

FoxDot >> Coding music

Part 1: Introduction for beginners

Tutor: Jens Meisner
Computerkuenstler und Mediengestalter

“Live Coding is a new direction in electronic music and video, and is getting somewhere interesting...



Live Coders exposes and rewires the innards of software while it generates improvised music.”
- toplap.org -

>> FoxDot

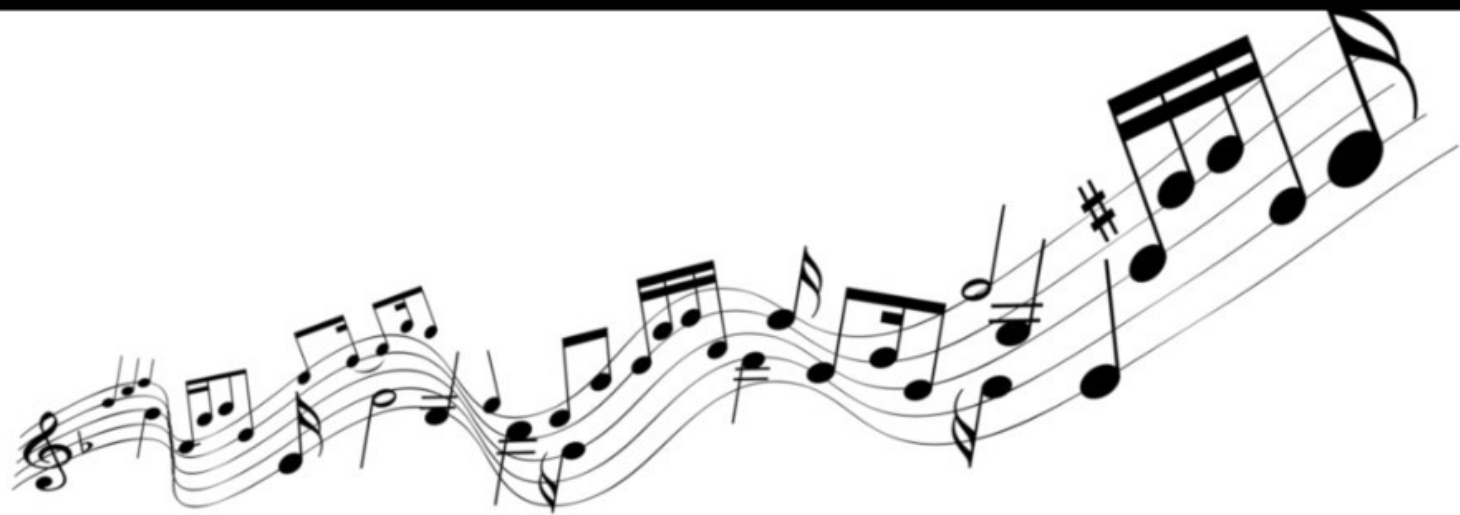
Live coding with Python and SuperCollider

An Introduction >>

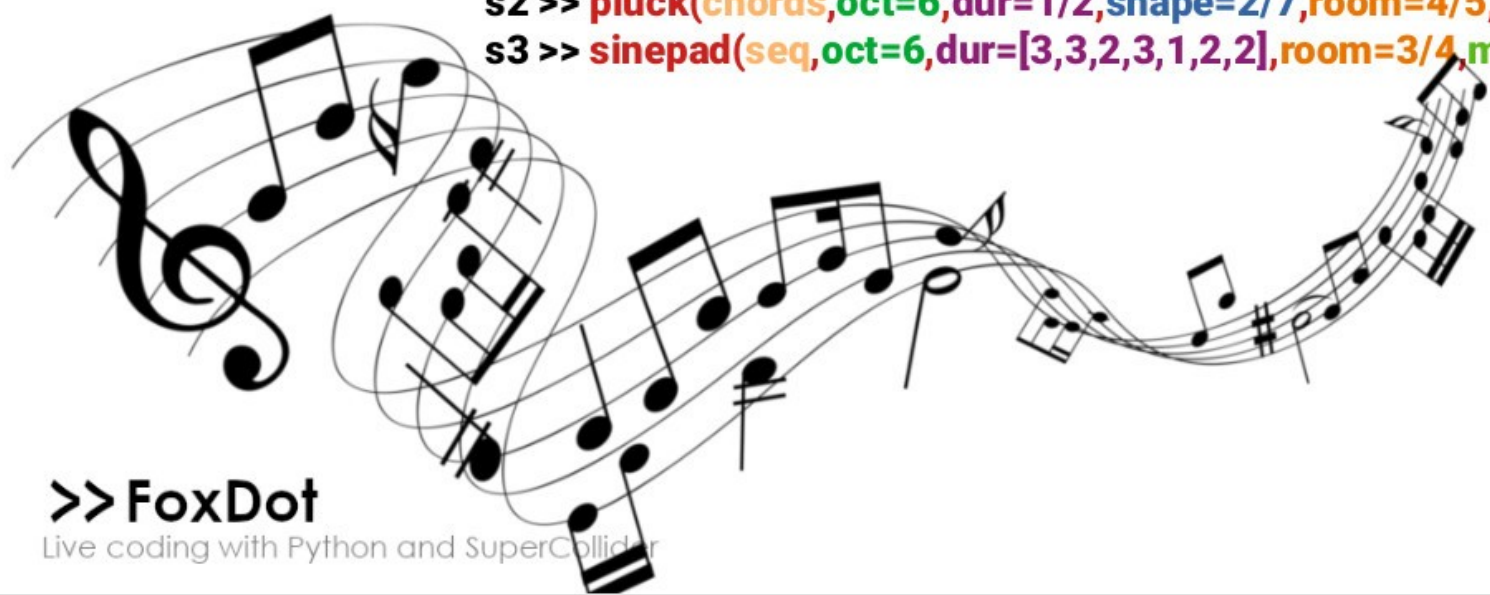
Why using code for music ?

>> **FoxDot**

Live coding with Python and SuperCollider



```
s1 >> klank(bassline,oct=4,dur=[6,4,2,2,2],shape=2/5,amplify=3/5,amp=var([0,1],8))  
s2 >> pluck(chords,oct=6,dur=1/2,shape=2/7,room=4/5,mix=1/2,amplify=1/2,amp=1)  
s3 >> sinepad(seq,oct=6,dur=[3,3,2,3,1,2,2],room=3/4,mix=1/2,amplify=4/5,amp=1)
```



>> **FoxDot**

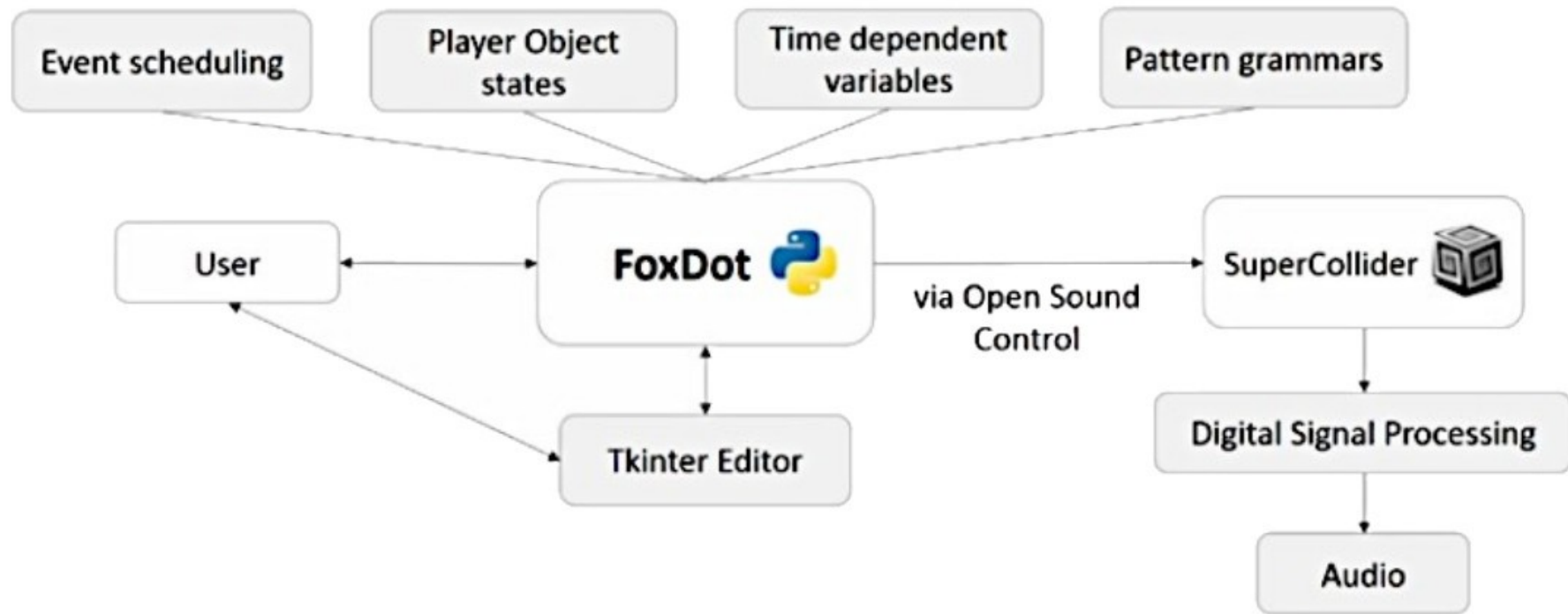
Live coding with Python and SuperCollider

An Introduction >>

What is FoxDot?

>> FoxDot

Live coding with Python and SuperCollider



>> **FoxDot**

Live coding with Python and SuperCollider

```
T Troop - Ryan@localhost:57890
File Edit Code Constraints Help
1 # Whos here to dance?
2
3 d1 >> play("x", dur=PDur([2,2,3,5,7],8), sus=1, pan=PStep(7, P+(-1,1))) + 2
4
5 b1 >> bass(dur=16, sus=4, room=1, mix=1, shape=0.5, lpf=linvar([1000,8000],12)) + (0, PRand
  ([6,7,9]))
6
7 l1 >> play("shroom", rate=PRand([5,10]), shape=.1).every(6, "stutter", 4, pan=[-1,1])
8
9 l2 >> sitar(PWalk(), oct=[2,3], dur=[4,2,2], sus=[1,2,3,4,5], slide=0.2) + [1,3,5]
10 Ryan
11 z1 >> soprano(12.degree + (2,[4,6,7]), dur=[8,4,2,1], oct=4, sus=2, blur=4, amp=)
12 loz
13 Scale.default=("mixolydian")
14
loz : z1 >> soprano(12.degree + (2,[4,6,7]), dur=[8,4,2,1], oct=7, sus=2
, blur=4)
loz : z1 >> soprano(12.degree + (2,[4,6,7]), dur=[8,4,2,1], oct=5, sus=2
, blur=4)
Ryan : d1 >> play("x", dur=PDur([2,2,3,5,7],8), sus=1, pan=PStep(7, P+(-
1,1))) + 2
loz : z1 >> soprano(12.degree + (2,[4,6,7]), dur=[8,4,2,1], oct=4, sus=2
, blur=4)
```

>> FoxDot

Live coding with Python and SuperCollider

Getting Started With FoxDot >>

If you want to know more about a function or a class,
type in help + the Python object in brackets

help(object)

In example, if you want know more about the “pluck” instrument, type in:

help(pluck)

Getting Started With FoxDot >>

Player Objects

“pluck” is an Python object, that represents a synthesizer definition
in SuperCollider called SynthDef.

To see all available SynthDefs, type in:

```
print(SynthDefs)
```

>> Before we start, some basics >>

- To execute a single line, press ALT + ENTER
- To execute an entire block of lines, press CTRL + ENTER
- If you want to stop all sound playing, press CTRL + .
- If you want to stop a single line, add `.stop()` at the end, and press ALT + ENTER:

To create a working instrument, you need name it first. You only can use one letter with a following number: Try it and all above with an instrument of the list

```
s1 >> pluck()
```

Getting Started With FoxDot >>

Patterns

Getting Started With FoxDot >>

TimeVars

>> **FoxDot**

Live coding with Python and SuperCollider

Getting Started With FoxDot >>

“play” and “loop” Players

>> **FoxDot**

Live coding with Python and SuperCollider

- Beside synthesizer instruments in SuperCollider called SynthDefs, there are 2 other Players to be used in FoxDot
- “play” and “loop” are actually audio sample player
- “play” uses letters and characters in brackets to call samples, that are part of the FoxDot basic installation
- e.g.:

b1 >> play("x-o-")

Use {} [] () <> like so: **b1 >> play("x-o[-{ox}]")**

{} - Random, *[]* - All-in-one step, *()* - In turn, *<>* - Simultaneously

- There are many variables, that can be add to the “play” Player by adding it behind the quotes separated by a comma, like so:

```
b1 >> play("<x-o-><..+.[+]>", pan=(-1, 1))
```

- One of this variables is “sample”.
- Each character relates to a folder of files arranged in alphabetical order. To select a different file, use the sample keyword:

```
b1 >> play("x-o-", sample=1)
```

- The following list shows a loose categorization of currently available samples

		Symbol/Sample	0	1	2	3	4	5	6	7	M							Symbol/Sample	0	1	2	3	4	5	6	7
Bass	Orange	a									n							1								
Snare	Yellow	A	Orange	Orange	Orange						N	Blue	Blue	Blue	Blue	Blue		2	Purple	Purple						
Hi Hat	Light Green	b	Green								o	Yellow	Yellow	Yellow	Yellow			3	Purple	Purple						
Open Hat	Light Blue	B	Grey	Grey	Green	Green					O	Yellow	Yellow	Yellow	Yellow	Yellow		4	Teal	Teal	Orange					
Crash Hat	Light Blue	c	Orange	Orange	Orange	Orange	Orange	Orange			p	Orange	Orange	Orange	Orange	Orange		&	Teal	Teal	Orange					
Cymbal	Purple	C	Purple				Purple				P	Orange	Orange	Orange	Orange	Orange		*	Light Green	Light Green	Light Green	Light Green	Light Green			
Low Tom	Brown	d	Yellow	Yellow	Yellow	Yellow					q	Grey	Grey	Grey	Grey		@	Grey	Grey	Grey	Grey	Grey	Grey			
Mid Tom	Light Green	D	Yellow	Yellow	Yellow	Yellow					Q	Grey	Grey	Grey	Grey		\	Grey	Grey	Grey	Grey	Grey	Grey			
Hi Tom	Light Green	e	Orange	Orange	Orange	Orange	Orange	Orange			r	Orange	Orange	Orange	Orange	Orange		bar	Orange	Orange	Orange	Orange	Orange	Orange		
Triangle	Teal	E	Orange	Orange	Orange	Orange	Orange	Orange			R	Orange	Orange	Orange	Orange	Orange		^	Orange	Orange	Orange	Orange	Orange	Orange		
Clap	Light Green	f	Orange	Orange	Orange	Orange	Orange	Orange			s	Blue	Blue	Blue	Blue		:	Light Green	Light Green	Light Green	Light Green	Light Green	Light Green			
Snap	Purple	F	Grey	Grey	Grey	Grey	Grey	Grey			S	Light Green	Light Green	Light Green	Light Green		\$	Brown	Light Green	Yellow	Blue	Orange	Purple	Yellow	Blue	
Bell/Metal	Light Green	g	Orange	Orange	Orange	Orange	Orange	Orange			t	Orange	Orange	Orange	Orange	Orange		=	Light Blue	Light Blue	Light Blue	Light Blue	Light Blue	Light Blue		
Noise	Grey	G	Orange	Purple							T	Orange	Orange	Orange	Orange		!	Purple	Purple	Purple	Purple	Purple	Purple	Purple		
Scratch	Yellow	h	Purple	Purple	Light Green	Light Green	Light Green				v	Orange	Orange	Orange	Orange	Orange		/	Purple	Purple	Purple	Purple	Purple	Purple		
8 Bit	Light Green	H	Light Green	Light Green	Light Green	Light Green	Light Green				V	Orange	Orange	Orange	Orange	Orange		#	Purple	Purple	Purple	Purple	Purple	Purple		
Voice	Purple	T	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow			w	Orange	Orange	Orange	Orange	Orange		-	Orange	Orange	Orange	Orange	Orange	Orange		
Beep	Green	I	Light Green	Light Green	Light Green	Light Green	Light Green	Light Green			W	Orange	Orange	Orange	Orange	Orange		<	Orange	Orange	Orange	Orange	Orange	Orange		
Various	Grey	j	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow			x	Orange	Orange	Orange	Orange	Orange		%	Grey	Grey	Grey	Grey	Grey	Grey		
Percussion	Orange	J	Grey	Grey	Grey	Grey	Grey	Grey			X	Orange	Orange	Orange	Orange	Orange		+	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow		
Hom	Light Blue	k	Orange	Orange	Orange	Orange	Orange	Orange			y	Orange	Orange	Orange	Orange	Orange		?	Purple	Purple	Purple	Purple	Purple	Purple		
Shaker	Blue	K	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow			Y	Grey	Grey	Grey	Grey	Grey		:	Grey	Grey	Grey	Grey	Grey	Grey		
Tambrine	Light Green	l	Yellow	Light Green	Light Green	Light Green	Light Green	Light Green	Light Green	Light Green	z	Yellow	Yellow	Yellow	Yellow		~	Grey	Grey	Grey	Grey	Grey	Grey	Grey		
Clave (Woodstick)	Orange	L	Orange	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Z	Grey	Grey	Grey	Grey	Grey										

- The “loop” Player can be used to implement your own samples, like a singing voice, an analog instrument, or you have your own drum samples
- The variable dur is used to set a duration of the loop

```
l1 >> loop("path/to/my/file.wav", dur=32)
```

- You can put files in a special folder located in FoxDot/snd/_loop_ which can be opened by going to “Help & Settings” and then “Open Samples Folder” from the FoxDot menu.
- You don’t need to supply the full path (or extension) for files in this folder:

```
l1 >> loop("my_file", dur=4)
```

First, A Little Bit Music Theory >>

Song Structure

>> **FoxDot**

Live coding with Python and SuperCollider

- Common Structures for a song is as followed:

Intro (4 Bars) - **Verse** (8 -16 Bars) + **Prechorus** (Optional) - **Chorus**(8 - 16 Bars) -
2nd Verse (8 – 16 Bars) + **Prechorus** (Optional) - **Chorus** (8-16 Bars)

- The following is an example of a song structure in common electronic music:

Intro	Break	Buildup	Drop	Break	Buildup	Drop	Outro
16 Bars	16 Bars	4/8/16 Bars	16 Bars	16 Bars	4/8/16 Bars	16 Bars	16 Bars

First, A Little Bit Music Theory >>

Chords and Notes

Chords

Melodies – From Chords To Melody

Melodies – From Melody To Chords

Melodies – Add a counter melody (Arpeggio)

Melodies – From Chord Progression to Bass lines

Melodies – From Bass lines to Chord Progression

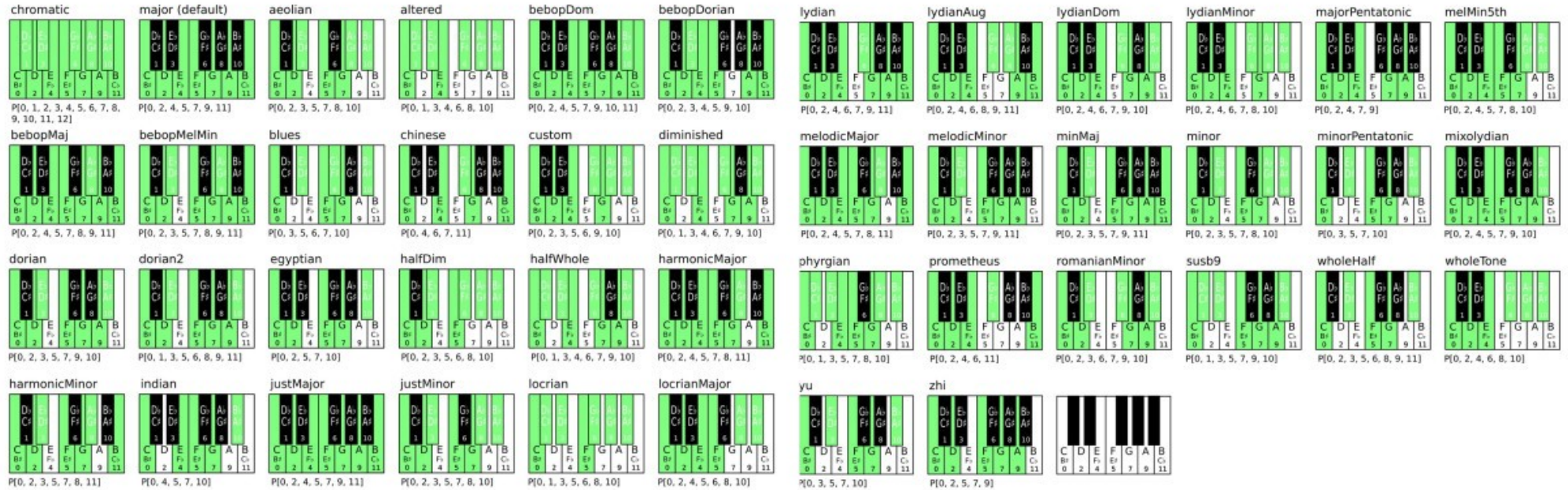
First, A Little Bit Music Theory >>

Scales and Modes

Using Scales

Using Modes

- The schematics shows all modal Scales available by name in FoxDot



>> FoxDot

Live coding with Python and SuperCollider

First, A Little Bit Music Theory >>

BeatBox

>> **FoxDot**

Live coding with Python and SuperCollider

Live Jam >> Let's make music together



>> FoxDot

Live coding with Python and SuperCollider

Live Jam >>

Connect and get Troop started

>> **FoxDot**

Live coding with Python and SuperCollider

Live Jam >>

Connect to the Wifi Network

SSID: **Wifi4DS**
Password: **25M4Ei7[89**

>> **FoxDot**

Live coding with Python and SuperCollider

- Open a terminal, execute “python /Path/To/Script/run-client.py”, fill in the form and press “Ok”

```
File Edit View Terminal Tabs Help
bbscar@ishapenoise:~$ python '/home/bbscar/Desktop/FOXDOT/Troop-0T/run-client.py'
bash: /home/bbscar/Desktop/FOXDOT/Troop-0T/run-client.py: Permission denied
bbscar@ishapenoise:~$ python '/home/bbscar/Desktop/FOXDOT/Troop-0T/run-client.py'
bbscar@ishapenoise:~$ python '/home/bbscar/Desktop/FOXDOT/Troop-0T/run-client.py'
bbscar@ishapenoise:~$ python '/home/bbscar/Desktop/FOXDOT/Troop-0T/run-client.py'
```

Troop v0.9.5	
Host:	10.3.141.170
Port:	57890
Name:	bbscar
Password:	
Language:	FoxDot
Ok	

```
Troop - yenz@10.3.141.170:57890
File Edit Code Help
1 This user is the host. It is the fastest and loudest of em. This computer
  needs to execute run-server.py first. The terminal will show the IP it is
  running on. With the FoxDotSpot Dongle plugged in, it should be some ip
  with 10.3.141.____ (in this case 170).
2
3
4 After this execute run-client.py, and enter the same IP, username and
  password into the popup window, and start it...
5
6 bbscar
7
8 This user is a client. It only need to execute run-client.py, while using
  the same IP Address as the host above. Everyone will see the same in
  their editor...Have fun!
9
10 yenz
11
12
13
Peer 'bbscar' has joined the session
Warning: Could not fetch info from SCLang server. Using defaults...
```

>> FoxDot

Live coding with Python and SuperCollider