Music Xml to Braille Compiler

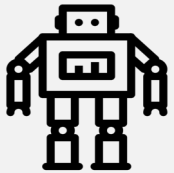
<Scanning & Parsing by JAVA CC>

Ahn Hyung Geun & Seo Han Seoung

INDEX

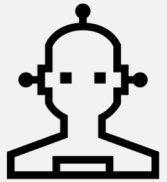
1

Example XML



2

Tokenizing



3

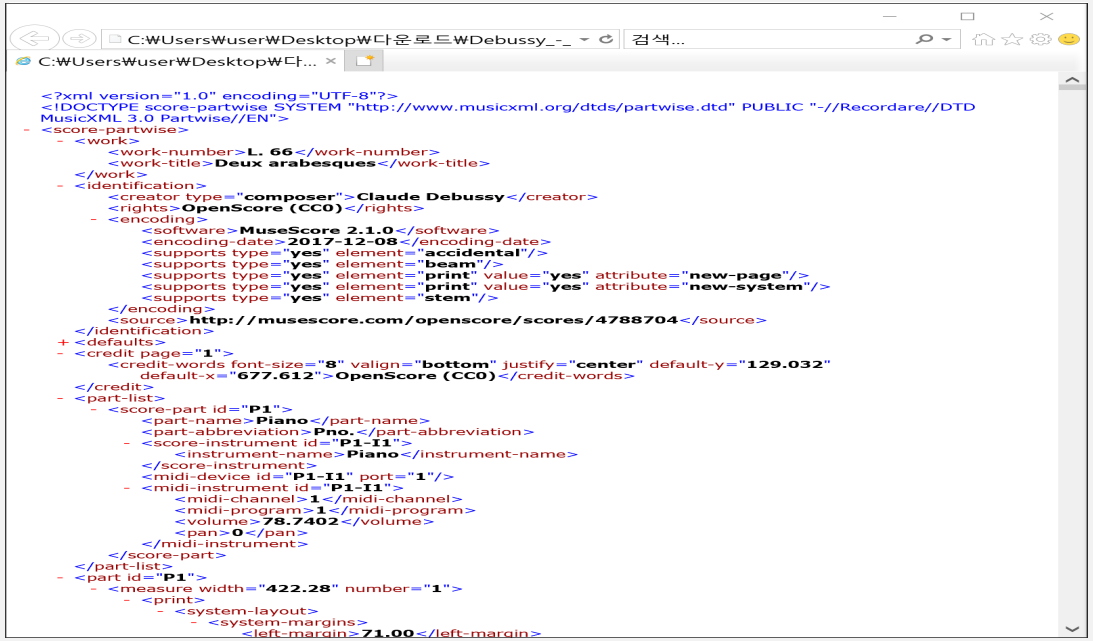
Source Code

4

Parsing Idea

What is XML?

✓ Example of Music XML code



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE score-partwise SYSTEM "http://www.musicxml.org/dtds/partwise.dtd" PUBLIC "-//Recordare//DTD
MusicXML 3.0 Partwise//EN">
- <score-partwise>
  - <work>
      <work-number>L. 66</work-number>
      <work-title>Deux arabesques</work-title>
    </work>
  - <identification>
      <creator type="composer">Claude Debussy</creator>
      <rights>OpenScore (CC0)</rights>
    - <encoding>
        <software>MuseScore 2.1.0</software>
        <encoding-date>2017-12-08</encoding-date>
        <supports type="yes" element="accidental"/>
        <supports type="yes" element="beam"/>
        <supports type="yes" element="print" value="yes" attribute="new-page"/>
        <supports type="yes" element="print" value="yes" attribute="new-system"/>
        <supports type="yes" element="stem"/>
      </encoding>
      <source>http://musescore.com/openscore/scores/4788704</source>
    </identification>
  + <defaults>
  - <credit page="1">
      <credit-words font-size="8" valign="bottom" justify="center" default-y="129.032"
        default-x="677.612">OpenScore (CC0)</credit-words>
    </credit>
  - <part-list>
    - <score-part id="P1">
        <part-name>Piano</part-name>
        <part-abbreviation>Pno.</part-abbreviation>
      - <score-instrument id="P1-I1">
          <instrument-name>Piano</instrument-name>
        </score-instrument>
        <midi-device id="P1-I1" port="1"/>
      - <midi-instrument id="P1-I1">
          <midi-channel>1</midi-channel>
          <midi-program>1</midi-program>
          <volume>78.7402</volume>
          <pan>0</pan>
        </midi-instrument>
      </score-part>
    </part-list>
  - <part id="P1">
    - <measure width="422.28" number="1">
      - <print>
        - <system-layout>
          - <system-margins>
              <left-margin>71.00</left-margin>
```

Part of Deux arabesques(Classic Music)

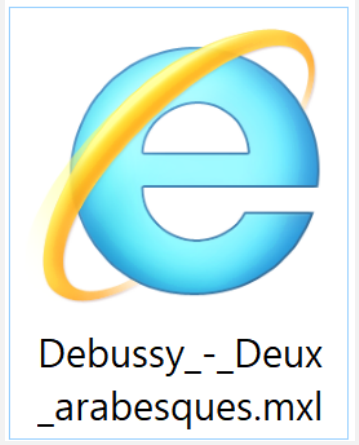
Not HTML. Seems like similar but XML and HTML has different purpose.

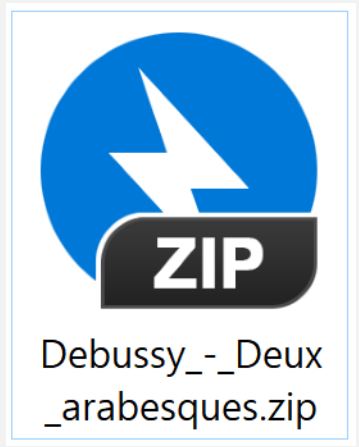
HTML got some promised tag. But XML can define that tag format.

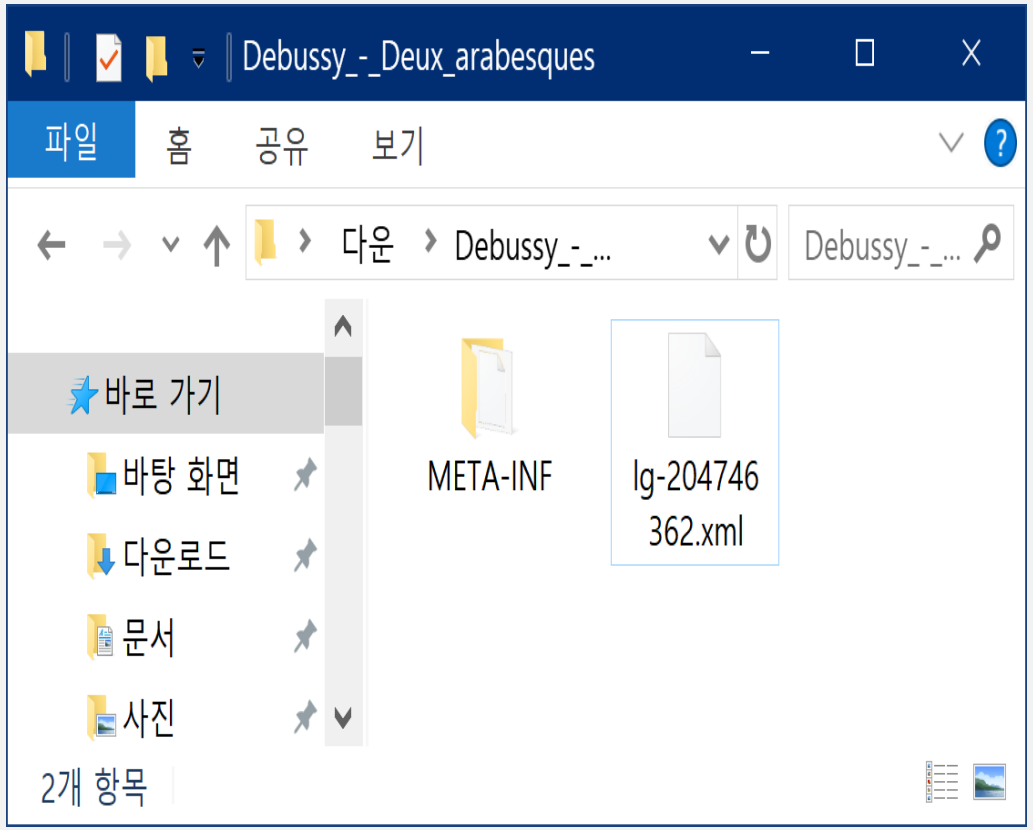
Purpose of HTML: Data Expression
Purpose of XML: Data Exchange

XML to Music XML

✓ **Music XML is zip of XML code**







Schema of Music XML

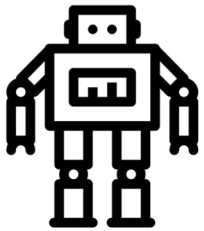
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE score-partwise SYSTEM "http://www.musicxml.org/dtds/partwise.dtd" PUBLIC "-//Recordare//DTD MusicXML 3.0
Partwise//EN">
- <score-partwise>
  - <work>
      <work-number>L. 66</work-number>
      <work-title>Deux arabesques</work-title>
    </work>
  - <identification>
      <creator type="composer">Claude Debussy</creator>
      <rights>OpenScore (CC0)</rights>
    - <encoding>
        <software>MuseScore 2.1.0</software>
        <encoding-date>2017-12-08</encoding-date>
        <supports type="yes" element="accidental"/>
        <supports type="yes" element="beam"/>
        <supports type="yes" element="print" value="yes" attribute="new-page"/>
        <supports type="yes" element="print" value="yes" attribute="new-system"/>
        <supports type="yes" element="stem"/>
      </encoding>
      <source>http://musescore.com/openscore/scores/4788704</source>
    </identification>
  - <defaults>
    - <scaling>
        <millimeters>6.2</millimeters>
        <tenths>40</tenths>
      </scaling>
    - <page-layout>
        <page-height>1915.67</page-height>
        <page-width>1355.22</page-width>
      - <page-margins type="even">
          <left-margin>64.5161</left-margin>
          <right-margin>64.5161</right-margin>
          <top-margin>64.5161</top-margin>
          <bottom-margin>129.032</bottom-margin>
        </page-margins>
      - <page-margins type="odd">
          <left-margin>64.5161</left-margin>
          <right-margin>64.5161</right-margin>
          <top-margin>64.5161</top-margin>
          <bottom-margin>129.032</bottom-margin>
        </page-margins>
      </page-layout>
      <word-font font-size="10" font-family="FreeSerif"/>
      <lyric-font font-size="11" font-family="FreeSerif"/>
    </defaults>
```

Schema of Music XML

```
- <part-list>
  - <score-part id="P1">
      <part-name>Piano</part-name>
      <part-abbreviation>Pno.</part-abbreviation>
    - <score-instrument id="P1-I1">
        <instrument-name>Piano</instrument-name>
      </score-instrument>
      <midi-device id="P1-I1" port="1"/>
    - <midi-instrument id="P1-I1">
        <midi-channel>1</midi-channel>
        <midi-program>1</midi-program>
        <volume>78.7402</volume>
        <pan>0</pan>
      </midi-instrument>
  </score-part>
</part-list>
```



```
- <part id="P1">
  - <measure width="422.28" number="1">
    - <print>
      - <system-layout>
        - <system-margins>
            <left-margin>71.00</left-margin>
            <right-margin>0.00</right-margin>
          </system-margins>
          <top-system-distance>245.00</top-system-distance>
        </system-layout>
      - <staff-layout number="2">
          <staff-distance>90.00</staff-distance>
        </staff-layout>
      </print>
    - <attributes>
        <divisions>6</divisions>
      - <key>
          <fifths>4</fifths>
        </key>
      - <time symbol="common">
          <beats>4</beats>
          <beat-type>4</beat-type>
        </time>
        <staves>2</staves>
      - <clef number="1">
          <sign>G</sign>
          <line>2</line>
        </clef>
      - <clef number="2">
          <sign>G</sign>
          <line>2</line>
        </clef>
      </attributes>
```



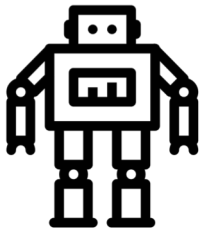
Idea about Tokenizing

✓ **Part – Part list**

```
<part-list>
  <score-part id="P1">
    <part-name>
      Piano
    <part-abbreviation>
      Pno.
    <score-instrument id="P1-I1">
      <instrument-name>
        Piano
    <midi-device id="P1-I1" port="1" />
    <midi-instrument id="P1-I1">
      <midi-channel>
        1
      <midi-program>
        1
      <volume>
        78.7402
      <pan>
        0
```

✓ **Work, Identification, Default …**

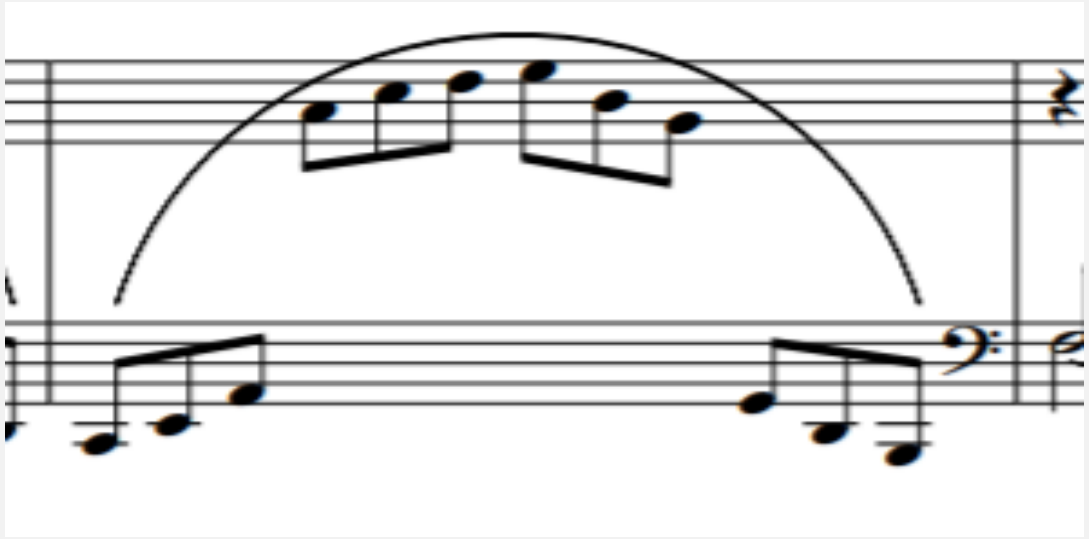
```
<note default-x="119.00" default-y="-180.00">
  <pitch>
    <step>
      C
    <alter>
      1
    <octave>
      4
  <duration>
    2
  <voice>
    1
  <type>
    eighth
  <time-modification>
    <actual-notes>
      3
    <normal-notes>
      2
  <stem>
    up
  <staff>
    2
  <beam number="1">
    begin
```



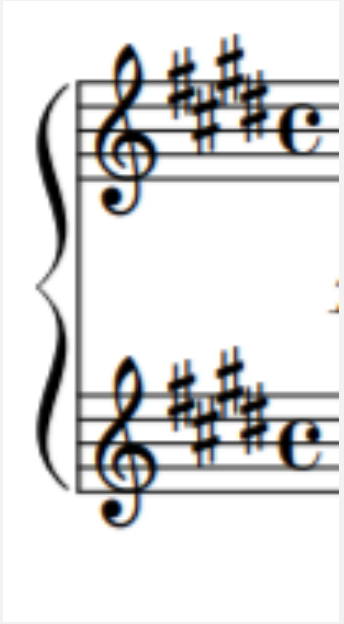
Idea about Tokenizing

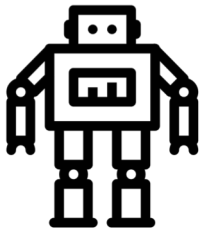
✓ **Below Note – tuplet, slur**

```
<notations>
  <tuplet type="start" bracket="no" show-number="none" />
  <slur type="start" number="1" />
```



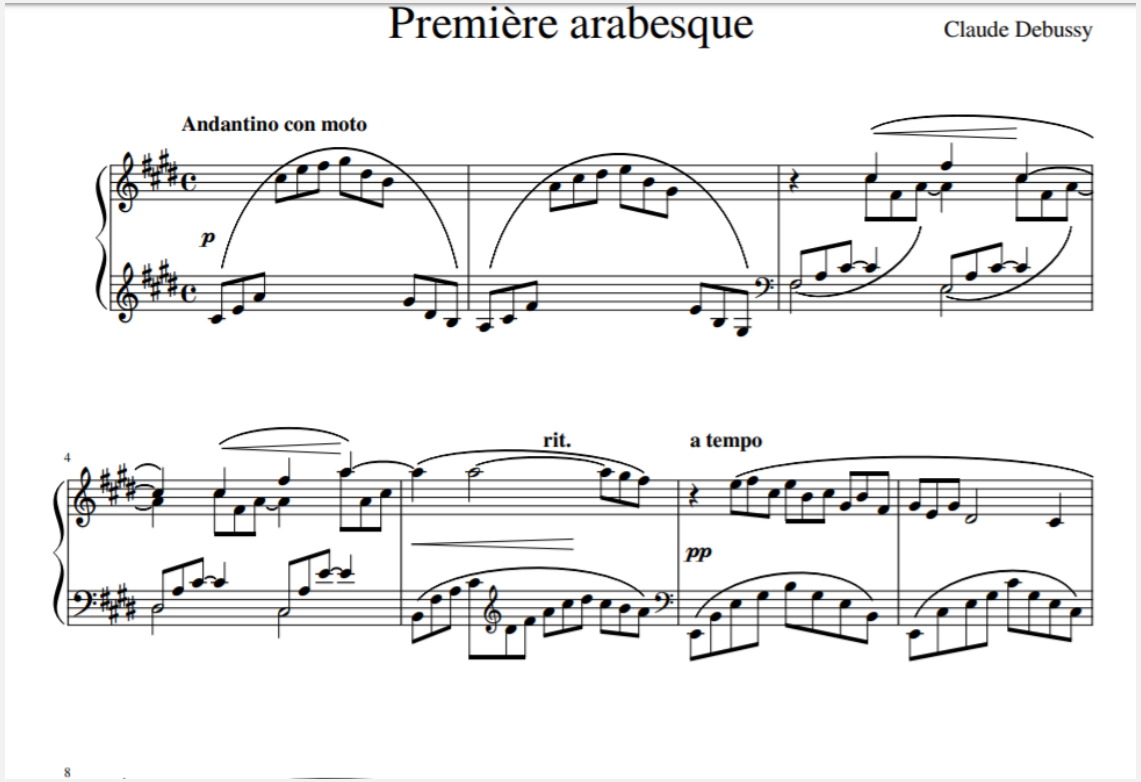
✓ **Attributes and Fifths, Clef**





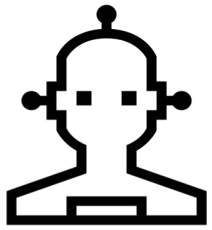
Idea about Tokenizing

✓ **To jump for exact measure**



✓ **Only use Print new system**

```
<print new-system="yes">
  <system-layout>
    <system-margins>
      <left-margin>
        21.00
      <right-margin>
        0.00
    <system-distance>
      200.00
  <staff-layout number="2">
    <staff-distance>
      90.00
```



Tokenizing Code In JAVACC

✓ **Source Code of Token**

```
TOKEN :
{
    < PART : "part" >
|   < MEASURE : "measure" >
|   < NOTE : "note" >
|   < ATTRIBUTE : "attribute" >
|   < DIVISIONS : "divisions" >
|   < KEY : "key" >
|   < FIFTHS : "fifths" >
|   < TIME : "time" >
|   < BEATS : "beats" >
|   < BEATTYPE : "beat-type" >
|   < STAVES : "staves" >
|   < CLEF : "clef" >
|   < SIGN : "sign" >
|   < LINE : "line" >
|   < PRINT : "print" >
}

TOKEN :
{
    < NUMBER : "number=\"[1-3]\"" >
|   < NEWLINE : "new-system=\"yes\"" >
}
```

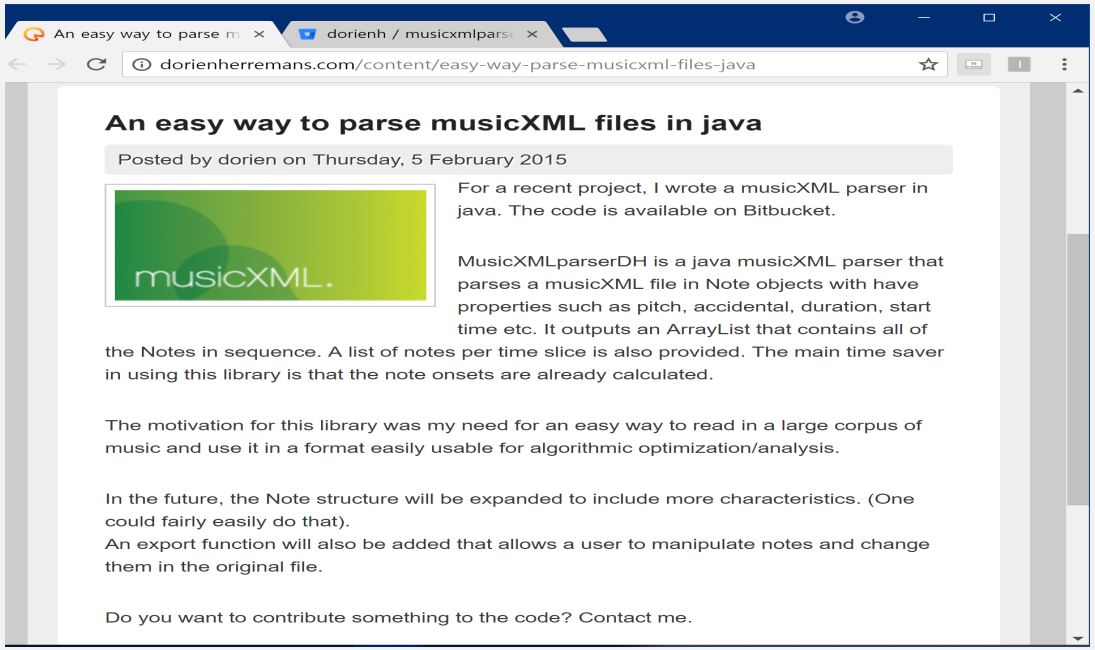
```
TOKEN :
{
    < _MEASURE : < PRINT >< ATTRIBUTES >< NOTE > >
|   < _ATTRIBUTES : < DIVISIONS >< KEY >< TIME >< STAVES >< CLEF > >
|   < _TIME : < BEATS >< BEATTYPE > >
|   < _CLEF : < SIGN >< LINE > >
}

TOKEN :
{
    < PART : "part" >
|   < MEASURE : "measure" >
|   < PRINT : "print" >
|   < ATTRIBUTES : "attributes" >
|   < DIVISIONS : "divisions" >
|   < KEY : "key" >
|   < FIFTHS : "fifths" >
|   < TIME : "time" >
|   < BEATS : "beats" >
|   < BEATTYPE : "beat-type" >
|   < STAVES : "staves" >
|   < CLEF : "clef" >
|   < SIGN : "sign" >
|   < LINE : "line" >
|   < NOTE : "note" >
|   < PITCH : "pitch" >
|   < STEP : "step" >
|   < OCTAVE : "octave" >
|   < DURATION : "duration" >
|   < TYPE : "type" >
}

TOKEN :
{
    < NUMBER : "number=\"[1-3]\"" >
|   < NEWLINE : "new-system=\"yes\"" >
}
```

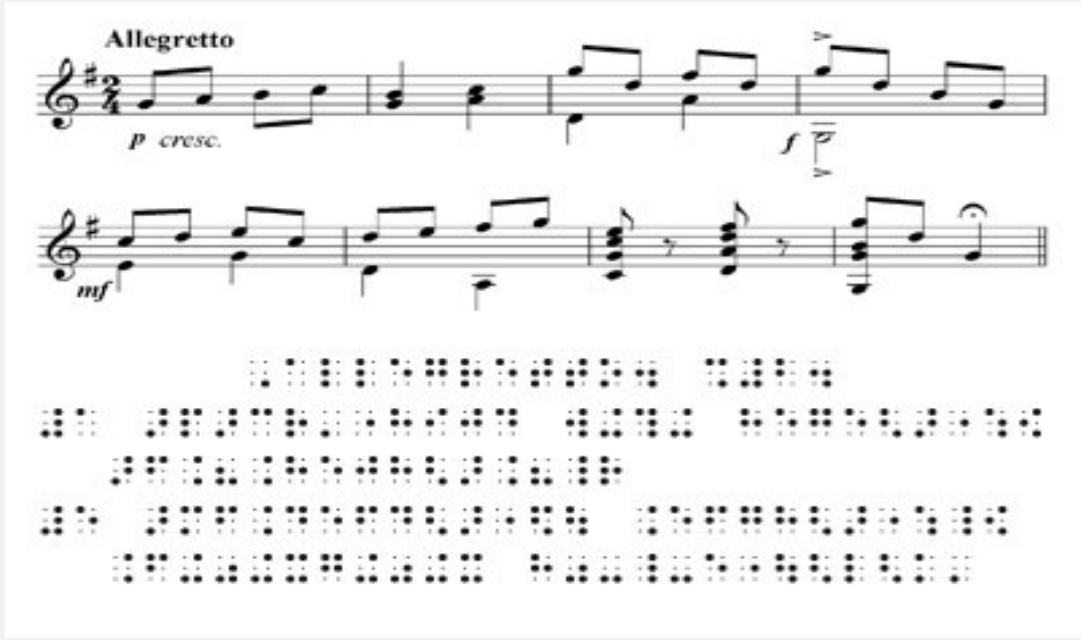
Parsing = Have to be simple

✓ XML Parser in Bitbucket



Not using JAVACC. But useful for study

✓ Study about Braille Music



Not easy for find data. Only access for the disalbed

Thank you