**import** scale

**import** chord

**import** ChordTypes

**import** random

listOfChordsOfChordTypes = []

listOfChords = []

sampleChords = []

flatSampleChords = []

ct = []

SAMPLE\_SIZE = 3 # temporary value for now

**def** chordgen():

**for** rootNote **in** scale.TWELVE\_NOTE\_SCALE:

**for** chordType **in** scale.CHORD\_TYPES:

listOfChords.append(chord.Chord(rootNote, chordType))

**print**(listOfChords)

**def** sample(SAMPLE\_SIZE):

# Makes list of list of chords. The top level list is broken down into lists of chords grouped by type

**for** type **in** range(0, ChordTypes.NUMBER\_OF\_CHORD\_TYPES):

listOfChordsOfChordTypes.append([])

**for** chord **in** range(0, ChordTypes.NUMBER\_OF\_CHORDS, ChordTypes.NUMBER\_OF\_CHORD\_TYPES):

listOfChordsOfChordTypes[type].append(listOfChords[chord + type])

**print**(listOfChordsOfChordTypes)

**print**(len(listOfChordsOfChordTypes))

**for** ctListIndex **in** range(len(listOfChordsOfChordTypes)):

sampleChords.append(random.sample(listOfChordsOfChordTypes[ctListIndex], SAMPLE\_SIZE))

flatSampleChords = [\_chord\_ **for** \_chordList\_ **in** sampleChords **for** \_chord\_ **in** \_chordList\_]

**print**(sampleChords)

**print**(flatSampleChords)

**print**("Sample size:", str(len(flatSampleChords)), "(" , str(ChordTypes.NUMBER\_OF\_CHORD\_TYPES), "chord types \*", SAMPLE\_SIZE, "chords per type)")

**if** \_\_name\_\_ == "\_\_main\_\_":

chordgen()

**if** SAMPLE\_SIZE < 12:

sample(SAMPLE\_SIZE)

**else**:

**print**("Sample invalid.")