

Muse

Hack Club

muse.hackclub.com

Sound Visualization

Getting Started:

1. Open the muse editor:

muse.hackclub.com

The screenshot displays the Muse editor interface. On the left is a code editor with the following code:

```
1 // settings
2 const type = "triangle" // sine | triangle | square | sawtooth | piano | acoustic | edm | organ
3 const bpm = 120
4
5 // song goes here
6 // a, a#, b, c, c#, d
7
8 createMuse({ type:"si
9
10
11
12
13
14 // below maps keys so
15 const key = 4
16 const muse = createMu
17
18 const a = () => muse.
19 const s = () => muse.
20 const d = () => muse.
21 const f = () => muse.
22 const g = () => muse.
23 const h = () => muse.
24 const j = () => muse.
25 const k = () => muse.
26 const l = () => muse.
27
28 // these keys get bou
29 bindKeys({
30   a,
31   s,
32   d,
33   f,
34   g,
35   h,
```

The center of the interface features a grid of sample buttons. The 'Starter' button is highlighted with a red box, and a red arrow points from the 'examples' button at the bottom of the interface to it. Other buttons in the grid include 'Addams Family', 'Circus', 'Musical Keyboard', 'Anon Jam', 'America the Beautiful', 'Sample Keyboard', 'Tune Factory', 'Mortal Runs', 'Simple Beats', 'synco-chords', 'Simple Beats 2', 'Fill-up Glassy', 'Circus Bugged', 'Anon Jam', 'Mario Underground', 'Addams Run', 'Halfstep Keyboard', and 'Samples in the Klerb'.

On the right side, there is a 'Played Notes' section with a 'Clear Sounds' button and a list of notes: '4 c5 d5 e5 f5 d5 f5 d5 f5', 'e5 b4 b4 c5 d5 d5 d5', 'f5 d5 d5 d5 d5 d5 d5', 'f5 d5 d5 d5 d5 d5 d5', 'f5 d5 d5 d5 d5 d5 d5', 'f5 d5 d5 d5 d5 d5 d5', 'f5 d5 d5 d5 d5 d5 d5', 'f5 d5 d5 d5 d5 d5 d5', 'f5 d5 d5 d5 d5 d5 d5', 'f5 d5 d5 d5 d5 d5 d5', '4 b4 a4 a4 b4 c5 e5'. Below this list are buttons for 'Record new sample', 'confetti', 'flash3', 'pinwheel', 'piston3', 'prism3', 'splits', 'squiggle', and 'strike'.

At the bottom of the interface, there are buttons for 'play/attach', 'stop', 'share', 'examples', 'light/dark', and 'GitHub'. The 'examples' button is also highlighted with a red box.

Tips on composing your first song

Think small, then repeat

- Most music (and code) is made up of repetitive chunks that build into larger structures.
- Using this approach in Muse can make composing easier and reduce typing.

```
8 createMuse({ type:"sine", bpm }).play~
9
10 [c4; e4; g4; e4;]
11
12 ~
```

Start with a simple cell

- Example: `[c4; e4; g4; e4;]`
- Then, multiply it to create patterns.

Use the Starter template in Examples

- This template has key bindings for notes.
- Allows you to sketch ideas quickly before finalizing them.

Composing our first song

```
createMuse({ type:"sine", bpm }).play`
```

```
[ f5+ ;+ g5+ ;+ c5+ ; g5+ ;+ a5 ;+ c6 ;-- a#5 ;-- a5 ; f5+ ;+ g5+ ;+ c5+ ;+++ ] x 2
```

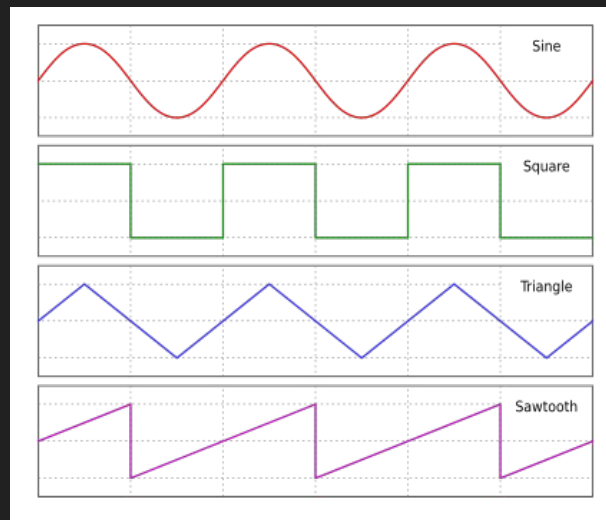
```
`
```

- You can lengthen notes by appending a +
- shorten them by appending a -
- To repeat something use x and some number
- Add a rest (pause) using ;

Composing our first song

```
1 // settings
2 const type = "triangle" // sine | triangle | square | sawtooth | piano | acoustic | edm | organ
3 const bpm = 120
```

- Bpm - beats per minute
- Type - type of synthesizer to use
- Offset notes up by half steps with a ^
- Offset notes down by half steps with a _







Composing our first song

- Layer sounds for depth
 - Once you have a musical idea, write additional `createMuse` functions to build on it.
 - You can add as many layers as you want to enhance complexity.
- Build chords in Muse
 - Stack individual notes to form chords, e.g., `[c4 e4 g4 b4;]`.



Using Samples



```
1 createMuse()
2 .play~
3 [bubbles ;+ bubbles ;+] x 10
4 ~
5 .play~
6 [ ; flash1 ;+ bubbles ;+ ] x 10
7 ~
8 .play~
9 [ ;; pinwheel ;; ] x 10
10 ~
11
```



 bubbles 



 clay 



 confetti 



 corona 



 dottedspiral 



 flash1 

 flash2 

 flash3 

 glimmer 

 moon 



 pinwheel 

 piston1 



 piston2 



 piston3 

 prism1 

 prism2 

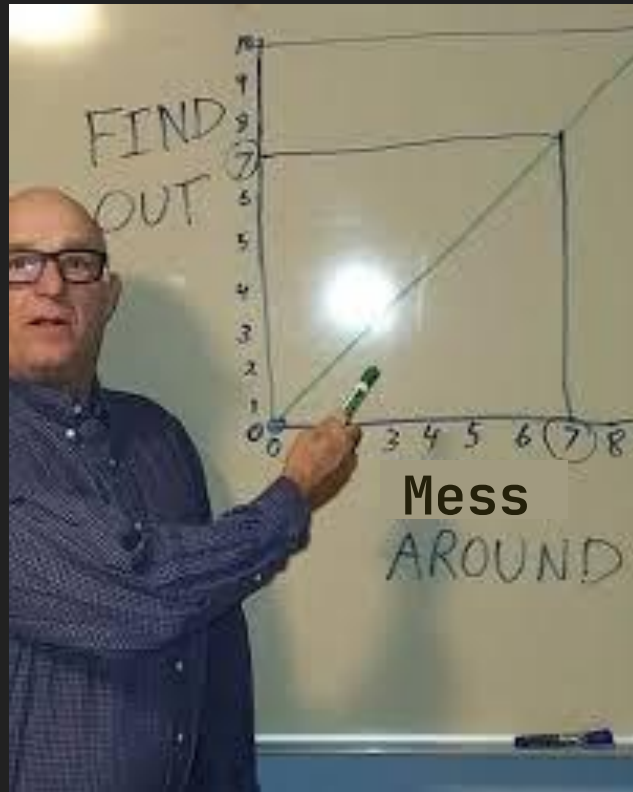
 prism3 

 splits 

 squiggle 

 strike 

Hacking Time



Try Running This!

```
[ f5++ ;- ; g5++ ;- ; c5++ ; g5++ ;- ; a5++ ;- ; c6 ;-- a#5 ;-- a5 ;- f5++ ;- ; g5++ ;- ; c5++ ; ;++ ] x  
2
```

```
d5+ ;- e5+ ;- f5+ ;- f5+ ;- g5+ ;- e5 ;-- d5 ;-- c5++ ; ;++ d5+ ;- d5+ ;- e5+ ;- f5+ ;- d5++ ; c5+ ;-  
c6++ ; c6+ ;- g5++ ; ;+
```

```
d5+ ;- d5+ ;- e5+ ;- f5+ ;- d5+ ;- f5+ ;- g5++ ; e5+ ;- d5+ ;- c5++ ; ;+ d5+ ;- d5+ ;- e5+ ;- f5+ ;- d5+  
;- c5++ ; g5+ ;- g5+ ;- g5+ ;- a5+ ;- g5++ ;+
```

```
f5+++ ; ;+ g5+ ;- a5+ ;- f5+ ;- g5+ ;- g5+ ;- g5+ ;- a5+ ;- g5++ ; c5++ ; ;+ d5+ ;- e5+ ;- f5+ ;- d5+ ;  
g5+ ;- a5+ ;- g5+ ;
```

```
c5 ;-- d5 ;-- f5 ;-- d5 ;-- a5+ ;-- ;- a5+ ;-- ;- g5++ ;- ; c5 ;-- d5 ;-- f5 ;-- d5 ;-- g5+ ;-- ;- g5+  
;-- ;- f5+ ;-- ;- e5 ;-- d5+ ;-
```

```
c5 ;-- d5 ;-- f5 ;-- d5 ;-- f5++ ; g5+ ;- e5+ ;-- ;- d5 ;-- c5++ ; c5+ ;- g5++ ; f5++ ;++
```

```
c5 ;-- d5 ;-- f5 ;-- d5 ;-- a5+ ;-- ;- a5+ ;-- ;- g5++ ;- ; c5 ;-- d5 ;-- f5 ;-- d5 ;-- c6+ ;-- ;- e5+  
;-- ;- f5+ ;-- ;- e5 ;-- d5+ ;-
```

```
c5 ;-- d5 ;-- f5 ;-- d5 ;-- f5++ ; g5+ ;- e5+ ;-- ;- d5 ;-- c5++ ; c5+ ;- g5++ ; f5++ ;++
```