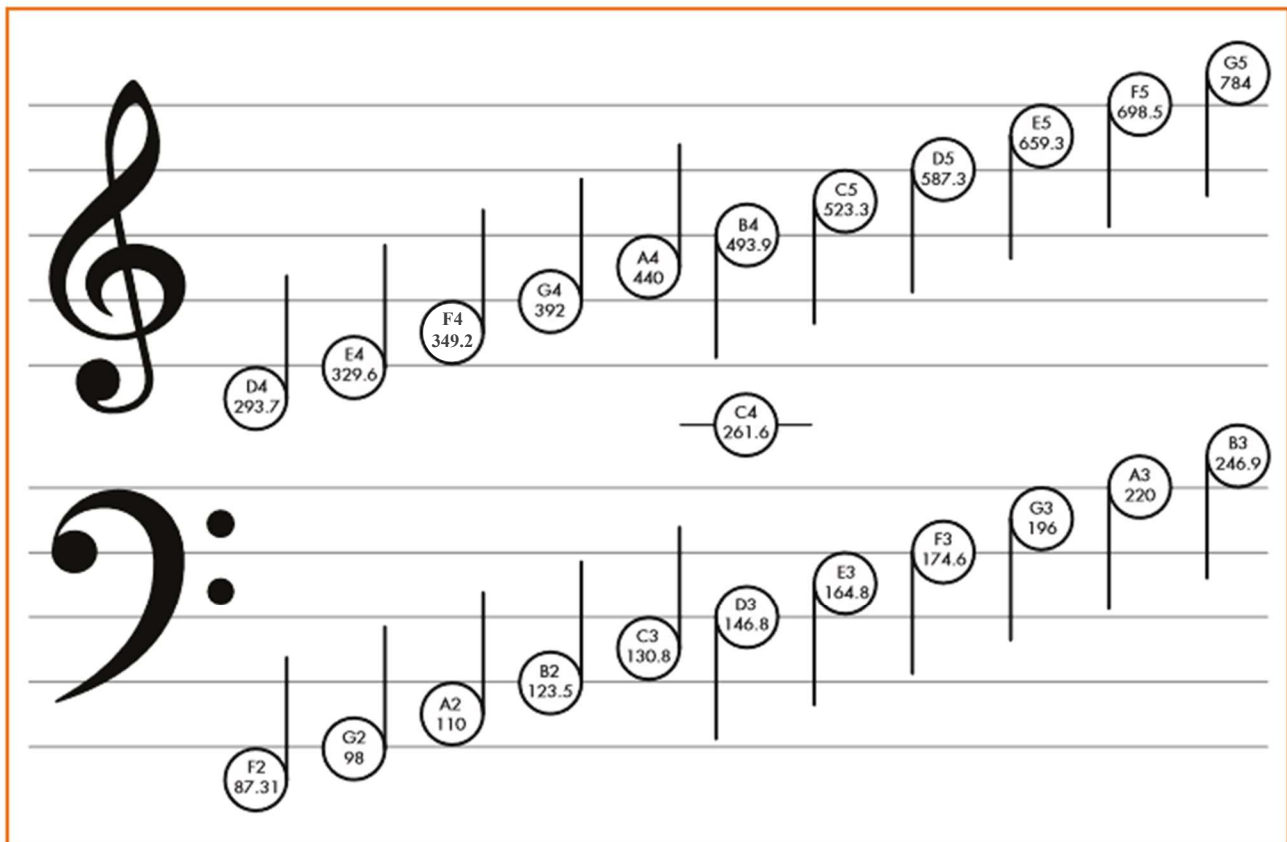


Music Note Frequencies

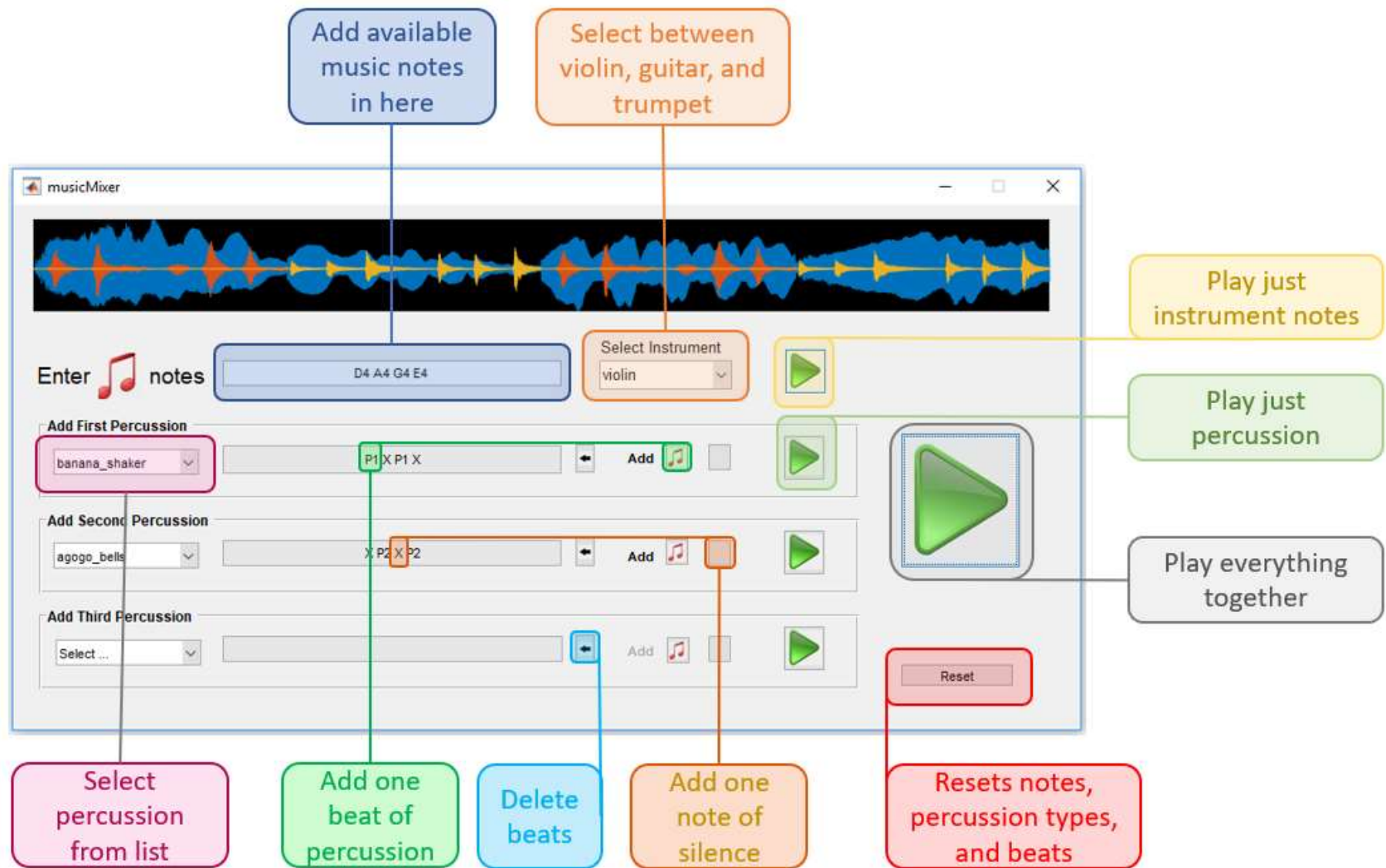


| | C | C# | D | E \flat | E | F | F# | G | G# | A | B \flat | B |
|---|-------|-------|-------|-----------|-------|-------|-------|-------|-------|-------|-----------|-------|
| 0 | 16.35 | 17.32 | 18.35 | 19.45 | 20.60 | 21.83 | 23.12 | 24.50 | 25.96 | 27.50 | 29.14 | 30.87 |
| 1 | 32.70 | 34.65 | 36.71 | 38.89 | 41.20 | 43.65 | 46.25 | 49.00 | 51.91 | 55.00 | 58.27 | 61.74 |
| 2 | 65.41 | 69.30 | 73.42 | 77.78 | 82.41 | 87.31 | 92.50 | 98.00 | 103.8 | 110.0 | 116.5 | 123.5 |
| 3 | 130.8 | 138.6 | 146.8 | 155.6 | 164.8 | 174.6 | 185.0 | 196.0 | 207.7 | 220.0 | 233.1 | 246.9 |
| 4 | 261.6 | 277.2 | 293.7 | 311.1 | 329.6 | 349.2 | 370.0 | 392.0 | 415.3 | 440.0 | 466.2 | 493.9 |
| 5 | 523.3 | 554.4 | 587.3 | 622.3 | 659.3 | 698.5 | 740.0 | 784.0 | 830.6 | 880.0 | 932.3 | 987.8 |
| 6 | 1047 | 1109 | 1175 | 1245 | 1319 | 1397 | 1480 | 1568 | 1661 | 1760 | 1865 | 1976 |
| 7 | 2093 | 2217 | 2349 | 2489 | 2637 | 2794 | 2960 | 3136 | 3322 | 3520 | 3729 | 3951 |
| 8 | 4186 | 4435 | 4699 | 4978 | 5274 | 5588 | 5920 | 6272 | 6645 | 7040 | 7459 | 7902 |

BYTES AND BEATS

Handout

Using the Music Mixer App



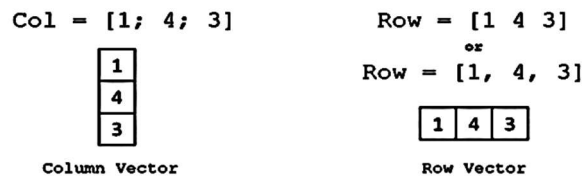
MATLAB Vocabulary

MATLAB – A programming language and computing environment used by engineers and scientists around the world to analyze data and model applications such as Mars rover, airplanes, playing music, and many more.

Command Window – The writing area in MATLAB that has the “>>” prompt. Single line commands can be executed directly from here.

Workspace – The area that contains all the variables you create and store in memory during a MATLAB session.

Numeric arrays – an ordered arrangement of numbers. Arrays are created within square brackets [].



Strings – A word or a sentence enclosed within double quotes " ".

E.g. `>> str = "I love MATLAB"`

Indexing – Extracting values from an array using parenthesis ()

Live Script – A script is code file that contains a sequence of MATLAB statements and commands but cannot take inputs or give outputs. Live Scripts are interactive documents that can contain formatted text and images along with code. The output of a live script is contained and displayed within the file.

Function – Similar to scripts, a function contains a sequence of MATLAB commands. Unlike scripts, a function can accept inputs and produce outputs.

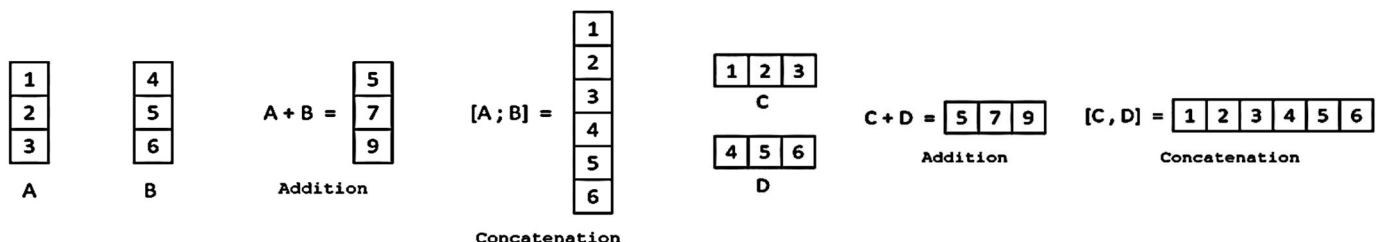
For loop – Executes a block of code repeatedly for a specific number of times (e.g., the following executes 10 times).

E.g. `for k = 1:10 ... end`

If-else-end – Checks whether a certain condition is true. If true, the code executes the block of code inside the if branch; otherwise it executes the block of code inside the else branch.

E.g. `if x < 0 ... else ... end`

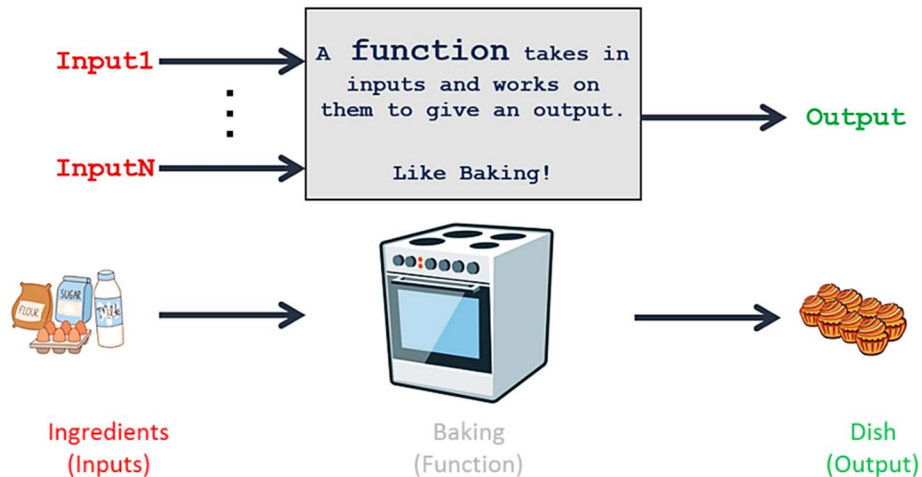
Adding and concatenating vectors –



Figures and Callbacks – Figures are containers for your app's graphics. Figures and other graphics have some properties that correspond to specific user actions. Each of these properties can be associated with a user-defined function. Inside this callback function, you can specify what the app must do when the associated user action occurs. Graphics callback functions always take two inputs: `source` and `event`.

MATLAB Functions

Output = **function**(**Input1**, ..., **InputN**)



Functions can have zero or more inputs and outputs.

| Function | Syntax / Usage Example | Description |
|-----------------------------------|---|---|
| General MATLAB | | |
| clc | <code>clc</code> | Clears the Command Window |
| clear | <code>clear</code> | Clears variables from Workspace |
| uifigure | <code>uifigure("Name", "My figure")</code> | Creates a new figure window with name 'My figure' |
| close | <code>close</code> <code>close all</code> | Closes the last opened figure Closes all open figures |
| pause | <code>pause(0.5)</code> | Pauses MATLAB execution for 0.5 seconds |
| save | <code>save("MyVars.mat")</code> | Saves all the Workspace variables to a file named "MyVars.mat" |
| load | <code>load("MyVars.mat")</code> | Loads all variables from "MyVars.mat" into Workspace |
| Numeric and string vectors | | |
| zeros | <code>X = zeros(5,1)</code> | Creates a column vector with five zeros in it. |
| join | <code>str = ["I", "love", "MATLAB"];</code> <code>joinedStr = join(str)</code> | Joins the strings in the input vector separating them at spaces. <code>joinedStr</code> will be "I love MATLAB" |
| split | <code>str = "I love MATLAB";</code> <code>splitStr = split(str)</code> | Separates a string into a vector of strings at spaces. <code>splitStr</code> will be ["I", "love", "MATLAB"] |
| length | <code>l = length(X)</code> | Returns the length of X. If X is a vector, it returns the number of elements in it. |

| | | |
|--------------------------------|--|--|
| find | <code>k = find(X)</code> | It finds and returns the indices(positions) of non-zero elements in X. |
| Bytes and Beats toolbox | | |
| playNumber | <code>playNumber([1 2 3])</code> | Plays a series of sine frequencies corresponding to the numeric input vector. In this case, the notes played will be C4, D4 and E4. |
| sineSound | <code>sineSound(440, 0.2, 0.5)</code> | Plays a sinusoidal note enveloped in a sine wave. It takes in an input frequency, duration(optional) and amplitude(optional). In this case, a sine tone of 440Hz will be played of 0.2s at half the maximum amplitude. |
| Audio processing | | |
| sound | <code>sound(y, Fs)</code> | Plays the sound represented by vector y at a sampling frequency of Fs. |
| audioread | <code>[y, Fs] = audioread(filename)</code> | Reads audio from the input audio file (e.g. .WAV, .MP3) and returns the corresponding sound vector and sampling frequency. |
| audiowrite | <code>audiowrite(filename, y, Fs)</code> | Writes audio signal from MATLAB to an audio file specified by 'filename' |