Tinte - User Guide

Typesetting Sheet Music

Sven Eric Panitz
Hochschule Rhein-Main

Contents

1 Introduction

Welcome to Tinte, a tool for typesetting music.

1.1 What Tinte is not

In order that you can stop reading early, some things you cannot expect from Tinte:

- Tinte is not wysiwyg. It has no graphical user interface and no means of previewing the music.
- Tinte cannot play the music written in it. There is now way to hear the music written with Tinte.
- Tinte is not a composing tool. You cannot transpose music, search in music, generate music or analyse music.

1.2 What Tinte is

It is a compiler, which reads some source code and generates postscript.

1.3 Related and unrelated systems

The closest thing to Tinte is lilypond [lil]. It uses the same approach. A textual description of music is read and transformed to a printable format. Some ideas of the lilypond syntax where adapted to Tinte. The main difference in notation is. in lilypond every note pitch is written with its accidentials. E.g., when in key a major, you have alwas write `cis` to get a c sharp note and you will write `c` to get a `c` note. In Tinte you will write `c` to get the c sharp and will write `na c` to get the natural c.

MuTeX, musecore, MusicXML,Haskore [HMGW96].

1.4 The name of the game

Tinte is a German word, which means ink. It is used as self referential acronym for: *tinte is not TEX*.

1.5 History of Tinte

Work on Tinte started in the early 90s, when I found an original print of a Monzino sonata in our local library. The thought of making it available to a wider audience on the internet was appealing. In those days, sheet music was not found on the internet.

So I decided to typeset the sonata. Unfortunatly, there were no good freely available programs for typesetting music in a high quality. I ended up with MuTeX a library extension to the typesetting system TeX for typesetting music. I succeeded in producing a nice print of the Monzino sonata. However, typesetting music with MuTeX was very tedious and many special means needed for guitar notation were not readily at hand.

Therefore, I decided to write my own typesetting program. The resulting program was called Tinte and had been implemented with the programming language Clean. [Pan97] Typesetting music with Tinte was a bit more convenient. A Nava duo, a Porro sonata and Schubert Lied have been created with Tinte in those days. However, implementing a typesetting system from scratch completely alone is a time consuming task. Some day I did not have any more time to invest into the Tinte project and stopped development.

Eventually other programmers had the time to implement a typesetting system. Today there is lilypond a powerful typesetting system for sheet music. In its structure it is quite similar to the ways I had taken with Tinte. It is a command line tool, which translates a textual description of the music into a printable format.

However, Tinte has not been completely sunsetted. I converted the Clean sources to the programming language Haskell, and I am still experimenting with it. The latest pieces typeset with Tinte are the Duetto by Luigi Moretti, Variations by v. Call and a Duo by Roeder.

Tinte now no longer uses the fonts from MuTeX and produces dvi as target code, but uses lilypon's feta/emmentaler fonts and now directly produces postscript output. This makes creation of slurs and beams much more convenient.

1.6 Limitations

Non Ascii characters for title or in composers name are not supported. Sorry, no music by Händel.

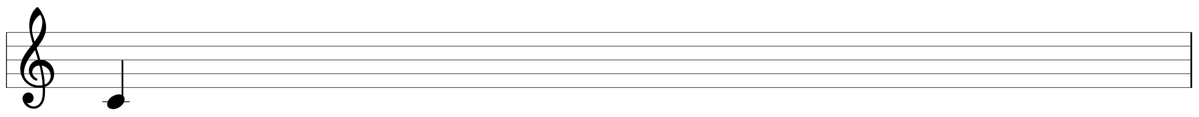
2 Simple musical element

2.1 SingleNotes

We start with »Hello World« in music.

```
singleNote=[[ n4c1 ]]
```

Listing 1: singleNote1.tinte



As you see, this produces a single crotchet (quarter note) of C^1 aka C4.

The n tells tinte to produce a singe note. The first number is the note value. The last two characters signify the pitch.

If you write the code above in a file called `singleNote1.tinte` and start tinte on command line by:

```
tintec singleNote1
```

It will produce a postscript file called `out.ps` with the contant above. If you have the simple programm `ps2pdf` installed then you can get a pdf-file by the following command:

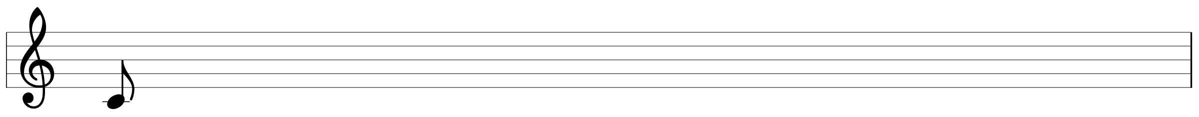
`ps2pdf out.ps singleNote.pdf`.

2.1.1 Note Values

Therefor a quaver is written as:

```
singleNote=[[ n8c1 ]]
```

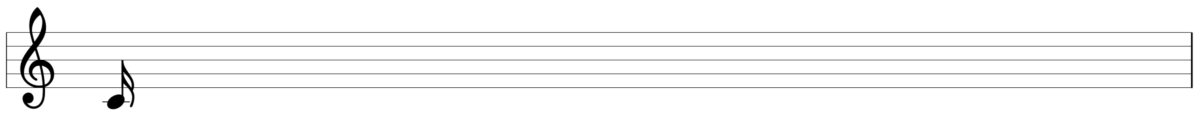
Listing 2: singleNote2.tinte



A semiquaver is:

```
singleNote=[[ n16c1 ]]
```

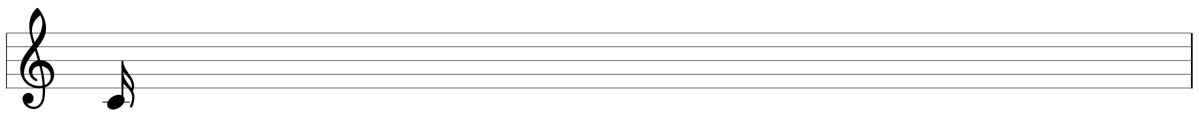
Listing 3: singleNote3.tinte



Here ist goes, a demisemiquaver:

```
singleNote=[[ n16c1 ]]
```

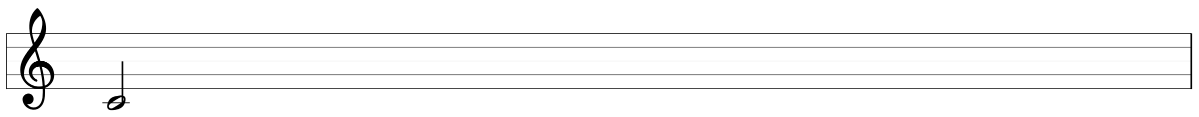
Listing 4: singleNote4.tinte



A minim is written as:

```
singleNote=[[ n2c1 ]]
```

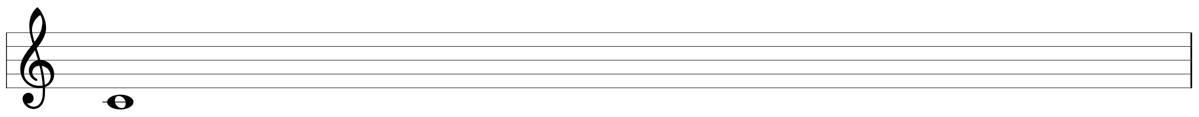
Listing 5: singleNote5.tinte



And eventually a semibreve is written as:

```
singleNote=[[ n1c1 ]]
```

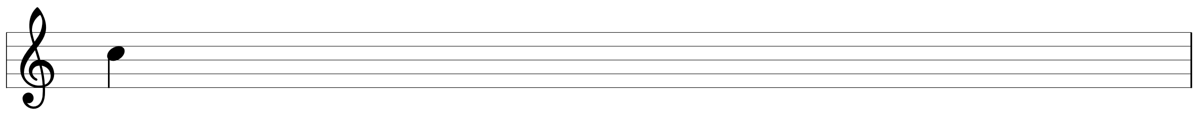
Listing 6: singleNote6.tinte



2.1.2 Note Pitches

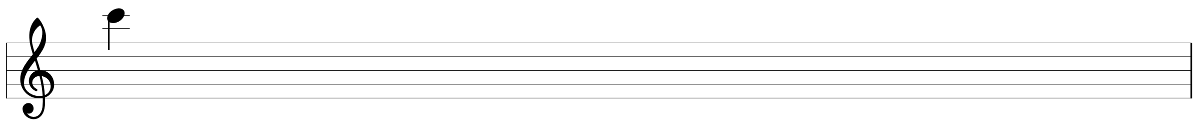
```
singleNote=[[ n4c2 ]]
```

Listing 7: singleNote7.tinte



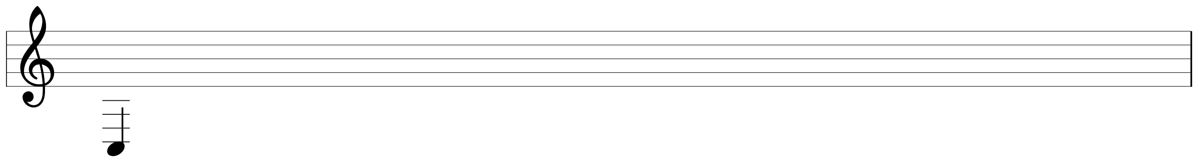
```
singleNote=[[ n4c3 ]]
```

Listing 8: singleNote8.tinte



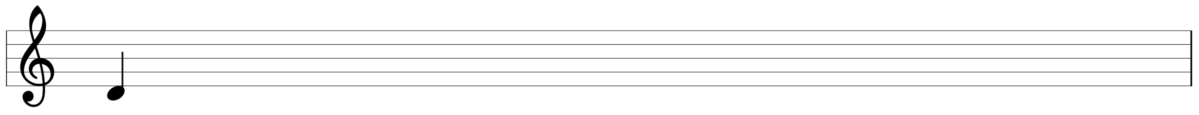
```
singleNote=[[ n4c0 ]]
```

Listing 9: singleNote9.tinte



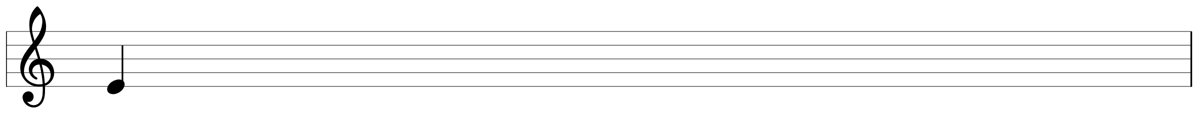
```
singleNote=[[ n4d1 ]]
```

Listing 10: singleNote10.tinte



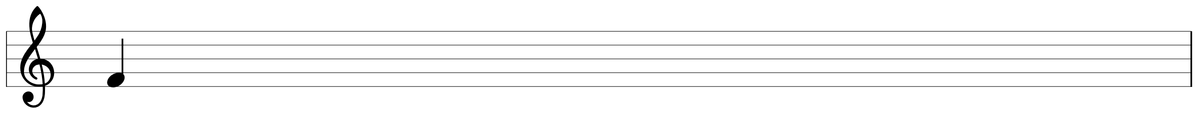
```
singleNote=[[ n4e1 ]]
```

Listing 11: singleNote11.tinte



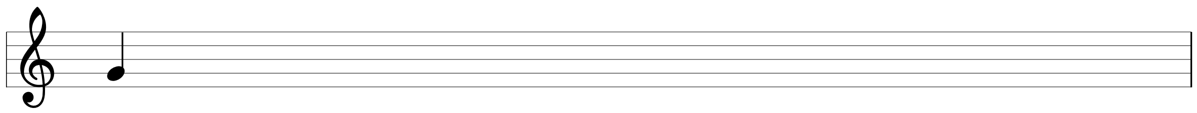
```
singleNote=[[ n4f1 ]]
```

Listing 12: singleNote12.tinte



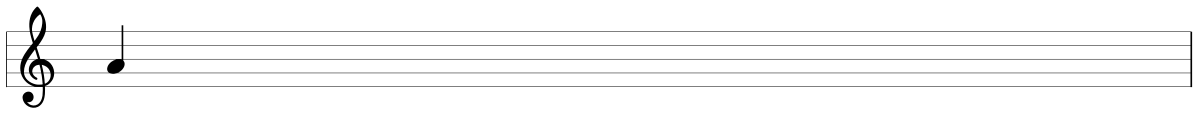
```
singleNote=[[ n4g1 ]]
```

Listing 13: singleNote13.tinte



```
singleNote=[[ n4a1 ]]
```

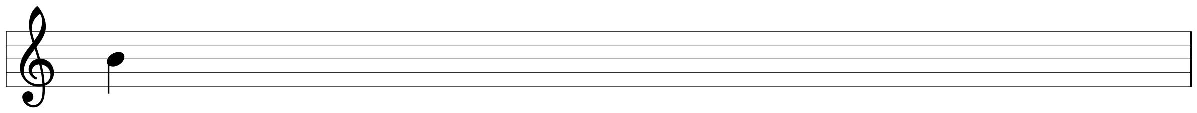
Listing 14: singleNote14.tinte



And now something different. The b is not notated as `b` but as `h` in tinte. This is the German name of a note b. The symbol `b` is reserved as accidental for flats

```
singleNote=[[ n4h1 ]]
```

Listing 15: singleNote15.tinte

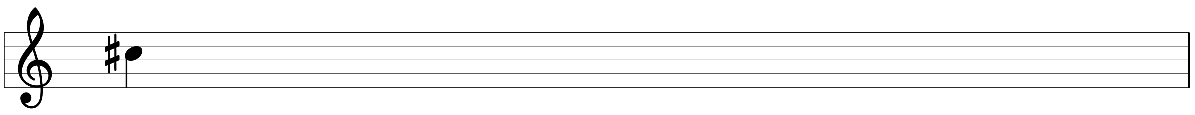


2.1.3 Accidentals: Sharp and Flat

You can write accidentals in front of a note. e.g. a sharp symbol.

```
singleNote=[[ #n4c2 ]]
```

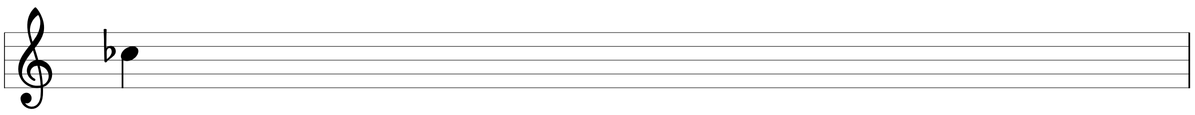
Listing 16: accidental1.tinte



Or the letter `b` for flats.

```
singleNote=[[ bn4c2 ]]
```

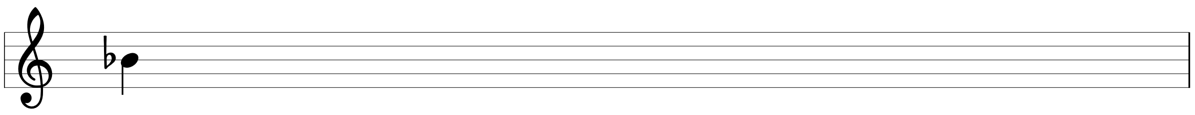
Listing 17: accidental2.tinte



Now you se, why we decided to use the `h` for the note `b`. The note b flat is written as:

```
singleNote=[[ bn4h1 ]]
```

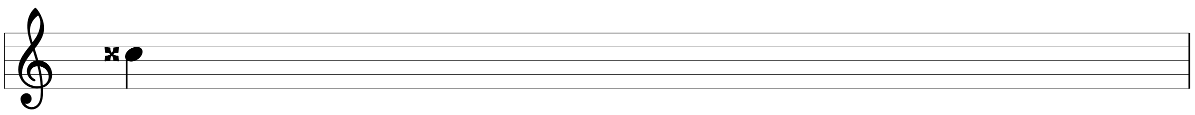
Listing 18: accidental3.tinte



Double accidentals are also supported:

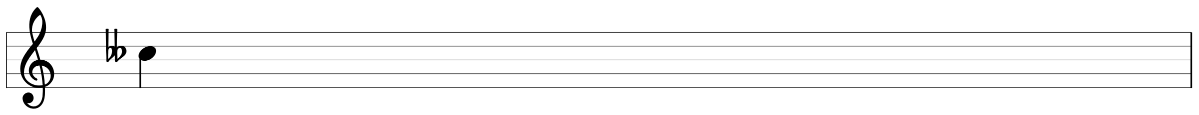
```
singleNote=[[ ##n4c2 ]]
```

Listing 19: accidental4.tinte



```
singleNote=[[ bbn4c2 ]]
```

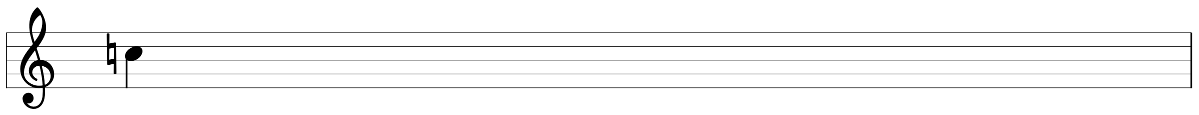
Listing 20: accidental5.tinte



Naturals are denoted by `na`:

```
singleNote=[[ nan4c2 ]]
```

Listing 21: accidental6.tinte

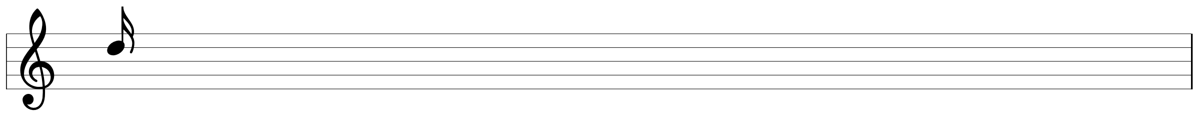


2.2 Stems

The direction of the stem is set by its usual defaults. However you can control the direction by adding one of the symbols `^` or `_` between note value and pitch:

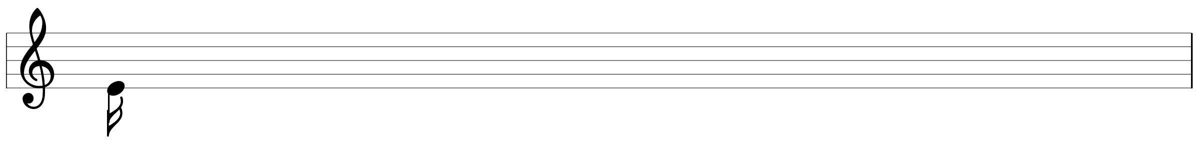
```
singleNote=[[ n16^d2 ]]
```

Listing 22: stem1.tinte



```
singleNote=[[ n16_e1 ]]
```

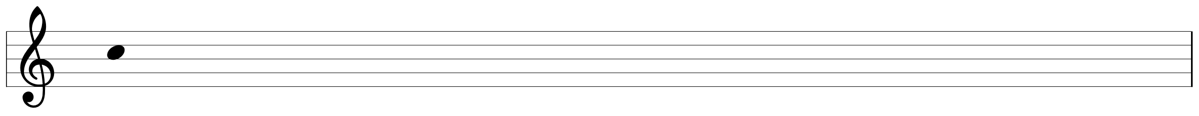
Listing 23: stem2.tinte



If you need no stem at all (this will be the case in chords and groups with beams) you replace the preceeding `n` with a `k`. (*Kopf* is the German word for head.)

```
singleNote=[[ k16c2 ]]
```

Listing 24: stem3.tinte

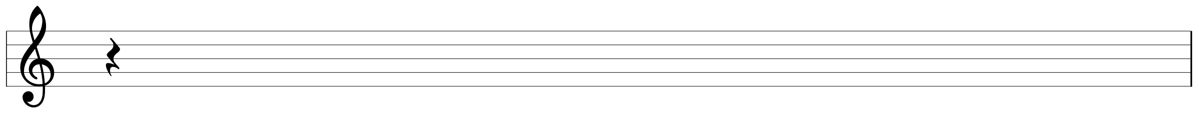


2.3 Rests

Rests are musical elemnts on its own. They are denoted by the letter `p` (rest means *Pause* in German) followed by a number for the duration of the rest. So `p4` stnds for a quarter rest.

```
rest1=[[ p4 ]]
```

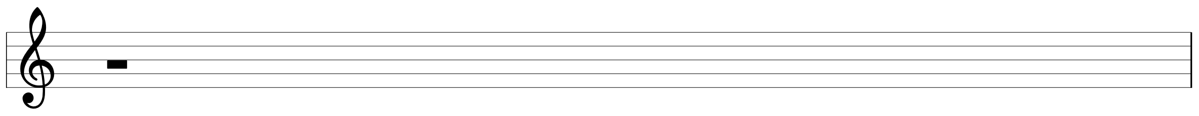
Listing 25: rest1.tinte



A whole rest ist denoted by `p1`.

```
rest2=[[ p1 ]]
```

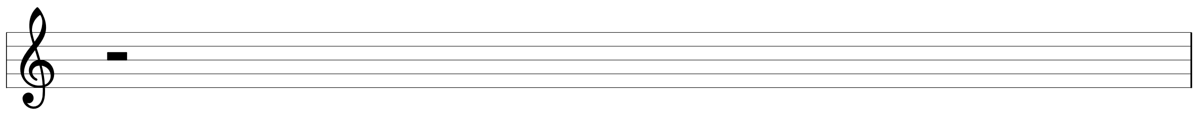
Listing 26: rest2.tinte



Accordingly go the other durations.

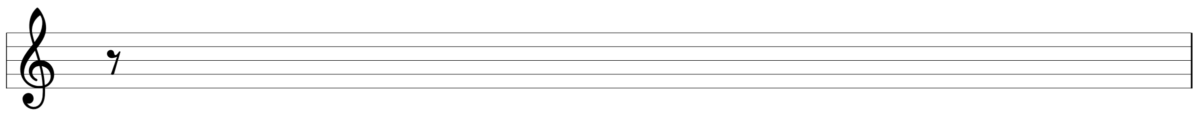
```
rest3=[[ p2 ]]
```

Listing 27: rest3.tinte



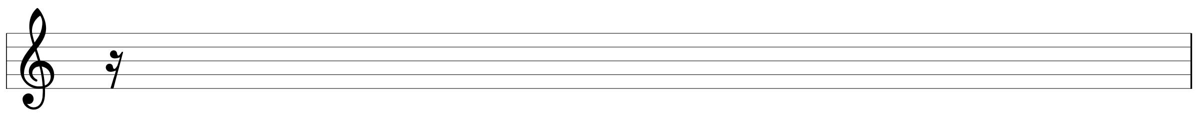
```
rest4 =[[ p8 ]]
```

Listing 28: rest4.tinte



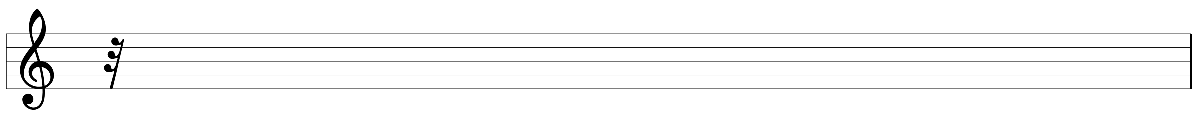
```
rest5 =[[ p16 ]]
```

Listing 29: rest5.tinte



```
rest6 =[[ p32 ]]
```

Listing 30: rest6.tinte

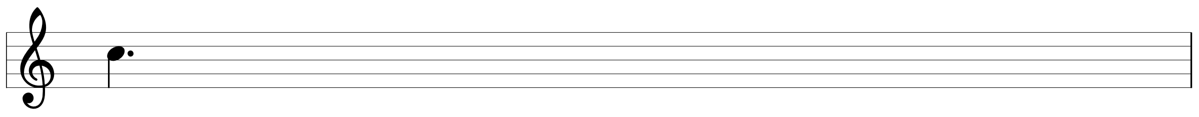


2.4 Dotted Notes

Dotted notes are simply created by a ot following the element.

```
dotted1 =[[ n4c2 . ]]
```

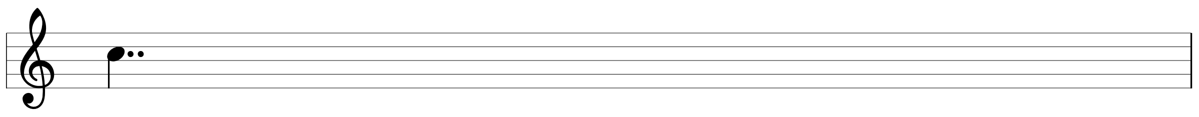
Listing 31: dotted1.tinte



You can create double dotted notes by two dots following the note element.

```
dotted2 =[[ n4c2 .. ]]
```

Listing 32: dotted2.tinte



3 Combining Musical Elements: Melodies and Polyphony

Musical elements, as notes rests etc. can be combined to new musical elements in two ways:

- Horizontically, which signifies that the elements are played one after the other. This enables a composer to write melodies.
- Vertically, which signifies, that the elements are two be played in parallel, i.e. at the same time. This will lead to polyphnic music.

Accordingly Tinte provides two combinations for arbitrary musical elements. It has the operator symbol `--` for horizontal combination of musical elements and the operator symbol `|/` for vertical combination.

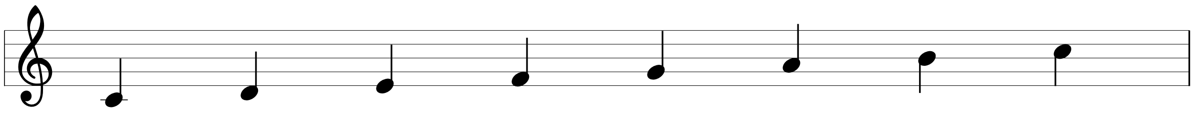
3.1 Sequences

3.1.1 Melodies

The sequential operator for musical elements is the operator `--`. Therefor a simple scale can be written as:

```
scale1 =[[ n4c1—n4d1—n4e1—n4f1—n4g1—n4a1—n4h1—n4c2 ]]
```

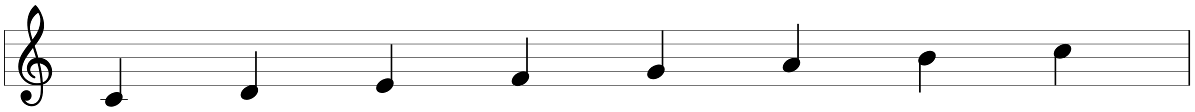
Listing 33: scale1.tinte



The sequential operator uses the last node value as default as well the last octave of the pitch. Therefore you leave them off and the scale notates shorter as:

```
scale2 =[[  n4c1—d—e—f—g—a—h—c2  ]]
```

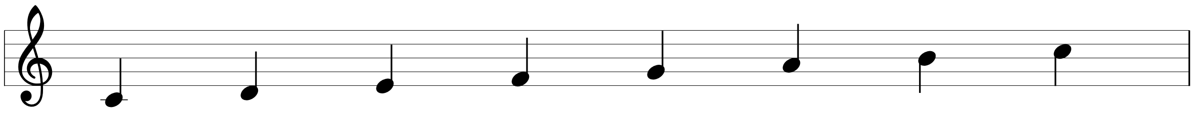
Listing 34: scale2.tinte



You can even leave out the operator alltogether. It is the default operator between musical elements:

```
scale2 =[[  n4c1defg  ahc2  ]]
```

Listing 35: scale3.tinte

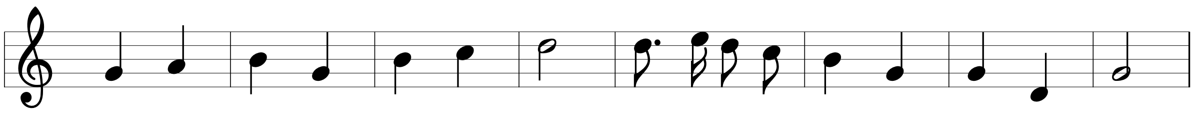


3.1.2 Bars

If you need a bar between two musical elements in sequential composition, then you can use the operator `-|-`.

```
bars1 =[[n4g1—a−|−h—g−|−h—c2−|−n2d2−|−n8d2.−−n16e2—n8d2—c−|−n4h1—
     g−|−g—d−|−n2g1]
    ]
```

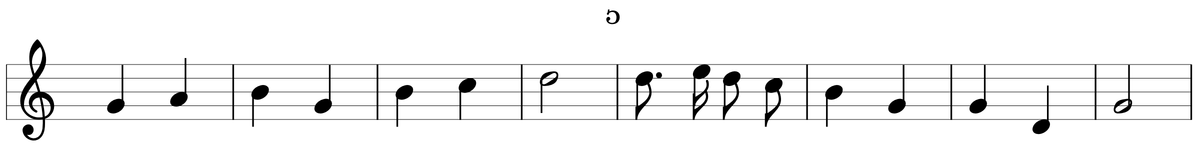
Listing 36: bars1.tinte



But it is preferred to have a list of bars.

```
bars2 =[[n4g1—a,h—g,h—c2,n2d2,n8d2.−−n16e2—n8d2—c,n4h1—g,g—d,
    n2g1]
        ]
```

Listing 37: bars2.tinte



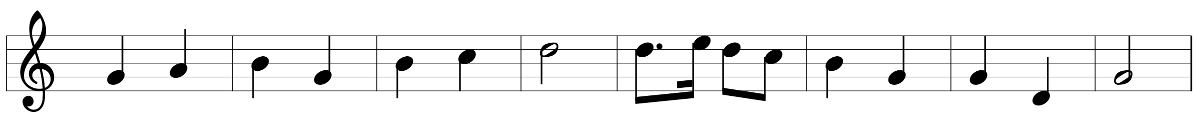
The list notation is rather helpfulwhen there is a syntactical error in a bar. Tinte will then try to skip the bar and go on with the following bars. By the use of the combinator `-|-` this cannot be done.

3.1.3 Beams

Musical notation uses beams to group notes of shorter duration than quarter notes. In tinte this is denoted by enclosing the group of elements, which are supposed to be grouped in square brackets. If the beam is to be set above the group of notes the closing bracket is followed by the symbol `^`, if it is underneath, it is followed by the symbol `_`.

```
beam=[
  [n4g1—a−|−h—g−|−h—c2−|−n2d2−|−[k8d2.−−k16e2]_−−[k8d2—c]_−|−n4h1—
     g−|−g—d−|−n2g1]
]
```

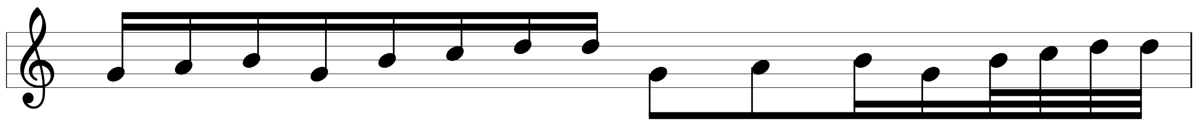
Listing 38: beam1.tinte



Note the you use k-elements in beamed groups. These are notes without a stem. Stems are created by the beam.

```
beam=[
    [  [(k16g1  a  h  g  h  c2  d2  d2)]^−−[(k8g1  a  k16h1  g  k32h1  c2  d2  d2)]_]
]
```

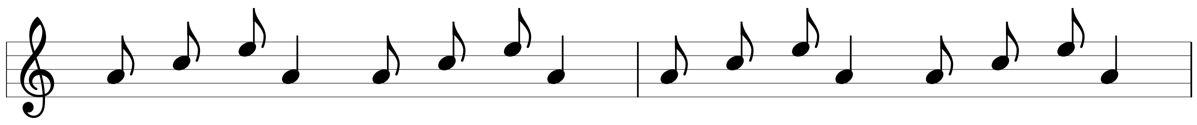
Listing 39: beam2.tinte



3.1.4 Triplets and General Tuplets

```
triplet=[[
  (n8^a1 n8^c2  n8^e2)\3 n4^a1 (n8^a1 n8^c2 n8^e2)\3 n4^a1
 ,(n8^a1 n8^c2  n8^e2)\3 n4^a1 (n8^a1 n8^c2 n8^e2)\3 n4^a1
]]
```

Listing 40: triplet.tinte

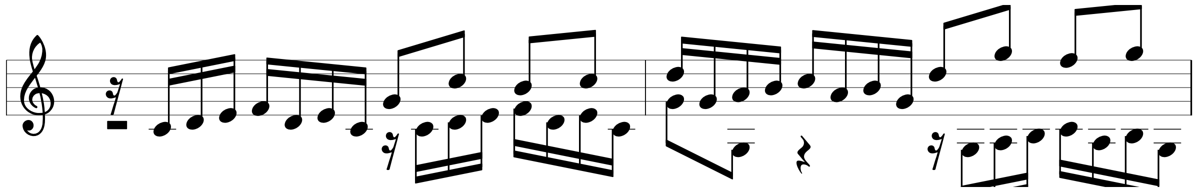


3.2 Polyphonic Combination

In polyphonic music two sequences are combined so that they are played simultanously. This is the vertical combination of music. In tinte traditionally there is the combinator `|/` for the vertical combination. But a simple `|` will do as well.

```
bachInvention1=[[
    (p16 [k16c1 d e]^ [f d e c]^ [k8g1 c2]^ [h1 c2]^
      -|-[k16d2 g1 a h]^ [c2 a1 h g]^ [k8d2 g]^ [f g]^)
 |  ( p2u p16u [k16c1 d e]_ [f d e c]_
      -|-[k8g1,g0]_ p4u p16u [k16g0 a h]_ [c1 a0 h g]_)
]]
```

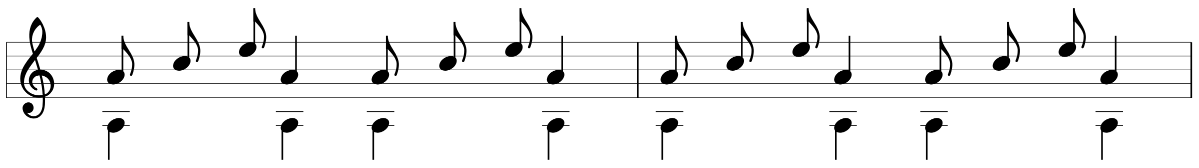
Listing 41: bachInvention1.tinte



Actually, now we can proof that the notation for triplets really constructs note elements with the correct duration:

```
triplet=[[
  (n8^a1 n8^c2  n8^e2)\3 n4^a1 (n8^a1 n8^c2 n8^e2)\3 n4^a1
   |n4_a0 n4_a0 n4_a0 n4_a0
 ,(n8^a1 n8^c2  n8^e2)\3 n4^a1 (n8^a1 n8^c2 n8^e2)\3 n4^a1
   |n4_a0 n4_a0 n4_a0 n4_a0
]]
```

Listing 42: triplet2.tinte

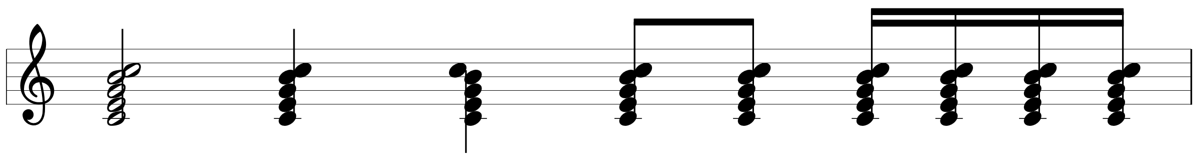


3.3 Chords

Chords are encapsulated in pointy brackets. The notes within these brackes are seperated by commas. An optional _ or ^ symbol signify the direction of the stem. If it is missing, no stem will be generated.

```
cmaj7=[[(<k2c1,e,g,h,c2>^--<k4c1,e,g,h,c2>^--<c1,e,g,h,c2>_--[<k8c1,e,
    g,h,c2>,<c1,e,g,h,c2>]^--[<k16c1,e,g,h,c2>,<c1,e,g,h,c2>,<c1,e,g,h,
    c2>,<c1,e,g,h,c2>]^
      )]]
```

Listing 43: cmaj7.tinte



Unfortunately due to a bug in the parser of tinte, groups of elements which contain chords will in most situation need to be enclosed in parantheses.

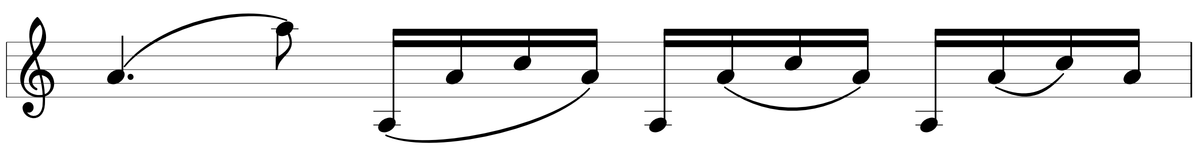
4 Articulation

4.1 Staccato

4.2 Slurs

```
slurs=[[
      \(^n4a1.--n8a2\)
   --\(_[k16a0,a1,c2,a1]^\)
   --[k16a0,\(_a1—c2—a1\)]^
   --[k16a0,\(_a1—c2\),a1]^
  ]]
```

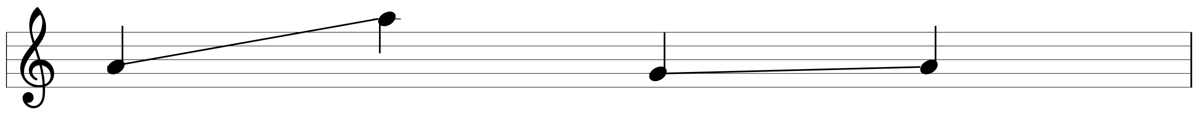
Listing 44: slurs.tinte



4.3 Glissando

```
glissando=[[
      (\~n4a1 a2\~!)        (\~n4g1 a1\~!)
  ]]
```

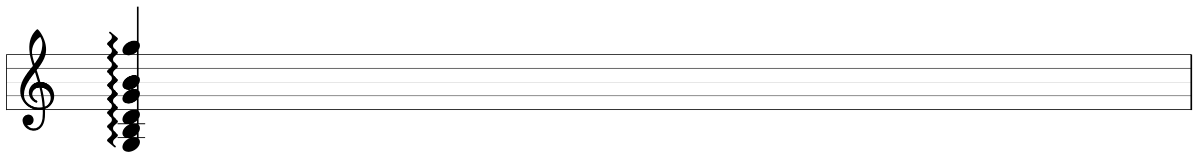
Listing 45: glissando1.tinte



4.4 Arpeggio

```
arpeggio1=[[
     ( arpeggio (<k4g0,h,d1,g,h,g2>^) )
  ]]
```

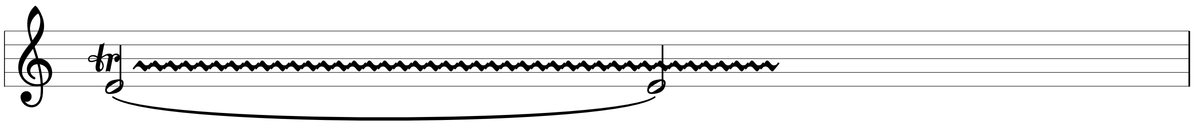
Listing 46: arpeggio1.tinte



4.5 Trill

```
trill=[[
      \tr \(_n2e1 e1\)\tr!
  ]]
```

Listing 47: trill1.tinte

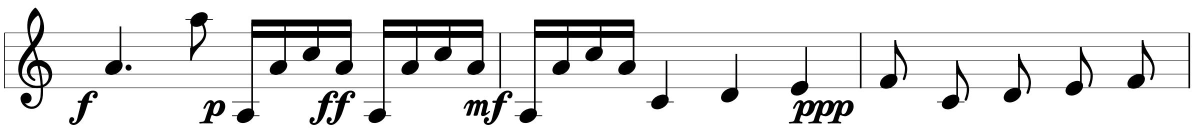


4.6 Dynamics

4.6.1 Fortes and pianos

```
forteAndPiano=[[
    \f n4a1.n8a2\p [k16a0a1c2a1]^ \ff[k16a0,a1c2a1]^
  , \mf [k16a0,a1c2,a1]^ n4c1de
  , \ppp f  n8c1def
]]
```

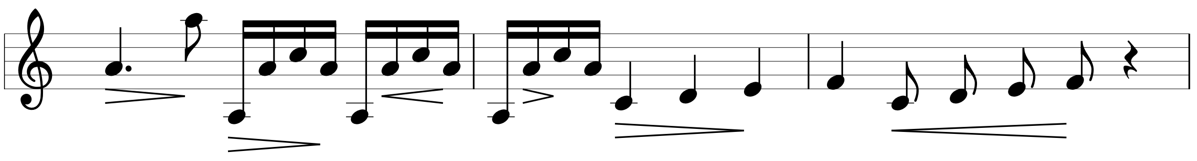
Listing 48: forteAndPiano.tinte



4.6.2 Crescendo and Decrescendo

```
crescendo=[[
    \>n4a1.n8a2\! \>[k16a0a1c2a1]^\! [k16a0,\<a1c2a1\!]^
  , [k16a0,\>a1c2\!,a1]^ \>n4c1de\!
  , f  \<n8c1def\!p4
]]
```

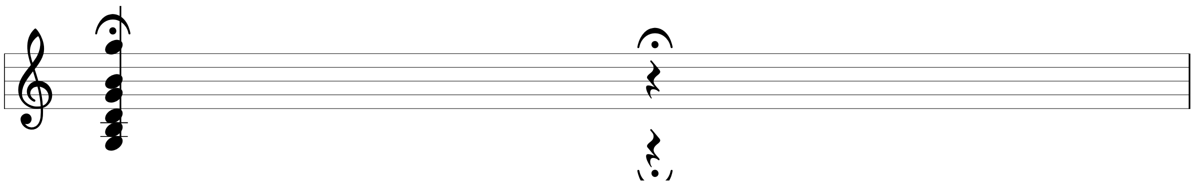
Listing 49: crescendo.tinte



4.7 Fermate

```
fermate1=[[
    ( fermateo (<k4g0,h,d1,g,h,g2>^) ) (fermateo p4|fermateu p4u)
  ]]
```

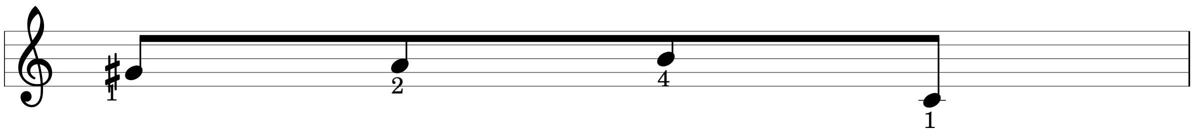
Listing 50: fermate1.tinte



4.8 Fingering

```
finger1=[[
   [(fingeru 1 #k8g1) (fingeru 2 a) (fingeru 4 h) (fingeru 1 c)]^
]]
```

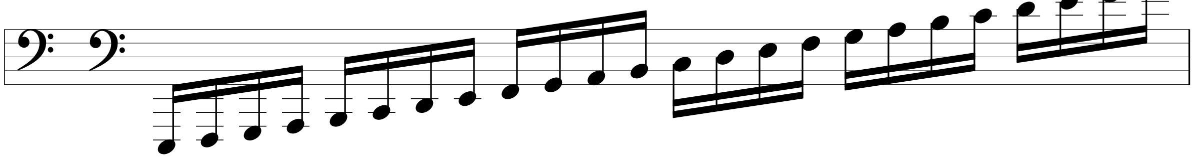
Listing 51: finger1.tinte



5 Clefs

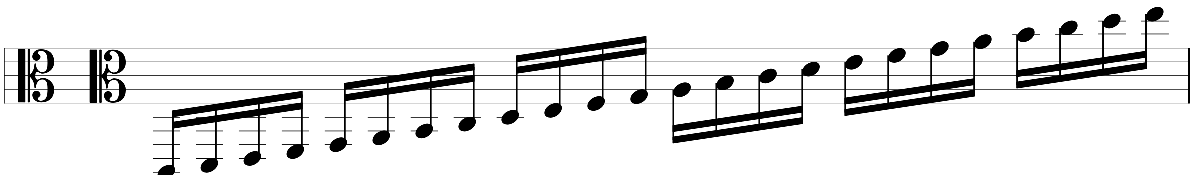
```
clef1=[[
    \changeclef Bass  [k16E1FGA]^[HC0DE]^[FGAH]^[c0def]_[gahc1]_[defg]_
  ]]
```

Listing 52: clef1.tinte



```
clef2 =[[
    \changeclef Tenor [k16c0def]^[gahc1]^[defg]^  [ahc2d]_ [efga]_[
   hc3de]_
  ]]
```

Listing 53: clef2.tinte



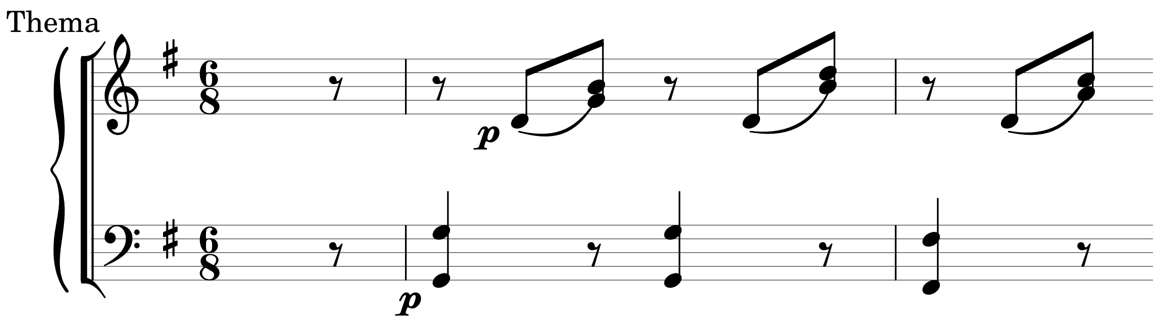
6 Overall Layout

6.1 Staffs

6.1.1 Piano Music

```
out =
 { title ="First Example for Piano Music Notation"
 , schluessel                    =[Violin,Bass]
 , instrument                    =["piano","piano"]
 , doubleStaff                   = [1]
 , tonartsymbol                  = GDur
 , systemanzahl                  = 2
 , stimme                        = 1
 , taktNummer                    = 0
 }
piano =[[
    stimme 1 (\movement "Thema" metrum 6 8 p8)
   |stimme 2 (metrum 6 8 p8)
  , stimme 1 (p8 \p [\(_k8d1 <g,h>\)]^ p8 \(_[k8d1 <h,d2>]^\))
   |stimme 2 (\p <k4G0,g0>^ p8 <k4G0,g0>^ p8)
  , stimme 1 (p8 \(_[k8d1 <a1,c2>]^\)  p8 \(^[<k8d2,h><h1,g2>]_\))
   |stimme 2 (<k4F0,f0>^ p8 <k4G0,g0>^ p8)
  ]]
```

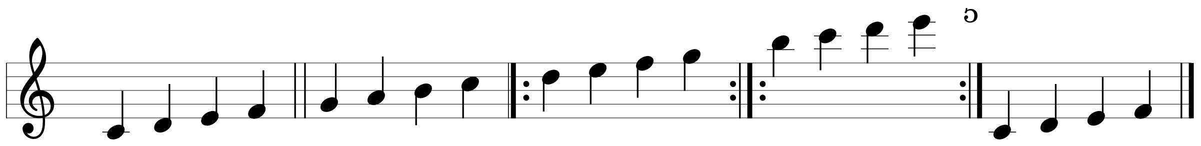
Listing 54: piano1.tinte



6.2 Bar Lines

```
barlines1=[[
    n4c1 d e f ||, g a h c2 ||:, d e f g :||:, h c3 d e :||, c1 d e f
    |||
]]
```

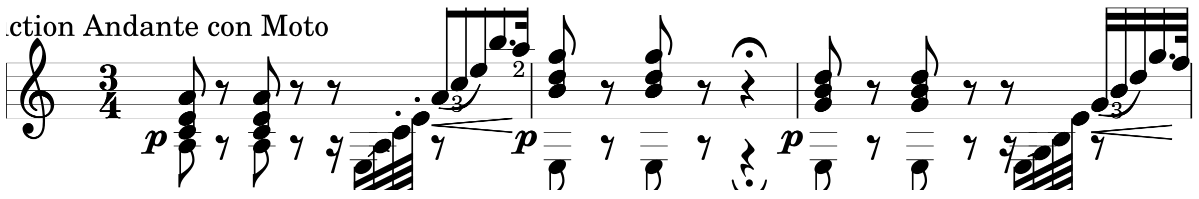
Listing 55: barlines1.tinte



7 Complex Example

```
op10 =
 [
  [
   (\movement "Introduction Andante con Moto" metrum 3 4 —<k8c1,e,a
   >^—p8—<k8c1,e,a>^—p8—p8
      —[\<triole(\(_k32a1— c2-3_—e\))—k32h2.--fingeru 2 k64a2\!]^)
   |/
   (metrum 3 4 --\p<k8a0>_—p8u—<k8a0>_—p8u—p16u--[\(^k64e0—a\)—
   stao
 c1—stao e]_—p8u)
  ,(
    anwo "sans chanterelle" <k8h1,d2,g>^—p8—<k8h1,d2,g>^— p8 —
   fermateo p4)|/
   (\p <k8e0>_—p8u—<k8e0>_—p8u—fermateu p4u)
  ,(<k8g1,h,d2>^—p8—<g1,h,d2>^—p8—p8
      —[\<triole(\(_k32g1—fingeru 3 h—d2\))—k32g2.--k64f2\!]^)|/
   (\p<k8e0>_—p8u—<k8e0>_—p8u—p16u--[\(^k64e0—g\)—h—e1]_—p8u)
 ]]
```

Listing 56: zaniop10.tinte



References

[HMGW96] Paul Hudak, Tom Makucevich, Syam Gadde, and Bo Whong. Haskore music notation – an algebra of music –. *Journal of Functional Programming*, 6(3):465–484, 1996.

[lil] LilyPond. `http://lilypond.org/`. Accessed: 2018-03-08.

[Pan97] Sven Eric Panitz. Tinte: Developing a prototype for typesetting music in clean (a case study). Technical report, J. W. Goethe-Universität, FB 20, Professur KIST, 1997.