The VexTab Tutorial

An N-step program for learning all about [VexTab](#)
by [Mohit Muthanna Cheppudira](#).

What is VexTab?

VexTab is a language that allows you to easily create, edit, and share standard notation and guitar tablature. Unlike ASCII tab, which is designed for readability, VexTab is designed for writeability.

Step 1: The Stave

The `tabstave` keyword is used to create a new tab stave. The green code box below is editable - try adding another stave by typing `tabstave` on a new line.

`tabstave` takes keyword arguments. For example, the string `tabstave notation=true` renders a standard notation stave above the tab stave. Try this on the tab stave below.
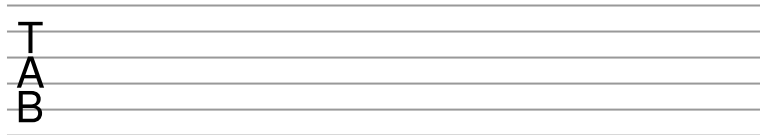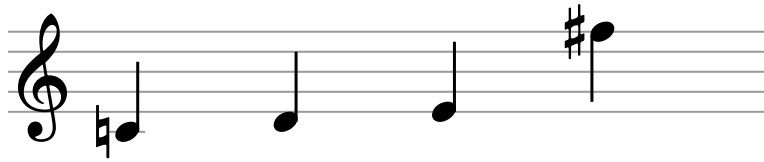
vexflow.com

```
tabstave
```

Step 2: Add some notes

The `notes` keyword can be used to add notes. Notes can be represented as `note/octave` or as `fret/string`. To add a long line of notes on the same string (or the same octave), you can use the format `fret-fret-fret/string` (or `note-note-note/octave`).
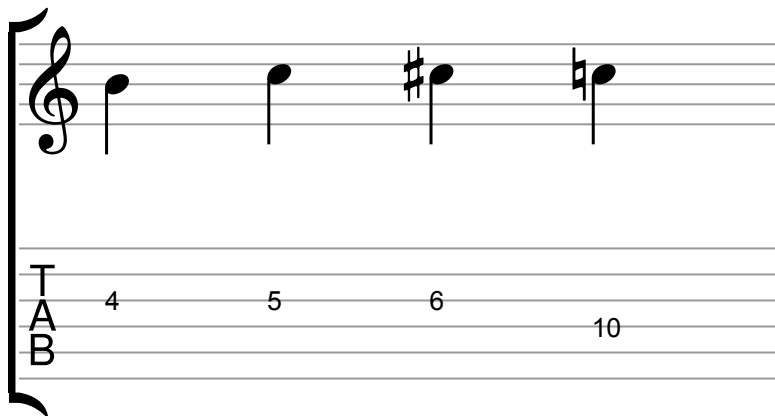
Try editing the notation below. Notice how the notes are justified in the stave. Notes can be suffixed by the following accidentals: `#, ##, @, @@,` and `n` signifying sharp, double-sharp, flat, double-flat, and natural, respectively.

vexflow.com

```
tabstave notation=true tablature=false
notes Cn-D-E/4 F#/5
```

Here's some guitar tab. Try adding more frets below.

vexflow.com

```
tabstave notation=true
notes 4-5-6/3 10/4
```

Step 3: Stave keywords

`tabstave` takes keyword arguments that can be combined.

| keyword	values
notation	true/false
tablature	true/false
clef	treble, alto, tenor, bass, percussion
key	c, Am, F, Dm, Bb, Gm, Eb, Cm, Ab, Fm, Db, Bbm, Gb, Ebm, Cb, Abm, G, Em, D, Bm, A, F#m, E, C#m, B, G#m, F#, D#m, C#, A#m
time	C, Cl, #/#
tuning	standard, dropd, eb, E/5,B/4,G/4,D/4,A/3,E/3

For example

- `tabstave notation=true` renders a standard notation stave above the tab stave.
- `tabstave notation=true tablature=false` only renders the notation stave.
- `tabstave notation=true clef=alto` changes the clef to alto.
- `tabstave notation=true clef=bass key=C# time=C|` would create two staves (one notation, one tab), bass clef in the key of C# and half common time

Try these on the tab stave below.
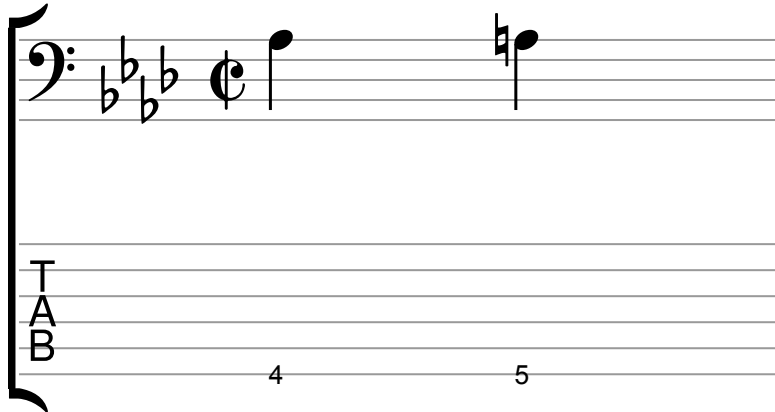
vexflow.com

```
tabstave notation=true clef=bass key=Ab
time=C|
notes 4-5/6
```

Step 4: Add rests and bar-lines

You can render a rest on the notation stave by typing `##`. Rests are positioned automatically, but you can specify a position by inserting a number in between the hashes, e.g., `#4#`. The position ranges from `0` to `9` (bottom to top.)

The | character places a bar-line between the notes. You can add as many bars as you want to a stave.

vexflow.com

```
tabstave notation=true
notes 4-5-6/3 ## | 5-4-2/3 2/2

tabstave notation=true tablature=false
notes C-D-E/4 #0# | C-D-E-F/4
```

Bar-lines can get fancy. You can render double bars, repeats, end bars using the various barline codes.

=|| Double Bar
=|: Repeat Begin
=:| Repeat End
=:: Double Repeat
=|= End Bar

vexflow.com

```
tabstave notation=true tablature=false
notes 4-5-6/3 ## =|: 5-4-2/3 2/2 =:|

tabstave notation=true tablature=false
notes C-D-E/4 #0# =:: C-D-E-F/4 =|=
```

Step 5: Bend a few notes

To draw bends, separate two frets with a `b` character. The difference between the frets determines how much to bend. E.g., `10b12` is a full-step bend.

If you separate three frets with a bend, and the first fret is the same as the last fret, a bend-and-release is displayed. E.g., `10b12b10`.
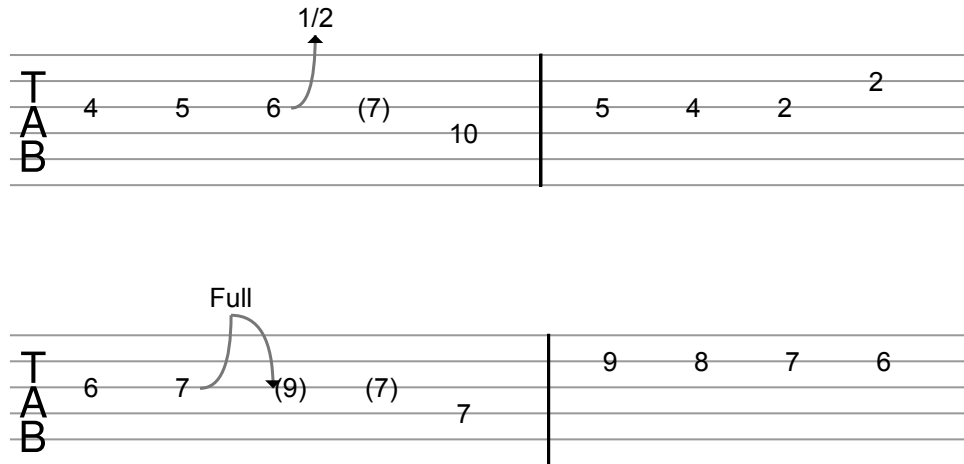
vexflow.com

```
tabstave
notes 4-5-6b7/3 10/4 | 5-4-2/3 2/2

tabstave
notes 6-7b9b7/3 7/4 | 9-8-7-6/2
```

Step 6: Mute notes, show strokes, and add vibrato

You can render a muted note by replace the fret (or note name) with an `x`.

Adding a `v` to the end of a note signifies a vibrato. A capital v (`V`) signifies a harsh vibrato. You can also add a vibrato to a bend.

You can also add a `u` or a `d` to designate upstrokes or downstrokes respectively.

vexflow.com

```
tabstave notation=true
notes 4-5-6b7v/3 10/1 | 5d-4u-Xd/3 2v/2
```

Step 7: Add some chords

To render chords, group notes in parenthesis and separate each `string/fret` (or `note/octave`) combination with a period. E.g., `(4/5.5/6)` or `C/4.E/4`.

For tablature, you can bend, slide, etc. individual notes in a chord just like you would for single notes.

vexflow.com

```
tabstave notation=true tablature=false
notes (C/4.E/4.G/4) C-E-G/4

tabstave notation=true
notes (8/2.7b9b7/3) (5b6/2.5b6/3) 7/4 |
notes (5/2.6/3.7/4.7/5)
```

Step 8: Add some hammer-ons, pull-offs, taps, and slides

To render hammer-ons, pull-offs, taps, or slides, use the characters `h`, `p`, `t` or `s`, respectively. For example, to render a hammer-on/pull-off combo from fret 6 to 8 back to 6, use `6h8p6`.

You can also hammer-on, pull-off, tap, and slide between chords.

vexflow.com

```
tabstave
notes (5/2.5/3.7/4) 5h6/3 7/4 |
notes t12p7p5h7/4 7/5 5s3/5

tabstave
notes (8/2.7b9b7/3) (5b6/2.5b6/3)v 7s0/4 |
notes (5/2.6/3.7/4)v

tabstave
notes (5/4.5/5)s(7/4.7/5)s(5/4.5/5) (5/4.5/5)h(7/5) |
notes t(12/5.12/4)s(5/5.5/4) 3b4/5 5V/6
```

Step 9: Add durations and tuplets.

By default, note durations are set to quarter notes. To change the duration of the following notes you can use the colon character `:` followed by a duration code.

For example, `:w` specifies a whole note. The currently available durations are: `w h q 8 16 32`, for whole, half, quarter, eighth, sixteenth, and thirty-second note durations. Suffixing a 'd' to the duration indicates a dotted note. For example, `:qd`.

You can indicate tuplets by enclosing a tuplet number (3, 5, 7, etc.) within caret signs (`^`). VexTab will create a tuplet out of the preceding notes. For example, `:8 4-5-6/4 ^3^` will create a triplet out of three notes.

vexflow.com

```
tabstave notation=true time=4/4 key=Ab
tuning=eb
notes :8 5s7s8/5 ^3^ :q (5/2.6/3)h(7/3) :8d 5/4 :16 5/5
```

Here's another example rendered with durations and standard notation. Notice how you can set the duration in the middle of slides, bends, or other types of ties by prefixing the fret with `:duration:`.

vexflow.com

```
tabstave notation=true

notes :q (8/2.7b9b7/3) (5b6/2.5b6/3)v :8 7s12/4
notes t:16:9s:8:3s:16:0/4
```

Step 10: Add lyrics, annotations, or other text.

You can annotate your notation by enclosing comma-separated text between dollar symbols ($). VexTab will associate each comma-separated set of words with each of the preceding notes. Annotations can be used for lyrics, chords, picking hints, etc.

Annotations can be positioned above or below the stave using the $.top.$ or $.bottom.$ commands. Here's an example:

vexflow.com

```
tabstave notation=true time=4/4 key=Ab
tuning=eb
notes :q 5/5 5/4 5/3 ^3^ $Fi,Ga,Ro!$ :h 4/4 $.top.$ $Blah!$
```

There are two ways to customize these annotations. You can use preset styles or custom styles. For preset styles, simply prefix your annotation with `.style.` inside the dollar signs. E.g., The code `$.big.C Major$`, will render the text "C Major" in a big bold font.

The current preset styles are `big`, `medium`, and `italic`.

For custom styles you can specify the font face, size, and style by prefixing your annotation with `.face.size.style` within the dollar signs. E.g., `$.Times-14-italic.Blah$` will render "Blah" in 14pt Italic Times Roman. Here are some examples of preset and custom styles.

vexflow.com

```
tabstave notation=true key=A
notes :q (5/2.5/3.7/4) $.big.A7#9$ 5h6/3 7/4 |
notes :8 7/4 $.italic.sweep$ 6/3 5/2 3v/1 :q 7v/5 $.Arial-10-bold.P.H$
:8 3s5/5
```

Step Eleven: Staccatos, Fermatas, and more!

Add beautiful articulations to your scores using the `$.ariculation/position.$` syntax.

Articulations have different codes, e.g., `a.` for staccato, `a@a` for up-fermata, etc. They can be positioned on `top` or on the `bottom`.

Below is a list of all the available articulations.

Stacatto *Staccatissimo* *Accent* *Tenuto* *Marcato* *LH pizzicato Snap pizzicato*

a.   av   a>   a–   a^   a+   ao

Open Note *Up Fermata* *Down Fermata* *Bow Up* *Bow Down*

ah   a@a   a@u   a| am

vexflow.com

```
options space=20 font-style=italic
font-face=times font-size=10

tabstave notation=true tablature=false
notes :q
notes C/4 $.a./bottom.$
```

Step 12: Even more text!

VexTab provides a more flexible alternative for adding text to yor scores. You can create a `text` line with text that can be both vertically positioned, and note aligned. This method is great for notating chords.

In the example below I've created two text spans, "G Minor" and "A Major", each with a duration of a half-note. Notice that the duration syntax is the same (except that spans are separated by commas.) Also notice how I set the default font size in the `options` line.


vexflow.com

```
options space=12 font-size=14

tabstave notation=true time=4/4 key=Ab tuning=eb
notes :q 5/5 5/4 5/3 ^3^ :h 4/4
text :h,G Minor,A Major
```

You can vertically position any text by adding a .*NUMBER* to your list of spans. If unset, *NUMBER* defaults to `0`, which positions your text above the stave. Higher numbers move the text downward, and lower numbers (to the negative) move the text upward.
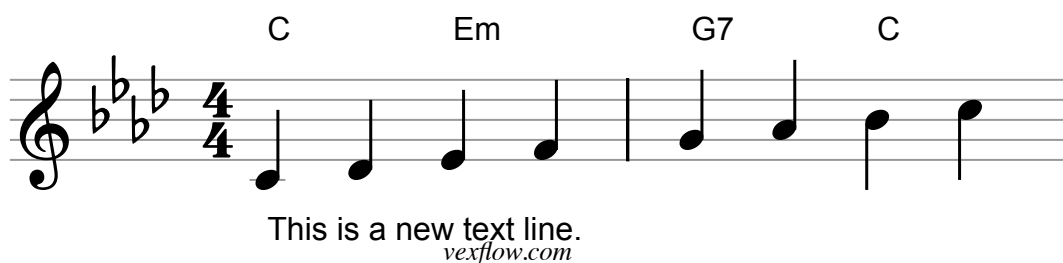
Here's an example:
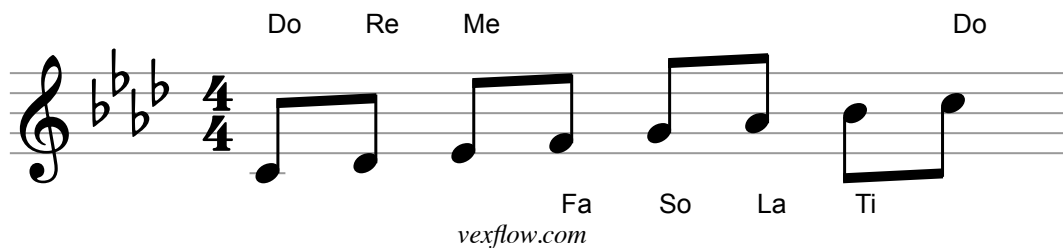

```
options space=10

tabstave notation=true tablature=false time=4/4 key=Ab tuning=eb
notes :8 C-D-E-F-G-A-B/4 C/5
text :8,.1,Do,Re,Me,.10,Fa,So,La,Ti,.1,Do

options space=10
```

If you have bars in your music, you will need to add a bar to your text so that the notes and text align correctly. Simply add a | between commas to indicate a bar. Bars can have text attached to them by suffixing the | symbol with your text, e.g., |bar.

Also, you can use the ++ command in your span to create a new text line on the same stave. Here's an example of these features:


```
options space=10 font-size=12

tabstave notation=true tablature=false
time=4/4 key=Ab tuning=eb
notes :q C-D-E-F/4 | G-A-B/4 C/5
text :h,.1,C,Em,|
text :h,G7,C
```

Text lines can also contain various musical symbols, such as codas and segnos. Simply prefix the symbol name with a # to render a symbol instead of text, e.g., `text #coda`.
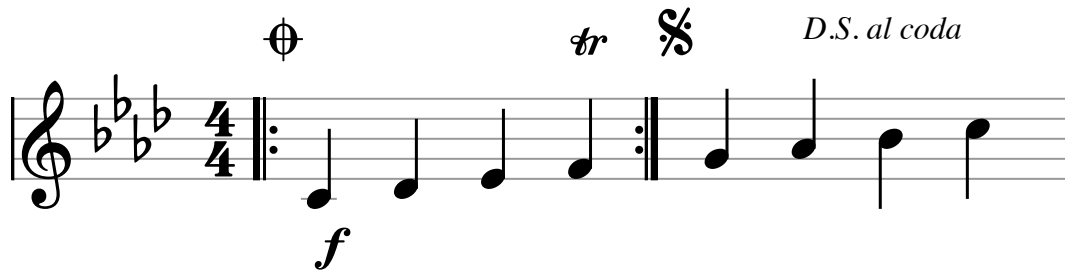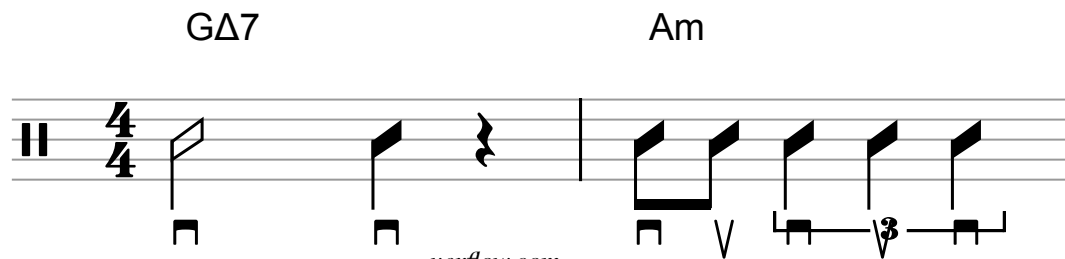
vexflow.com

```
tabstave notation=true tablature=false
time=4/4 key=Ab tuning=eb
notes :q =|: C-D-E-F/4 =:| G-A-B/4 C/5
text |#coda, :hd, , :q, #tr, |#segno
text ++, .11, |, #f
text ++, .0, .font=Times-12-italic, |, :w, ,|, :q, ,D.S. al
coda
```

Step 13: Notate rhythms with slash notation.

By suffixing a duration with s all following notes on the standard notation stave get rendered as slash notation. You can change the position of the note on the staff by changing the note name.

You can also use the u or d suffixes to designate upstrokes or downstrokes, respectively.

vexflow.com

```
options font size 14 space 15

tabstave notation=true tablature=false
time=4/4 clef=percussion
notes :2S Bd/4 :qS Bd/4 :q ## | :8S Bd/4 Bu/4 :qS Bd-Bu-Bd/4
^3^
text :w, GΔ7, |, Am
```

Step 14: Customize the appearance.

The rendered notation can be customized heavily using the `options` command. You can change the width, default fonts, scale, and spacing to prettify your notation to your hearts extent. For example, `options width=400` sets the width of the stave to 400 pixels.

Here's a list of the current options, followed by some examples.

option	description
width	Set the width of the stave in pixels.
scale	The zoom level of the notation. Default is 1.0.

option	description
space	Adds space (given in pixels) before the next stave.
stave-distance	Sets the distance (in pixels) between the note and tab staves.
font-face/font-style/font-size	Set the default font for annotations.
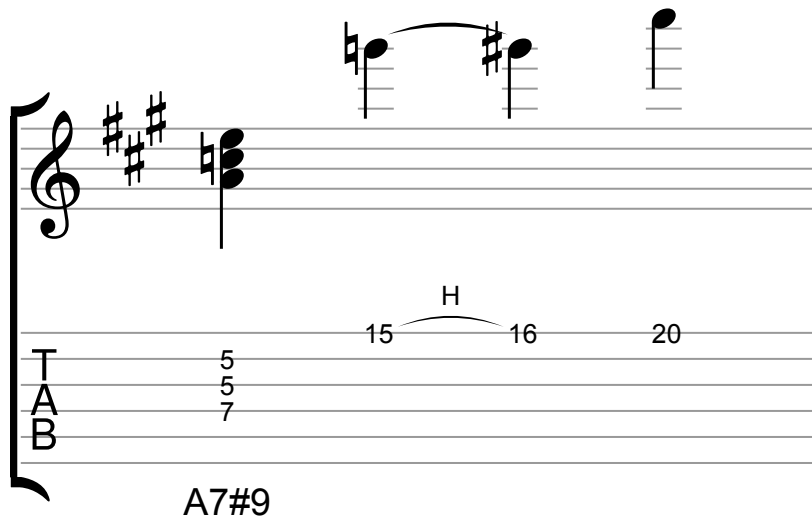
You can make it big.

vexflow.com

```
options width=100 scale=2.0
tabstave notation=true tablature=false
notes :q 5/5
```

You can add room above a stave for the high notes.

vexflow.com

```
options space=40 font-size=14
tabstave notation=true key=A
notes :q (5/2.5/3.7/4) $A7#9$ 15h16/1 20/1
```

Make more room between note and tab staves.

```
options stave-distance=50 scale=0.8
tabstave notation=true key=A
notes :q (5/2.5/3.7/4) ## 0h1/6 0/1
```

Step 15: Tablature can have stems too.

Use the `tab-stems` option to show stems in tablature. For single-voice tab, you can set the stem direction with the `tab-stem-direction` option.

Note that you might need to add some spacing using the `space` and `stave-distance` options to prevent overlap.

```
options space=12 font-size=14
tab-stems=true tab-stem-direction=up

tabstave time=4/4 key=A
notes :8 5/5 5/4 5/3 ^3^ :16 5-6-7-8/1 :8 9s10/1 :h s9v/1
```

Step N + 1: Coming soon

Meanwhile, you can stay in touch by following me at @11111110b.

The Blog