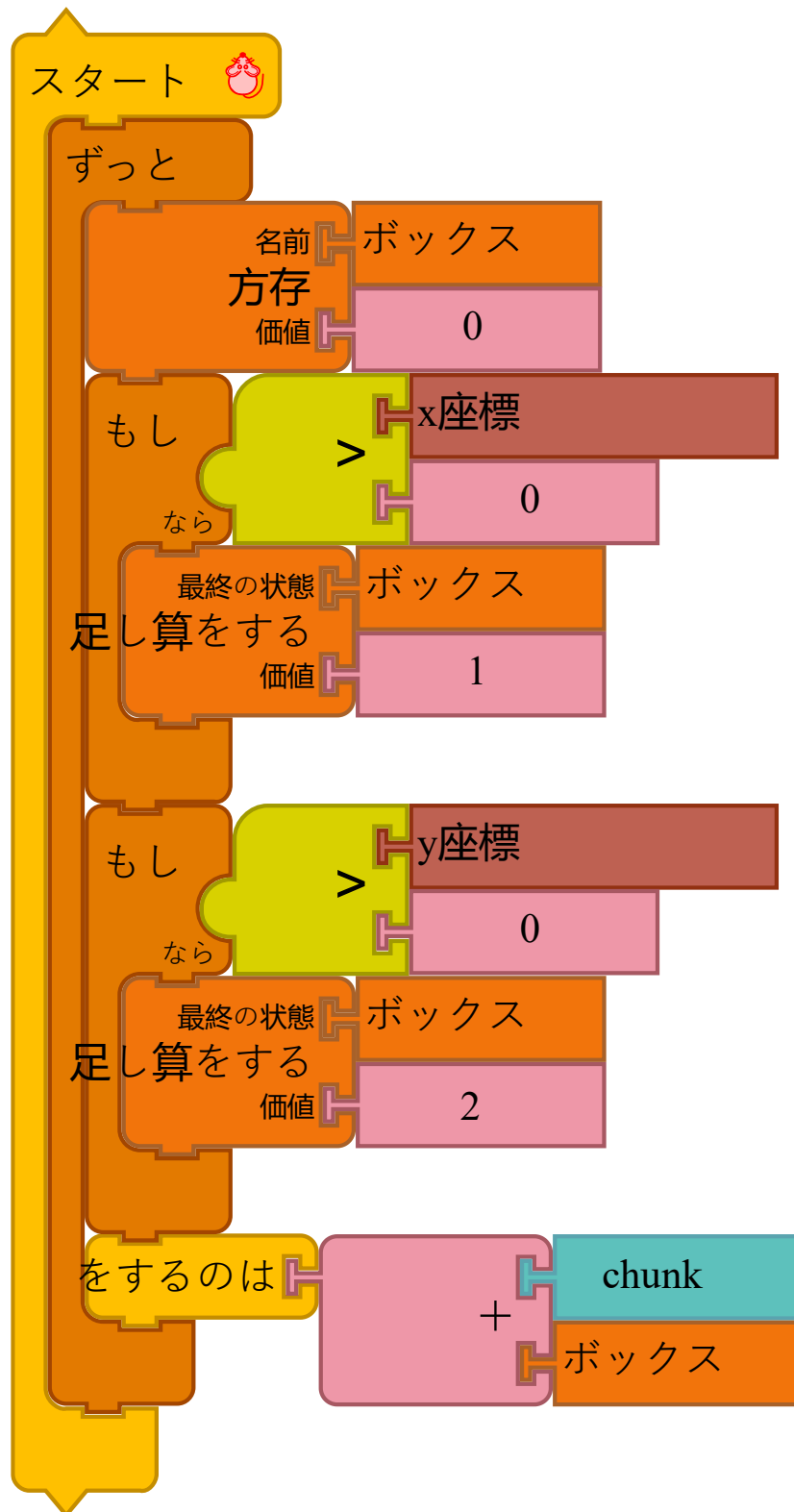
Another approach to グラフィックスとのもう一つの使い方が拍子でそのグラフィックスを 転化(Modulate)することです。 . In the example above, we call the same グラフィックス action for each note, but the parameters associated with the action, such as pen width, are dependent upon which 拍子 we are on. On 拍 1, the pen size is set to 50 and the 音量 to 75. On 拍 3, the pen size is set to 25 and the 音量 to 50. On off 拍子s, the pen size is set to 5 and the volumne to 5. The resultant graphic is shown below.



3.6 相互作用

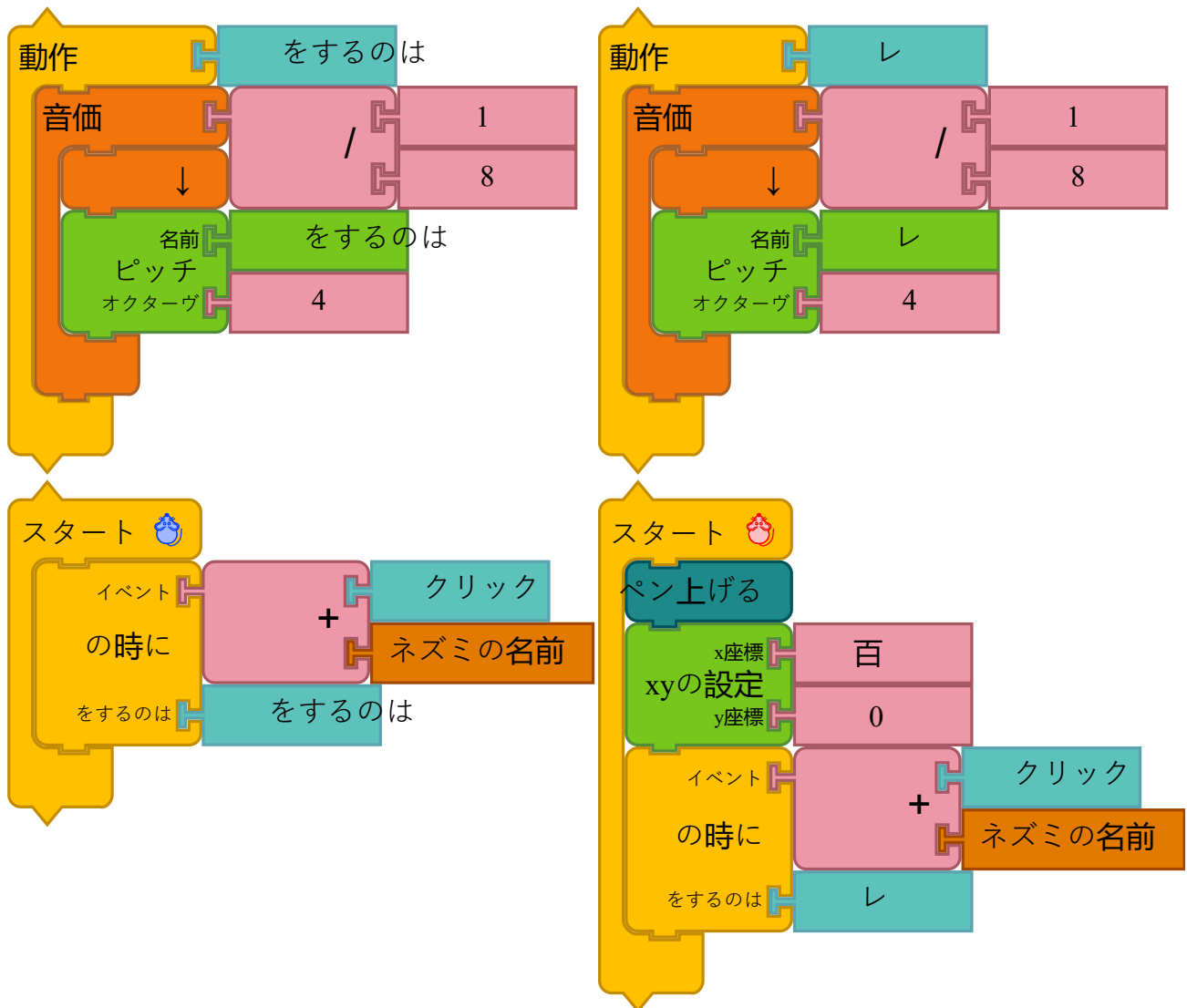
ミュージック・ブロックスでいろんな相互のコードの仕方があります。例えば他のネズミの動作を聞いて反応することもできます。

(grammar error in English)



上の例の用にチャンクに入っているmelodyが相互的に再生できます。マウスのカーソルが左下の四分円に入るとチャンクが再生し、右下の四分円に入るとチャンク1が再生し、左上に入るとチャンク2が再生し、右上の四分円に入るとチャンク3が再生します。

ライブで再生 (<https://musicblocks.sugarlabs.org/index.html?id=1523028011868930&run=True>)

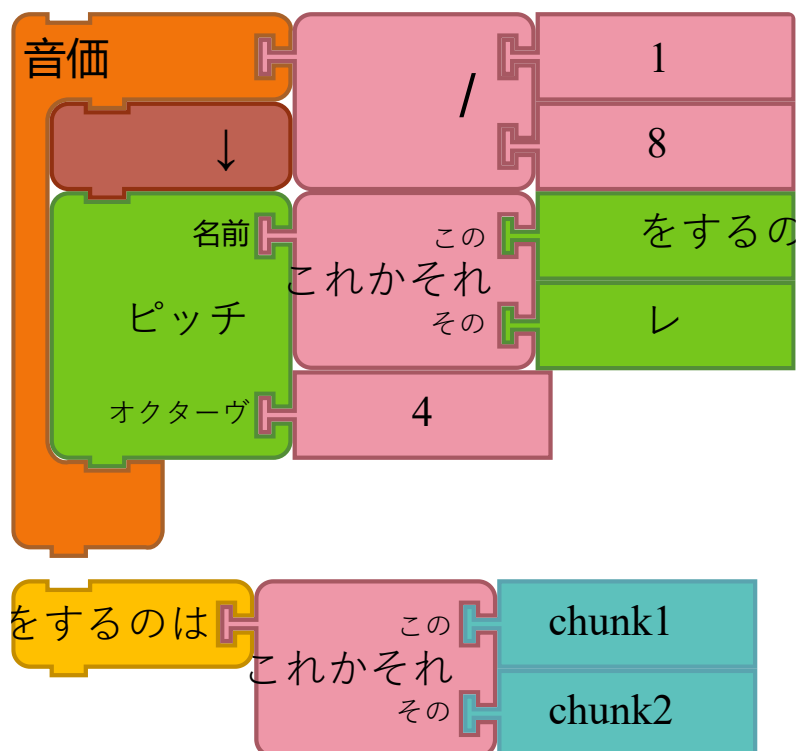


上の例で二つのキーのあるピアノが創作されています。それぞれの音符がクリックイベントを二つのネズミスプライトに分けます。

この例をよく勉強して8キーのピアノがミュージック・ブロックスで作れるんでしょうか？

("Turtles" again)

ライブで再生 (<https://musicblocks.sugarlabs.org/index.html?id=1518563680450252&run=True>).



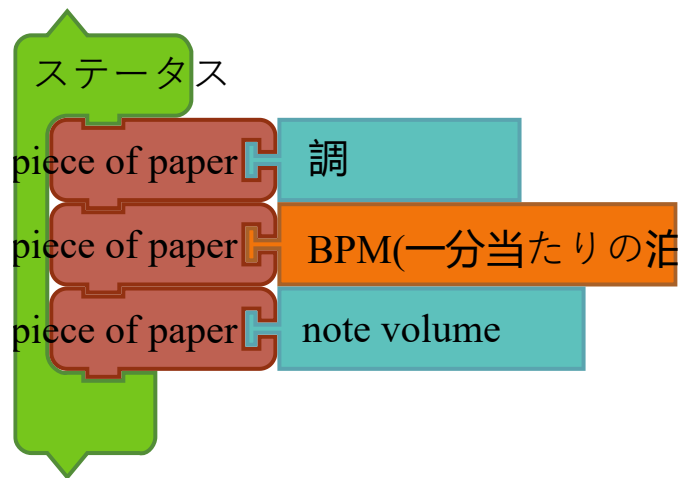
乱数的なものが音楽と一緒に入れることができます。上の例にこれかそれブロックは乱数的にド、それともレを、音価ブロックが毎回鳴る時その音符のピッチにします。その下の例について、これかそれのブロックが乱数的に チャンク1 を、それとも チャンク2 を選べます。

ウィジェット

[前のトピック \(3. 音楽でプログラミング\)](#) | [目次に戻す](#) | [次のトピック \(5. ミュージック・ブロックスから以遠\)](#)

案内のこれからがミュージック・ブロックスの色々なウィジェットを紹介して使い方を案内します。ミュージック・ブロックスにあるウィジェットがコードと音楽をもともと分かりやすくするためです。

4.1 ステータス

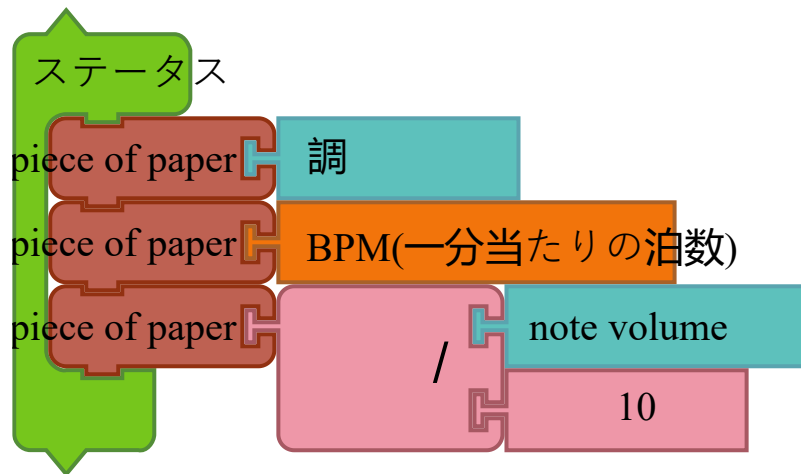


	調	bpm	音量	音符
	C Major	90	50	C4 8
	C Major	90	50	A5 4

(Needs to see these pictures... later)

ステータス・ウィジェットはミュージック・ブロックの音符と計算との色々の再生されているものをどうなっているかのステータスの印刷ができるツールです。ステータスのデフォルトで出るのが音階(またモード)、BPM、また音量です。そのうえに鳴らされている音符がみんな鳴るうちに放送します。一つの一行に一つの音声の鳴らされている音符のピッチ、また音価の数値がステータスで出ます。

デフォルトの印刷だけでなく、そのより自分の見たいブロックもステータス ウィジェットに入れることができます。例えば、音楽に対する音量、転化、音符の省略、スタッカート、スラー、など入れることができ、グラフィックスに対するx、y、向き、色、暗がり、灰色、ペンの大きさ、などを入れてそのそれぞれのブロックに関係ある放送の見るることができます。



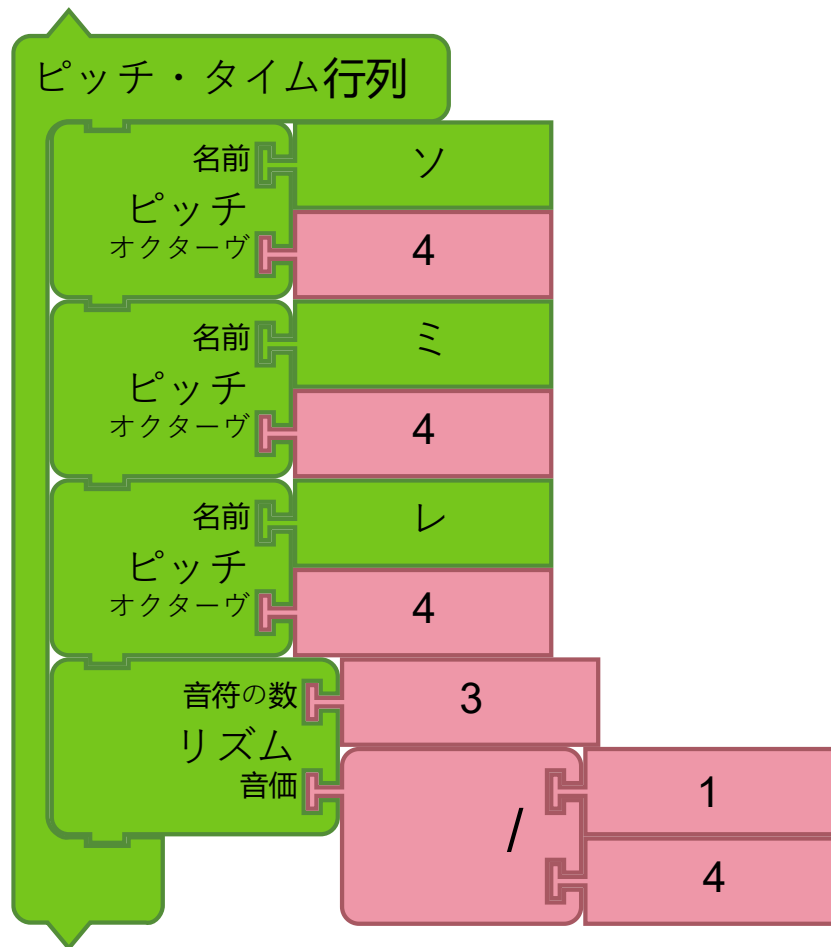
ステータス・ブロックでその他にプログラムできます。上の例でステータス・ブロックに入っているのは今の鳴らされている音量が放送する前、10割の計算がされています。

4.2 音符のチャンクの作り方

Using the ピッチ・タイム行列を使って、音符のチャンク(メロディ)を簡単に作ることができます。

(Do we really want to say, "at a much faster speed"? Maybe "in a more convenient and intuitive manner"? or something else)

4.2.1 The ピッチ・タイム行列



ミュージック・ブロックの一つのウィジェットがピッチ・タイム行列で、ピッチと時期が行列となっているから音符の流れ方がもと簡単に想像できます。

(It is easier because you can see pitch over time; you can see the movement of the notes. Please take another look at English as well)

ピッチ・タイム行列を使いたい際はパレットで「ウィジェット」をクリックして「ピッチ・タイム行列」をそのパレットから引きます。そのブロックをクリックすると行列が現れるべきです。ピッチが横でリズム(時期)が縦です。

(This in the English version is dated as we do not have the pitch-time matrix at the start)

Solfa							
sol							
mi							
re							
rhythmic note values	1/4 ♪	1/4 ♪	1/4 ♪				

The matrix in the figure above has three ピッチ blocks and one リズム block, which is used to create a 3 x 3 grid of ピッチ and time.

Note that the default matrix has five ピッチ blocks, hence, you will see five rows, one for each ピッチ. (A sixth row at the bottom is used for specifying the リズム・ブロック associated with each note.) Also by default, there are two リズム blocks, which specifies six quarter notes followed by one half note. Since the リズム blocks are inside of a 回繰り返す block, there are fourteen (2 x 7) columns for selecting notes.

Solfa							
sol							
mi							
re							
rhythmic note values	1/4 ♪	1/4 ♪	1/4 ♪				

By クリック on individual cells in the grid, you should hear individual notes (or 和音s if you クリック on more than one cell in a column). In the figure, three quarter notes are selected (black cells). First レ 4 , followed by ミ 4 , followed by ソ 4 .



If you クリック on the *Play* button (found in the top row of the grid), you will hear a sequence of notes played (from left to right): レ 4 , ミ 4 , ソ 4 .



Once you have a group of notes (a "チャンク") that you like, クリック on the *Save* button (just to the right of the *Play* button). This will create a stack of blocks that can be used to play these same notes programmatically. (More on that below.)

You can rearrange the selected notes in the grid and save other チャンク as well.



The *Sort* button will reorder the ピッチ in the matrix from highest to lowest and eliminate any 重に ピッチ blocks.



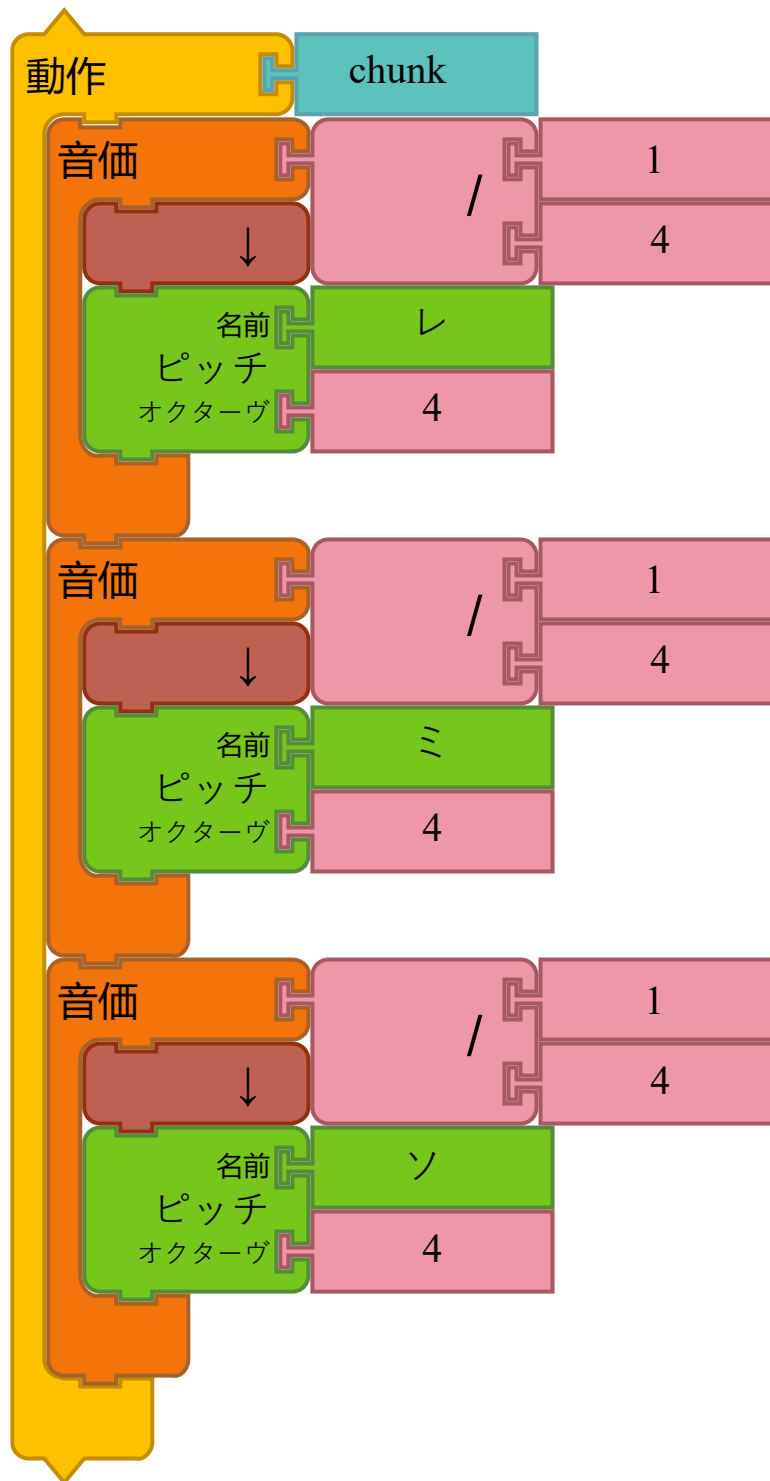
You can hide the matrix by クリック on the *Close* button (the right-most button in the top row of the grid.)



There is also an Erase button that will clear the grid.

Don't worry. You can reopen the matrix at anytime (it will remember its previous state) and since you can define as many チャンク as you want, feel free to experiment.

Tip: You can put a チャンク inside a ピッチ・タイム行列 block to generate the matrix that corresponds to that チャンク.

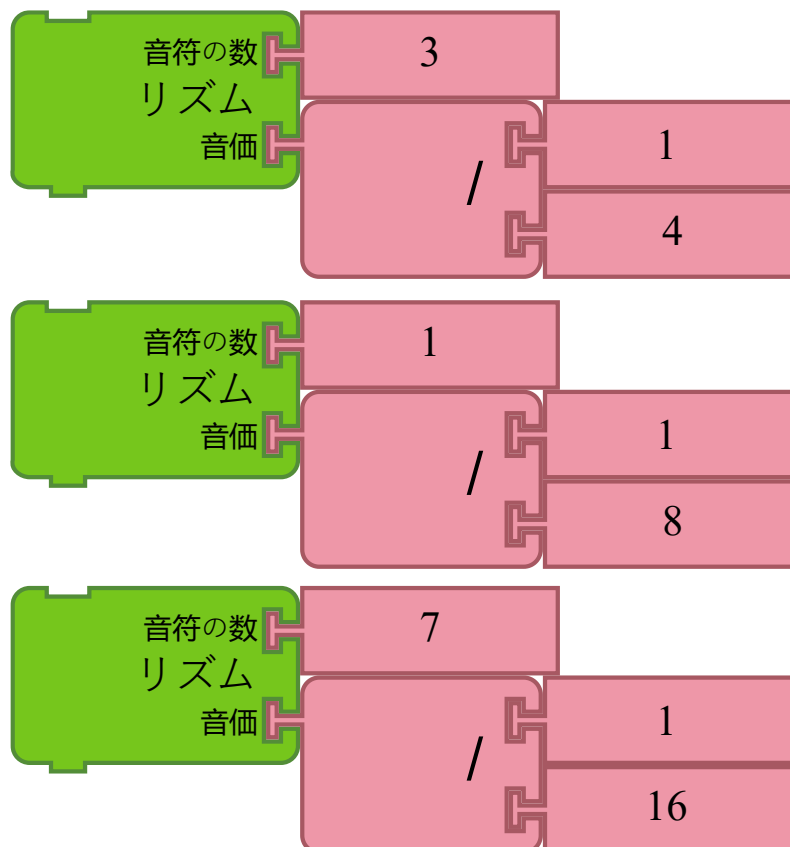


The チャンク created when you クリック on the matrix is a stack of blocks. The blocks are nested: an *動作* block contains three *Note value* blocks, each of which contains a *ピッチ* block. The *動作* block has a name automatically generated by the matrix, in this case, チャンク. (You can rename the action by クリック ing on the name.). Each note has a duration (in this case 4, which represents a quarter note). Try putting different numbers in and see (hear) what happens. Each note block also has a *ピッチ* block (if it were a 和音, there

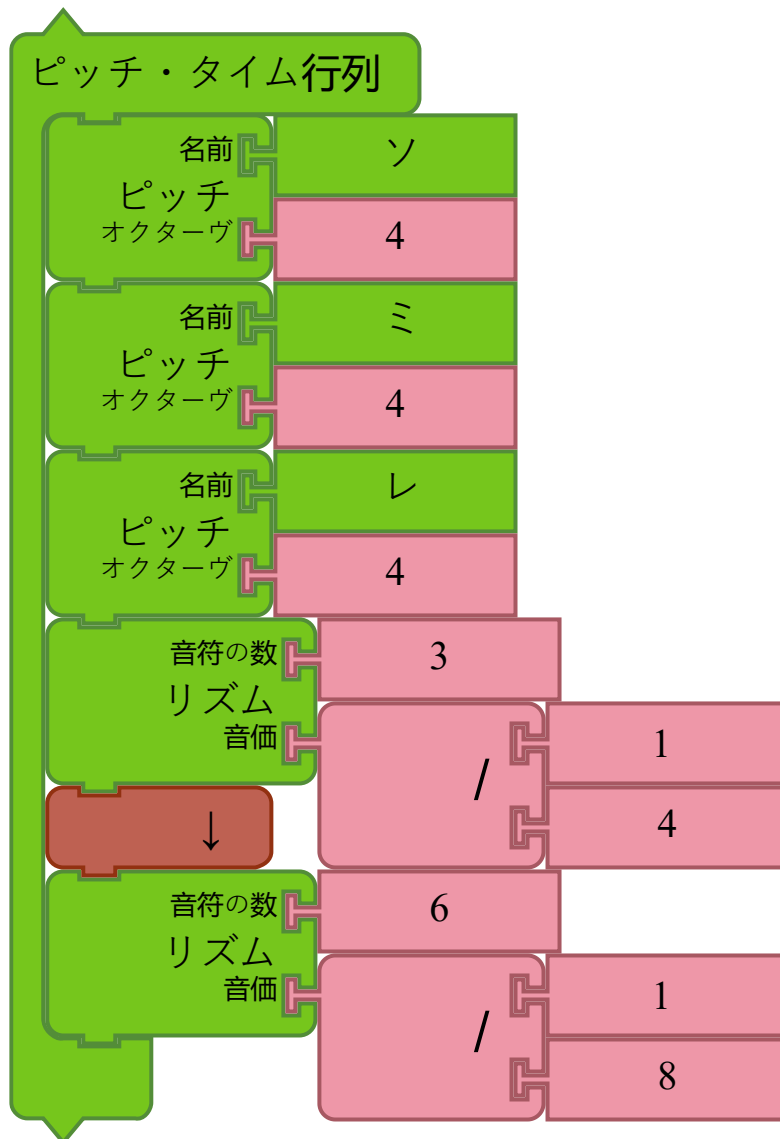
would be multiple ピッチ blocks nested inside the Note block's clamp). Each ピッチ block has a ピッチ name (レ, ミ, and ソ), and a ピッチ octave; in this example, the octave is 4 for each ピッチ. (Try changing the ピッチ names and the ピッチ octaves.)

To play the chuck, simply クリック on the action block (on the word action). You should hear the notes play, ordered from top to bottom.

4.2.2 リズムのブロック



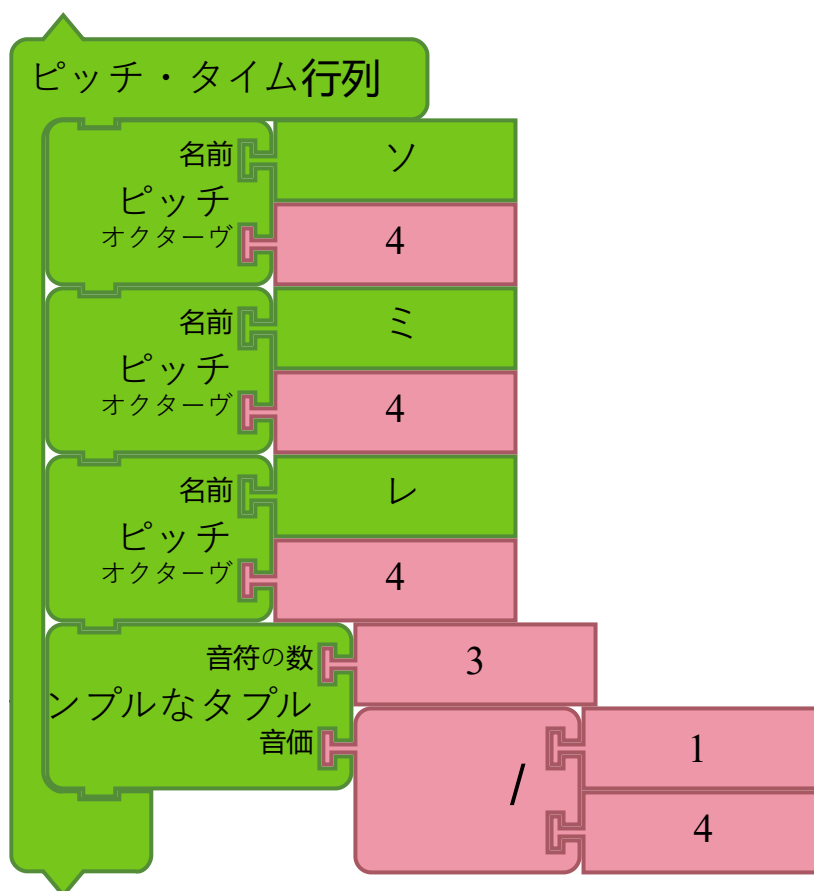
リズム blocks are used to generate リズム patterns in the ピッチ・タイム行列 block. The top argument to the リズム block is the number of notes. The bottom argument is the duration of the note. In the top example above, three columns for quarter notes would be generated in the matrix. In the middle example, one column for an eighth note would be generated. In the bottom example, seven columns for 16th notes would be generated.



Solfa							
sol							
mi							
re							
rhythmic note values	1/4	1/4	1/4	1/8	1/8	1/8	1/8

You can use as many *リズム* blocks as you'd like inside the *ピッチ・タイム行列* block. In the above example, two *リズム* blocks are used, resulting in three quarter notes and six eighth notes.

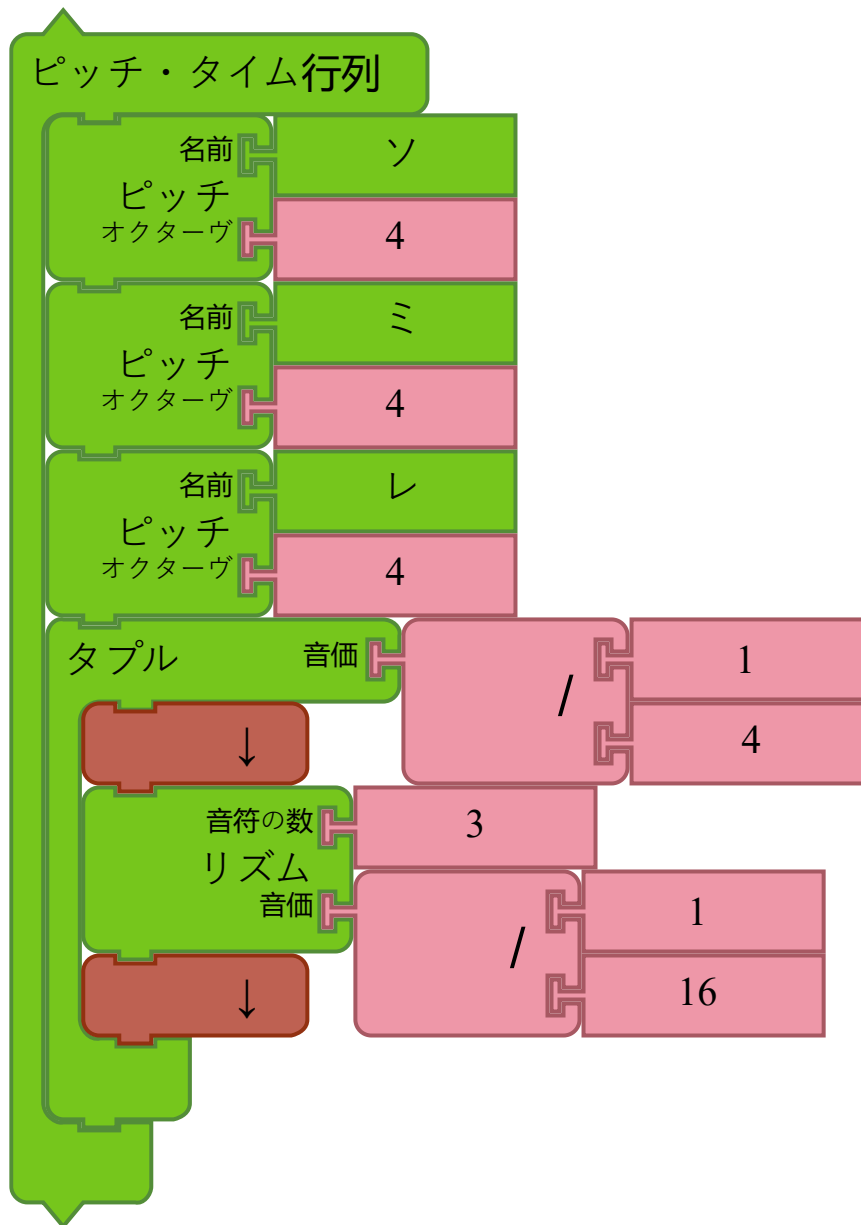
4.2.3 タプルのこと



Solfa							
sol							
mi							
re							
tuplet note values	1/12	1/12	1/12				
tuplet value	3						
rhythmic note values	1/4 ♩						

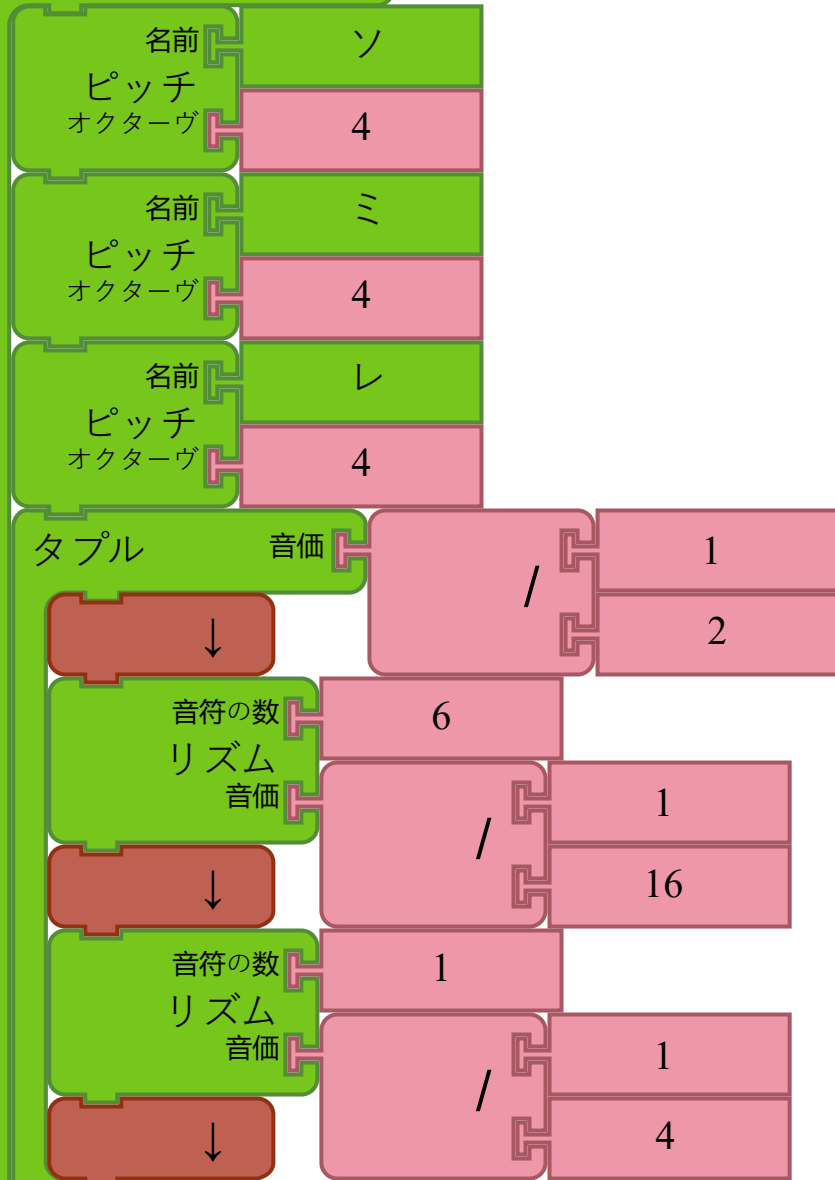
Tuplets are a collection of notes that get scaled to a specific duration. Using tuplets makes it easy to create groups of notes that are not based on a power of 2.








In the example above, three quarter notes—defined in the *Simple Tuplet* block—are played in the time of a single quarter note. The result is three twelfth notes. (This form, which is quite common in music, is called a *triplet*. Other common tuplets include a *quintuplet* and a *septuplet*.)



In the example above, the three quarter notes are defined in the *リズム* block embedded in the *Tuplet* block. As with the *Simple Tuplet* example, they are played in the time of a single quarter note. The result is three twelfth notes. This more complex form allows for intermixing multiple *リズム・ブロック* within single tuplet.

ピッチ・タイム行列



Solfa							
sol ₄							
mi ₄							
re ₄							
tuplet note values	1/20	1/20	1/20	1/20	1/20	1/20	1/5
tuplet value	7						
rhythmic note values	1/2 ♩						

In the example above, the two *リズム* blocks are embedded in the *Tuplet* block, resulting in a more complex *リズム*.

Note: You can mix and match *リズム* blocks and *Tuplet* blocks when defining your matrix.

4.2.4 What is a tuplet?


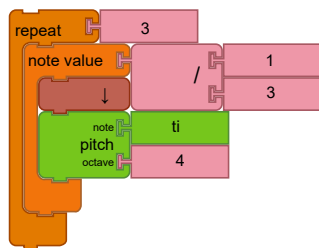
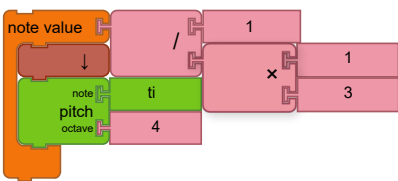

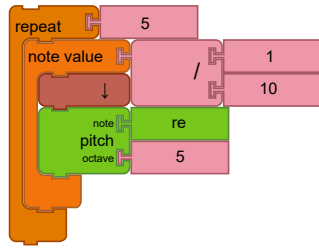
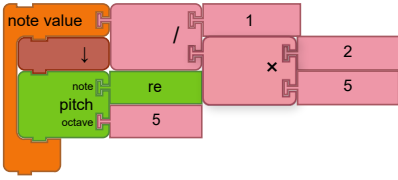
Using Tuplets

A tuplet is a specific group of notes played in a condensed amount of time.

x= power of the note* y= tuplet value

Formula: $\frac{1}{2^{x*} y} = \text{resulting note value}^{**}$

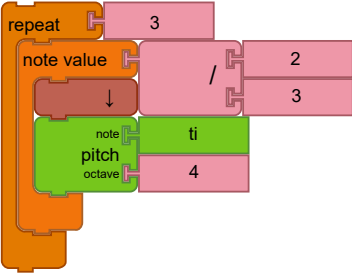
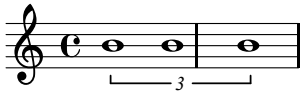
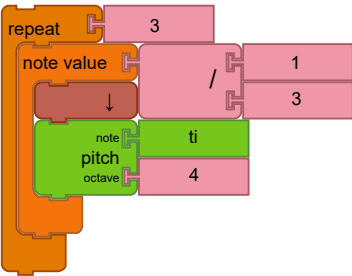

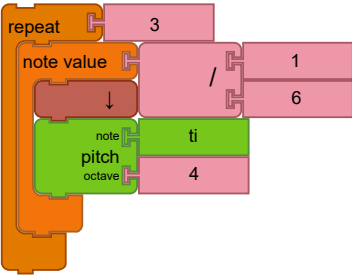

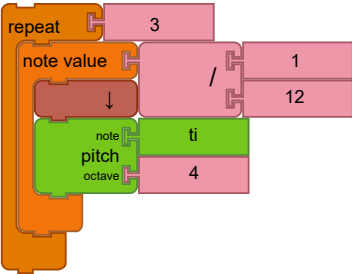

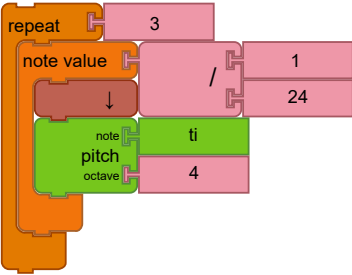

Examples:

Western Notation	Music Blocks Tuplet	Music Block Math for Tuplet
 <p>x= 0 and y=3</p> $\frac{1}{2^0 \times 3} = \frac{1}{1 \times 3} = \frac{1}{3}$		
 <p>x= 1 and y=5</p> $\frac{1}{2^1 \times 5} = \frac{1}{2 \times 5} = \frac{1}{10}$		

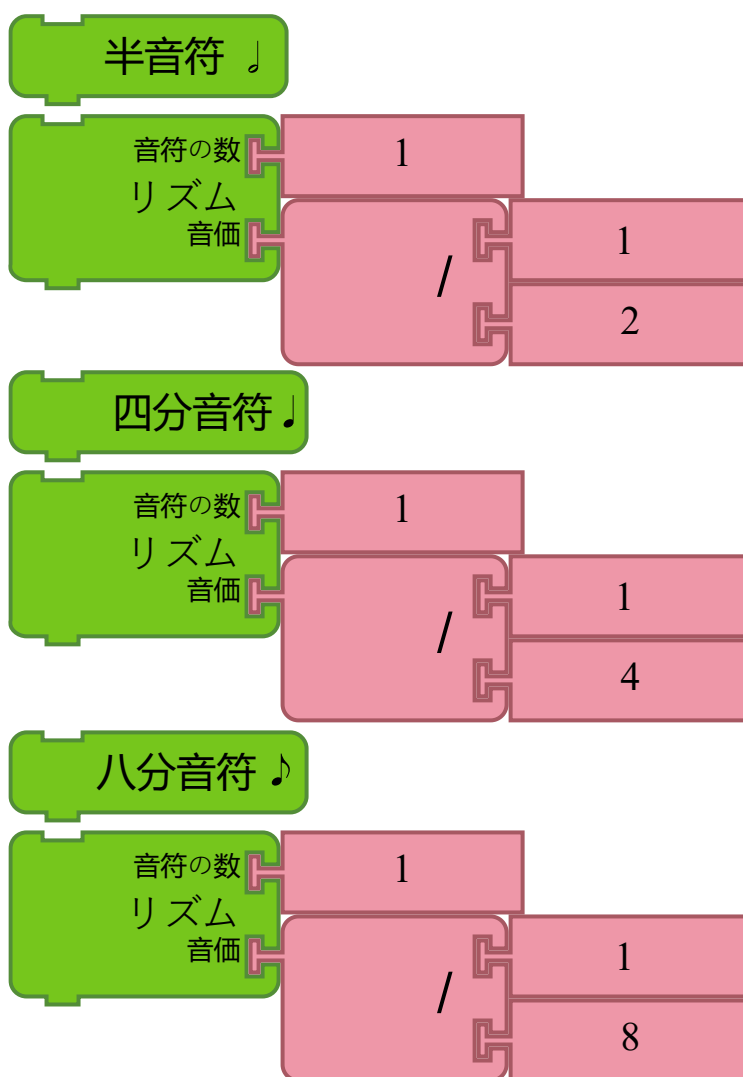
*The power of the note occurs as follows:

longa= -3, breve= -2, whole= -1, half=0, quarter=1, eighth=2, sixteenth=3, thirty-second=4, and continues in this pattern.

**Different tuplet values produce different rhythmic qualities when mixed with note values of different tuplet values.

Power of Two	Music Blocks Tuplet	Tuplet in Western Notation
$\frac{1}{2^{-1} \times 3} = \frac{2}{3}$		
$\frac{1}{2^0 \times 3} = \frac{1}{3}$		
$\frac{1}{2^1 \times 3} = \frac{1}{6}$		
$\frac{1}{2^2 \times 3} = \frac{1}{12}$		
$\frac{1}{2^3 \times 3} = \frac{1}{24}$		

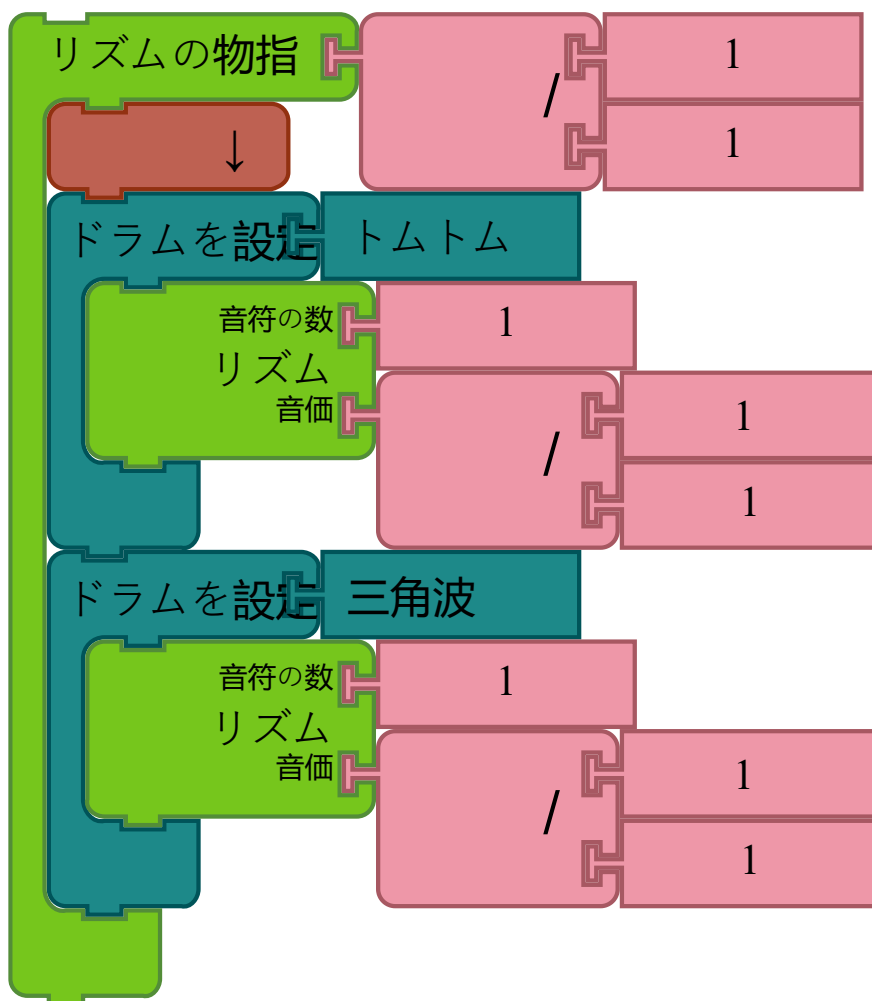
4.2.5 Using individual notes in the matrix



You can also use individual notes when defining the grid. These blocks will expand into *リズム* blocks with the corresponding values.

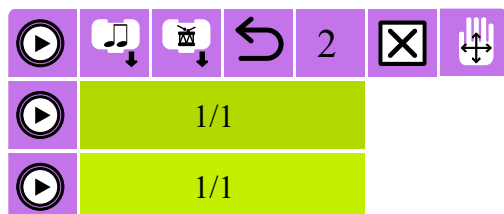
4.3 リズム・ブロックでリズムを

The *Rhythm Ruler* block is used to launch a *ウィジェット* similar to the *ピッチ・タイム行列* block. The *ウィジェット* can be used to generate rhythmic patterns.

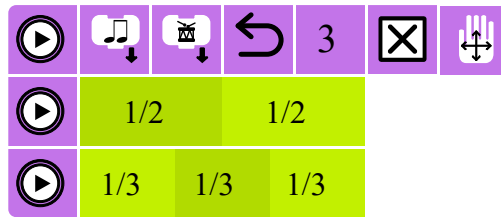


The argument to the *Rhythm Ruler* block specifies the duration that will be subdivided to generate a rhythmic pattern. By default, it is 1 / 1, e.g., a whole note.

The *Set* ドラム blocks contained in the clamp of the *Rhythm Ruler* block indicates the number of リズム・ブロック to be defined simultaneously. By default, two リズム・ブロック are defined. The embedded リズム blocks define the initial subdivision of each rhythm ruler.



When the *Rhythm Ruler* block is クリックed, the *Rhythm Ruler* ウィジェット is opened. It contains a row for each リズム ruler. An input in the top row of the ウィジェット is used to specify how many subdivisions will be created within a cell when it is クリックed. By default, 2 subdivisions are created.



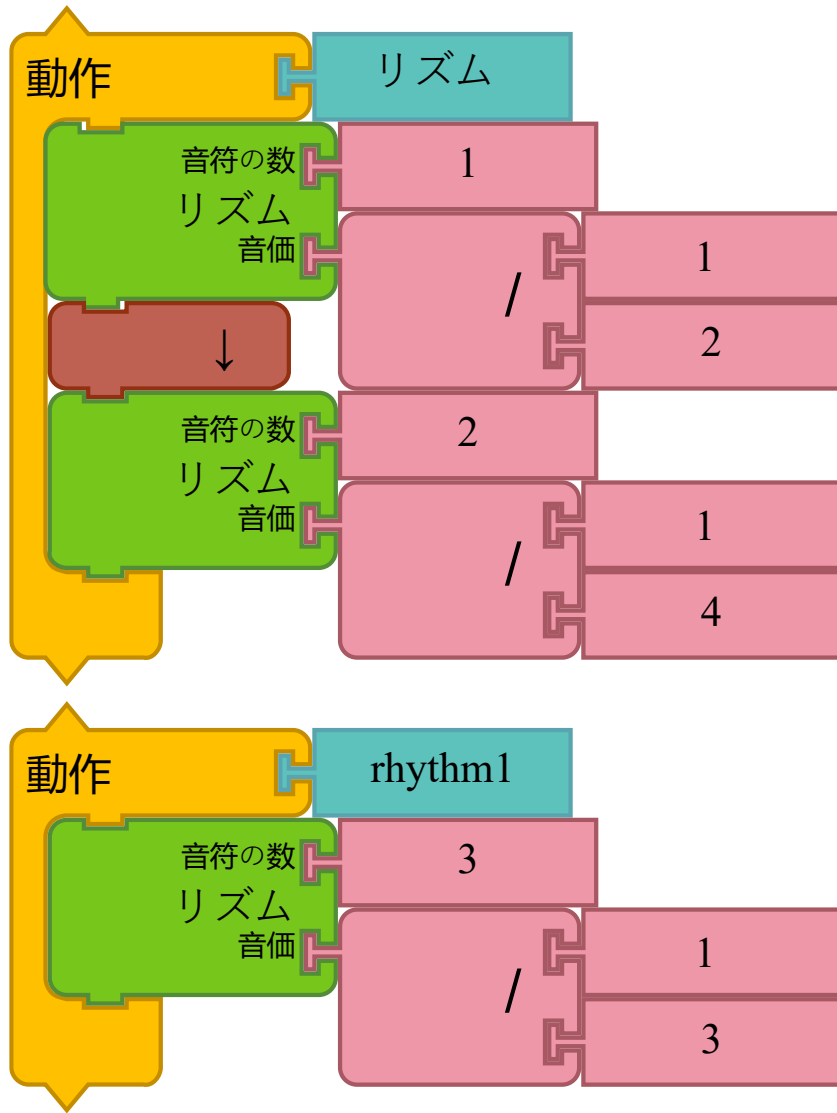
As shown in the above figure, the top rhythm ruler has been divided into two half-notes and the bottom rhythm ruler has been divided into three third-notes. クリックing on the *Play* button to the left of each row will playback the リズム using a ドラム for each 拍子. The *Play-all* button on the upper-left of the ウィジェット will play back all リズム・ブロック simultaneously.



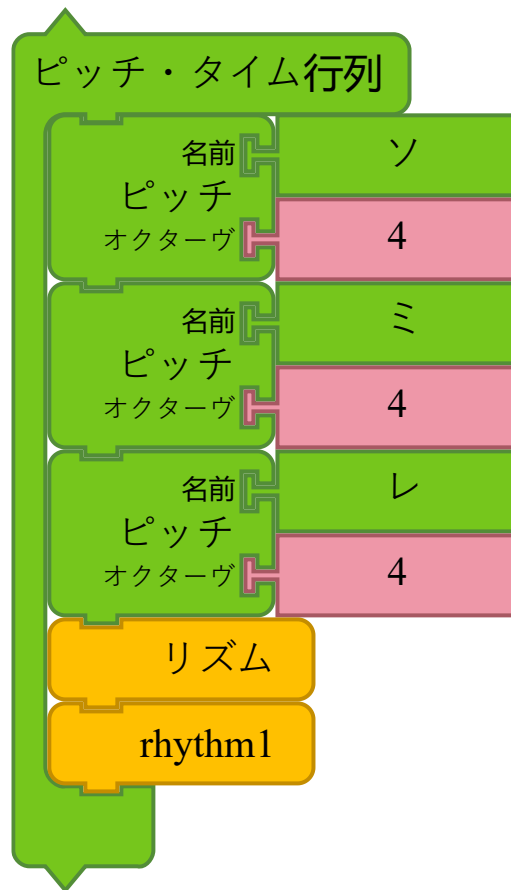
The リズム can be further subdivided by クリックing in individual cells. In the example above, two quarter-notes have been created by クリックing on one of the half-notes.



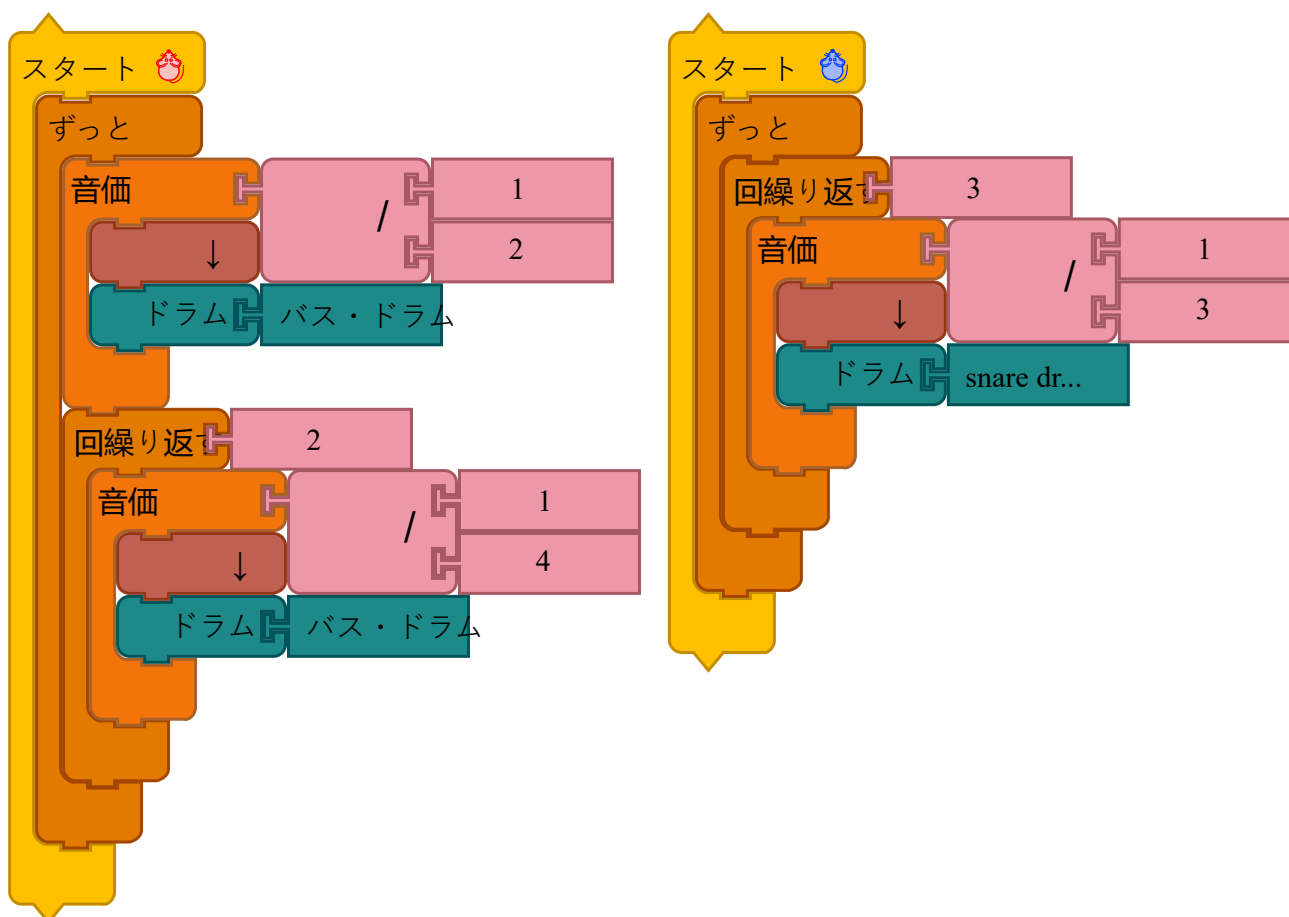
By dragging across multiple cells, they become タイd. In the example above, two third-notes have been タイd into one two-thirds-note.



The Save stack button will export リズム stacks.



These stacks of リズム・ブロック can be used to define rhythmic patterns used with the ピッチ・タイム行列 block.



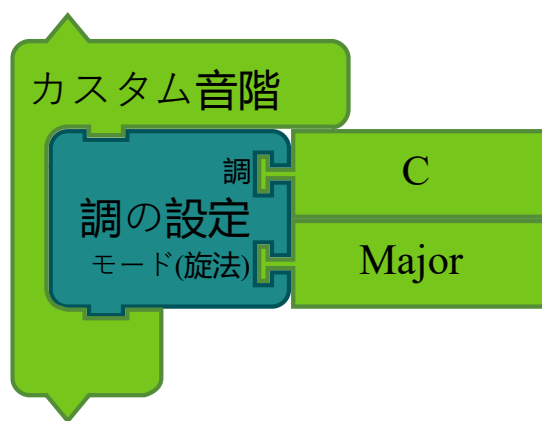
The ドラムループを保存 button will export スタート stacks that will play the リズム・ブロック as ドラムループ(既に再生 <=== check this!) machines.

4.4 音楽の音階とモード

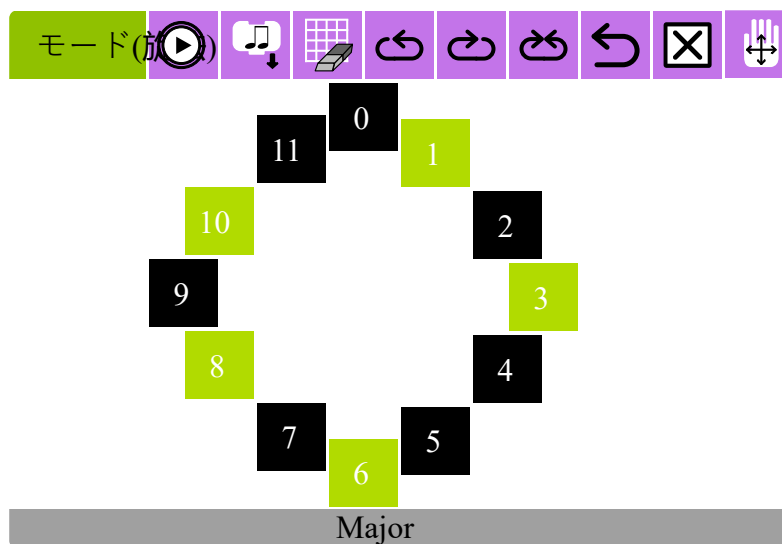
Musical 音階とモード are used to specify the relationship between intervals (or steps) in a scale. Since Western music is based on 12 half-steps per octave, 音階とモード specify how many half steps there are between each note in a scale.

By default, ミュージック・ブロックス uses the 長 mode, which, in the Key of C, maps to the white keys on a piano. The intervals in the 長 mode are 2, 2, 1, 2, 2, 2, 1. Many other common 音階とモード are built into ミュージック・ブロックス, including, of course, 短音階 (<===check this) mode, which uses 2, 1, 2, 2, 1, 2, 2 as its intervals.

Note that not every mode uses 7 intervals per octave. For example, the *Chromatic* mode uses 11 intervals: 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1. The *Japanese* mode uses only 5 intervals: 1, 4, 2, 3, 2], . What is important is that the sum of the intervals in an octave is 12 half-steps.



The *Mode* ウィジェット lets you explore 音階とモード and generate custom 音階とモード. You invoke the ウィジェット with the *Custom mode* block. The mode specified in the *Set key* block will be the default mode when the ウィジェット launches.



In the above example, the ウィジェット has been launched with *Major* mode (the default). Note that the notes included in the mode are indicated by the black boxes, which are arrayed in a circular pattern of twelve half-steps to complete the octave.

Since the intervals in the *Major* mode are 2, 2, 1, 2, 2, 2, 1, the notes are 0, 2, 4, 5, 7, 9, 11, and 12 (one octave above 0).

The ウィジェット controls run along the toolbar at the top. From left to right are:

Play all, which will play a scale using the current mode;

Save, which will save the current mode as the *Custom* mode and save a stack of ピッチ blocks that can be used with the *Pitch-time Matrix* block;

Rotate counter-clockwise, which will rotate the mode counter-clockwise (See the example below);

Rotate clockwise, which will rotate the mode clockwise (See the example below);

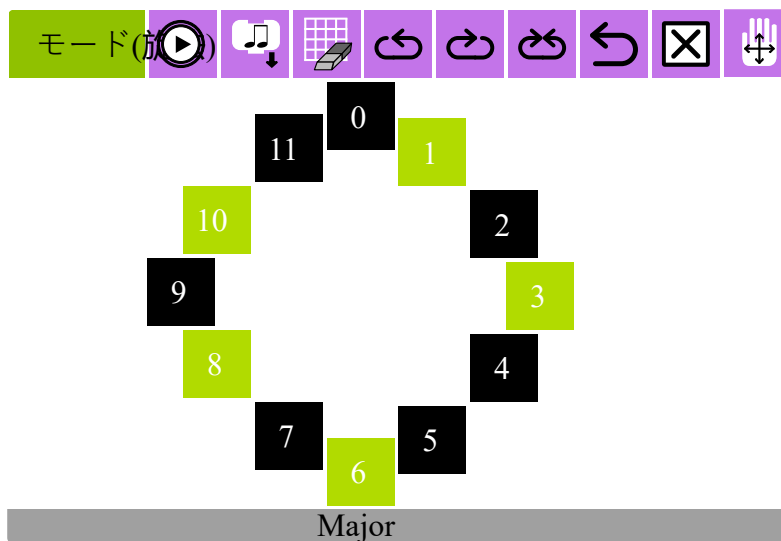
転回, which will 転回 the mode (See the example below);

Undo, which will restore the mode to the previous version; and

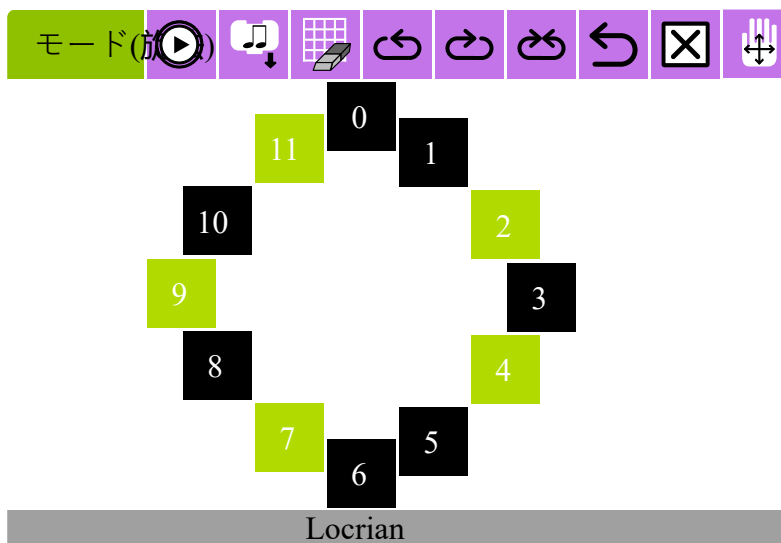
Close, which will close the ウィジェット.

You can also クリック on individual notes to activate or deactivate them.

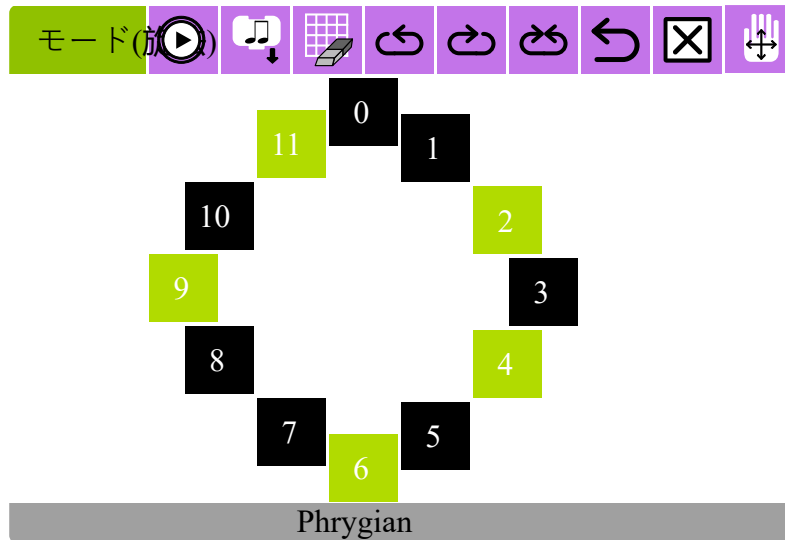
Note that the mode inside the *Custom mode* block is updated whenever the mode is changed inside the ウィジェット.



In the above example, the *Major* mode has been rotated clockwise, transforming it into *Dorian*.

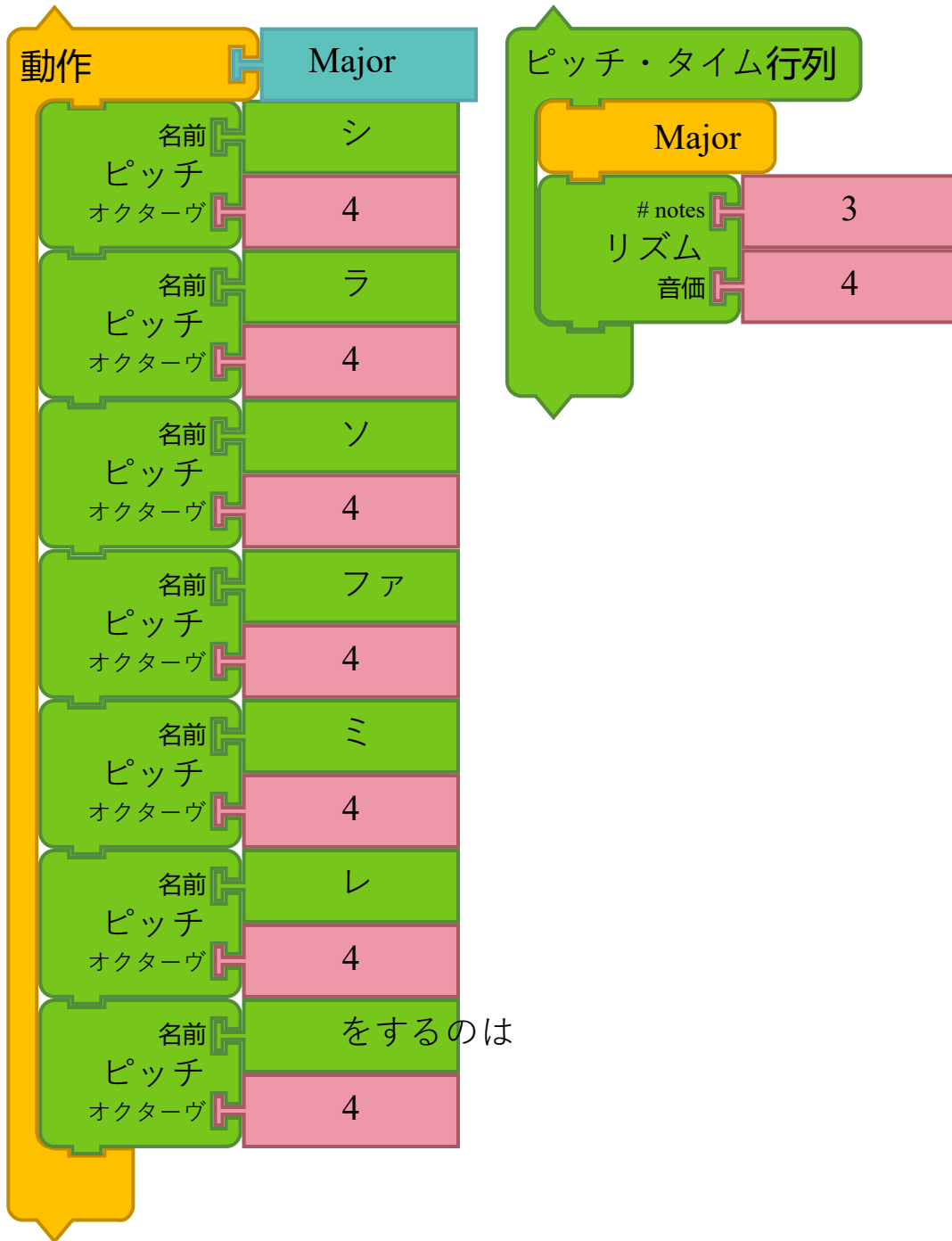


In the above example, the *Major* mode has been rotated counter-clockwise, transforming it into *Locrian*.



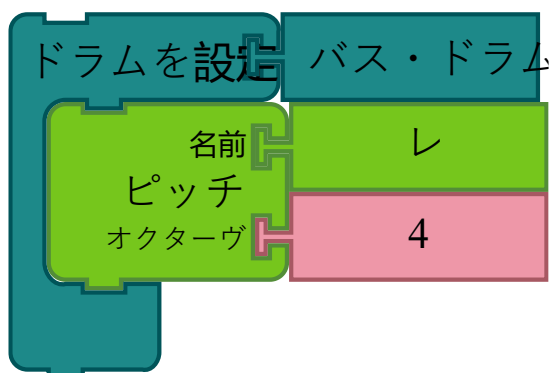
In the above example, the *Major* mode has been 転回ed, transforming it into *Phrygian*.

Note: The build-in 音階とモード in ミュージック・ブロックス can be found in [musicutils.js](https://github.com/sugarlabs/musicblocks/blob/master/js/utls/musicutils.js#L68) (<https://github.com/sugarlabs/musicblocks/blob/master/js/utls/musicutils.js#L68>).

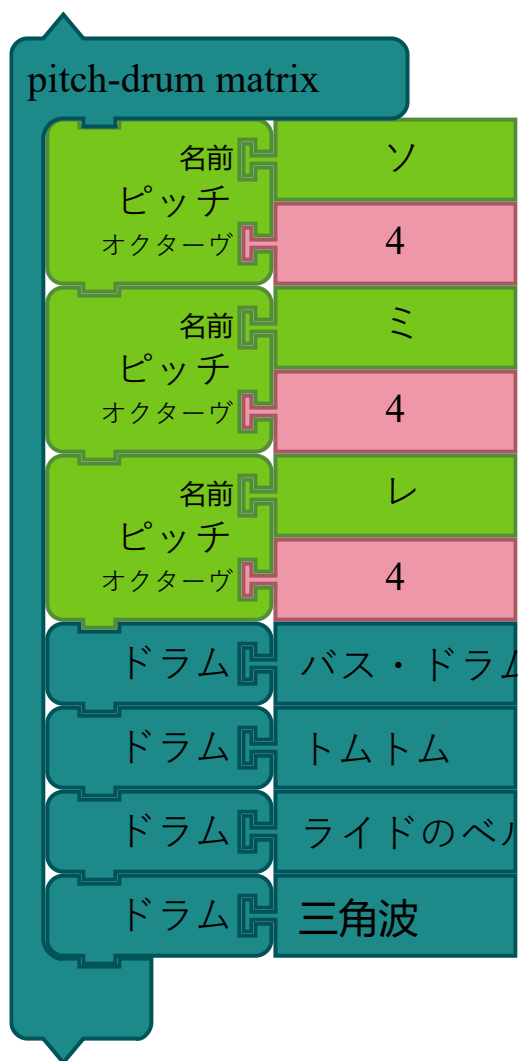


The Save button exports a stack of blocks representing the mode that can be used inside the `ピッチ・タイム行列` block.

4.5 ピッチ・ドラム・グラフ

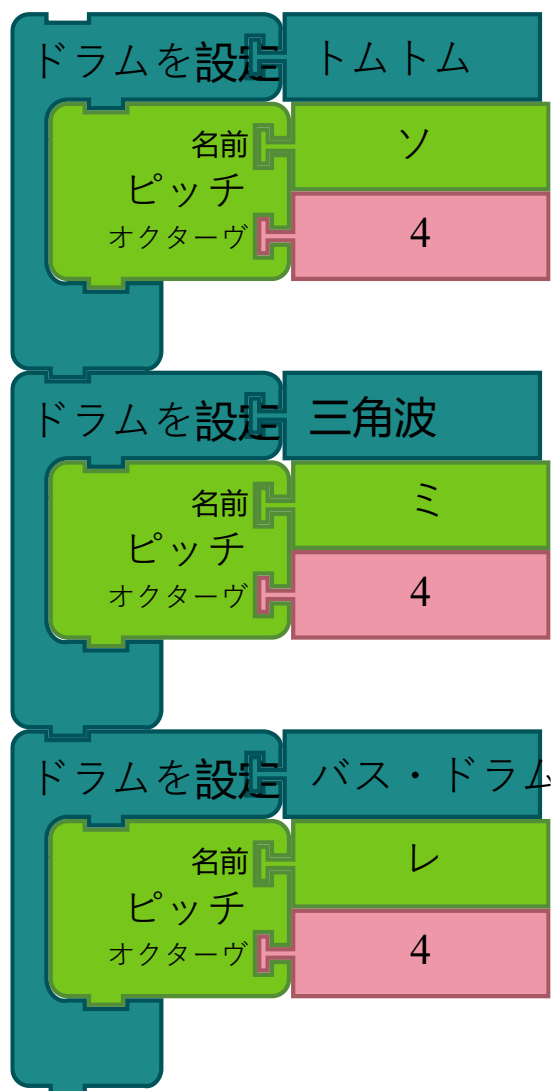


The *Set ドラム* block is used to map the enclosed pitches into ドラム sounds. ドラム sounds are played in a monopitch using the specified ドラム sample. In the example above, a kick ドラム will be substituted for each occurrence of a レ 4.



Solfa						
sol ₄						
mi ₄						
re ₄						

Solfa						
sol ₄						
mi ₄						
re ₄						

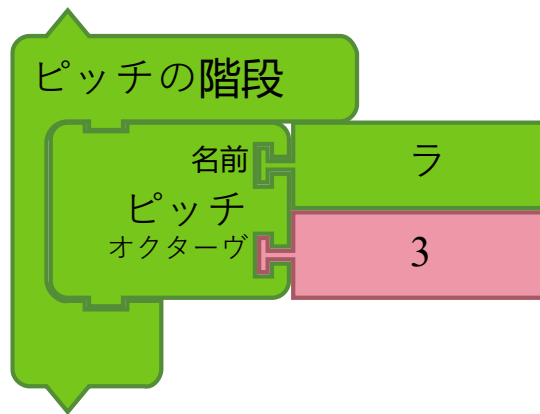


As an experience for creating mapping with the *Set* ドラム block, we provide the *Drum-Pitch Matrix*. You use it to map between pitches and drums. The output is a stack of *Set Dum* blocks.

4.6 音程の関係をことをピッチの階段一で発見

The *Pitch Staircase* block is used to launch a ウィジェット similar to the *ピッチ・タイム行列*, which can be used to generate different pitches using a given pitch and musical proportion.

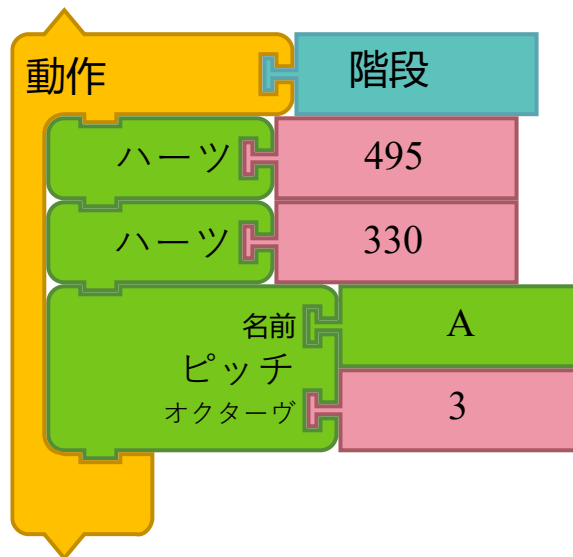
The *ピッチ* blocks contained in the clamp of the *Pitch Staircase* block define the pitches to be initialized simultaneously. By default, one pitch is defined and it have default note "la" and octave "3".



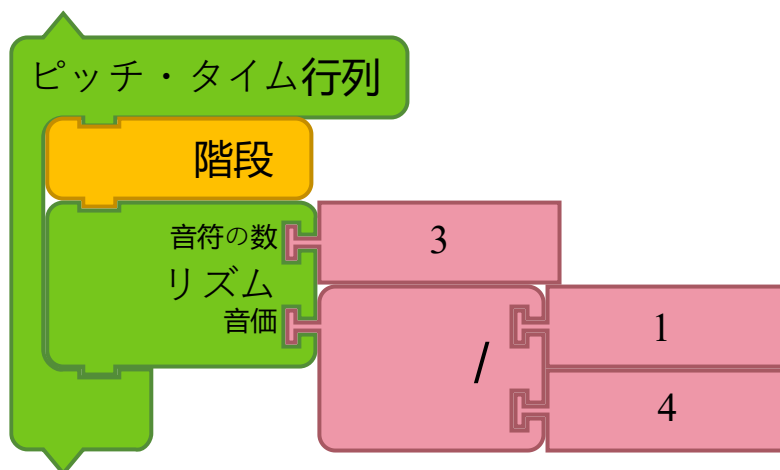
When *Pitch Staircase* block is クリックed, the *Pitch Staircase* ウィジェット is initialized. The ウィジェット contains row for every *ピッチ* block contained in the clamp of the *Pitch Staircase* block. The input fields in the top row of the ウィジェット specify the musical proportions used to create new pitches in the staircase. The inputs correspond to the numerator and denominator in the proportion respectively. By default the proportion is 3:2.

クリックing on the *Play* button to the left of each row will playback the notes associated with that step in the *ピッチの階段*. The *Play-all* button on the upper-left of the ウィジェット will play back all the pitch steps simultaneously. A second *Play-all* button to the right of the stair plays in increasing order of frequency first, then in decreasing order of frequency as well, completing a scale.

The *Save stack* button will export pitch stacks. For example, in the above configuration, the output from pressing the *Save stack* button is shown below:



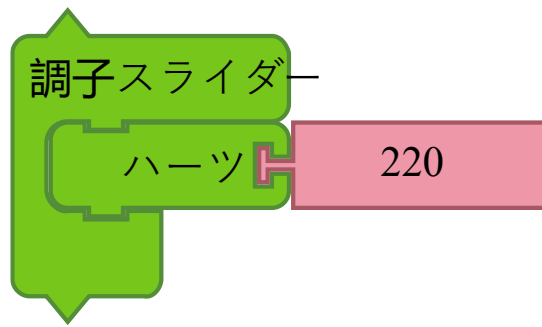
These stacks can be used with the `ピッチ・タイム行列` block to define the rows in the matrix.



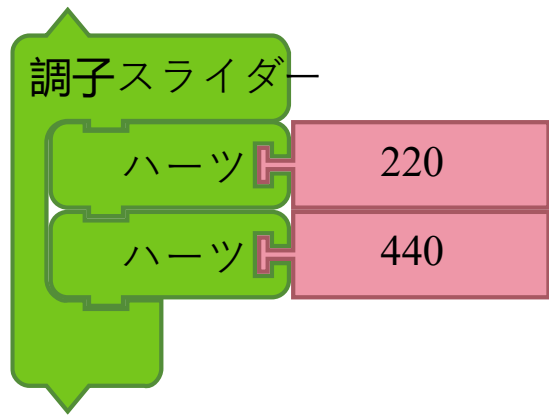
4.7 ピッチ・スライダーで何のピッチでも発見

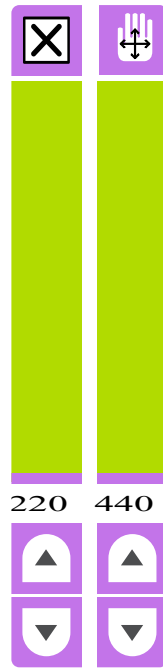
The `ピッチ・スライダー` block is used to launch a `ウィジェット` that is used to generate arbitrary pitches. It differs from the `Pitch Staircase` `ウィジェット` in that it is used to create frequencies that vary continuously within the range of a specified octave.

Each `Sine` block contained within the clamp of the `ピッチ・スライダー` block defines the initial pitch for an octave.

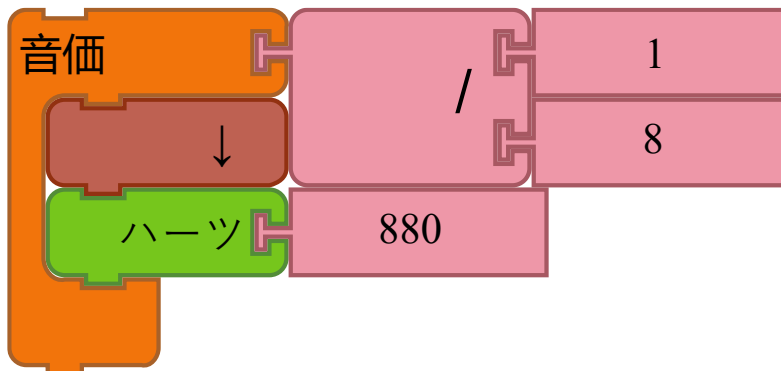
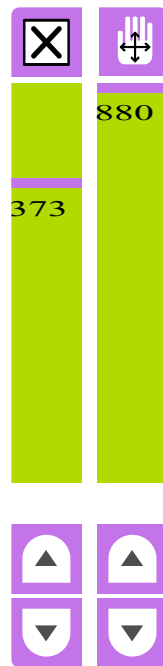


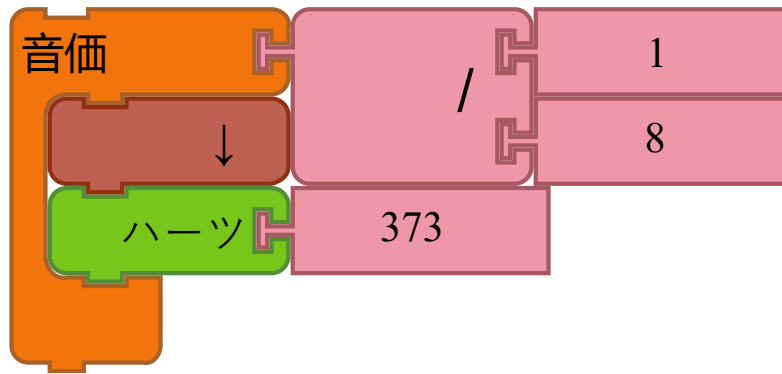
When the *ピッチ・スライダー* block is clicked, the *ピッチ・スライダー* ウィジェット is initialized. The ウィジェット will have one column for each *Sine* block in the clamp. Every column has a slider that can be used to move up or down in frequency, continuously or in intervals of 1/12th of the starting frequency. The mouse is used to move the frequency up and down continuously. Buttons are used for intervals. Arrow keys can also be used to move up and down, or between columns.





Clicking in a column will extract the corresponding *Note* blocks, for example:

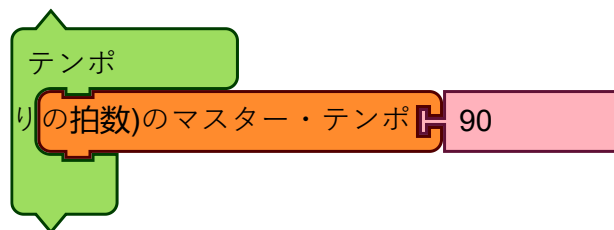




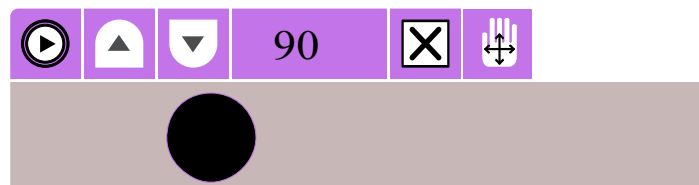
4.8 テンポを変えること

The **テンポ** block is used to launch a **ウィジェット** that enables the user to visualize **テンポ**, defined in **拍子s per minute (BPM)**. When the **テンポ** block is **クリック**ed, the **テンポ** **ウィジェット** is initialized.

The *Master 拍子 per Minute* block contained in the clamp of the **テンポ** block sets the initial **テンポ** used by the **ウィジェット**. This determines the speed at which the ball in the **ウィジェット** moves back and forth. If BPM is 60, then it will take one second for the ball to move across the **ウィジェット**. A round-trip would take two seconds.



The top row of the **ウィジェット** holds the *Play/pause* button, the *Speed up* and *Slow down* buttons, and an input field for updating the **テンポ**.

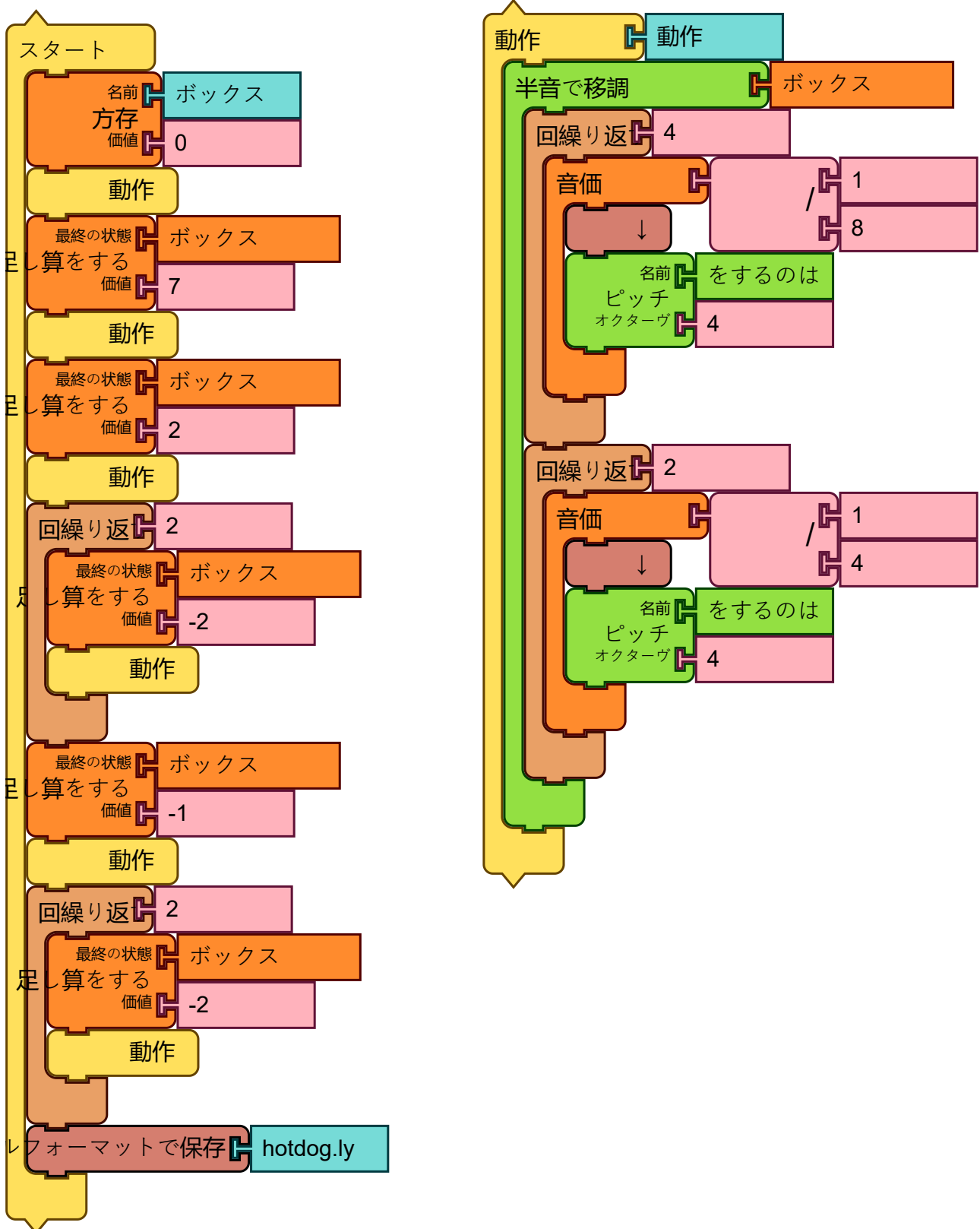


You can also update the **テンポ** by **クリック**ing twice in spaced succession in the **ウィジェ** **ット**: the new BPM is determined as the time between the two **クリック**s. For example, if there is 1/2 seconds between **クリック**s, the new BPM will be set as 120.

ミュージック・ブロックスから以遠

[前のトピック \(4. ウィジェット\)](#) | [目次に戻る](#)

ミュージック・ブロックス is a waypoint, not a destination. One of the goals is to point the learner towards other powerful tools. One such tool is [Lilypond \(http://lilypond.org\)](http://lilypond.org), a music engraving program.



The *Save as Lilypond* block will transcribe your composition. The output of the program above is saved to `Downloads/hotdog.ly`. There is also a *Save as Lilypond* button on the secondary toolbar.



```
\version "2.18.2"

mouse = {
c'8 c'8 c'8 c'8 c'4 c'4 g'8 g'8 g'8 g'8 g'4 g'4 a'8 a'8 a'8 a'8 a'4
a'4 g'8 g'8 g'8 g'8 g'4 g'4 f'8 f'8 f'8 f'8 f'4 f'4 e'8 e'8 e'8 e'8
e'4 e'4 d'8 d'8 d'8 d'8 d'4 d'4 c'8 c'8 c'8 c'8 c'4 c'4
}

\score {
<<
\new Staff = "treble" {
\clef "treble"
\set Staff.instrumentName = #"mouse" \mouse
}
>>
\layout {}
}
```



[ライブで再生 \(https://musicblocks.sugarlabs.org/index.html?id=1523043053377623&run=True\)](https://musicblocks.sugarlabs.org/index.html?id=1523043053377623&run=True)