Chapter I. Some Prerequisite Knowledge of Lilypond and Python

In my experience composing scores with the help of computational systems, I have found that the greatest power and flexibility are available from the Abjad API for formalized score control. Abjad is significant because of the freedom with which the composer is able to manipulate their musical material and the ability to not only control the musical elements of a score, but also other graphic features as well. When choosing to compose music with Abjad, there are some elements of the underlying software with which the musician should familiarize themselves. Every score that is created with Abjad is engraved by the Lilypond music notation engine. Because of this interdependence, the composer should be familiar with Lilypond’s model of music notation as well as elements of Lilypond syntax. Since Abjad is an API in the python programming language, it is essential that the composer be familiar with writing python code. In this chapter, the basics of Lilypond and Python will be discussed, while information directly related to the Abjad API follows in chapter two.

A: Lilypond’s model of music Notation

1. compare notational model with other models of music notation.

There are

a.) pitch systems from no accidental to accidentals

b.) rhythmic systems (written versus prolated durations)

2.) compare user interface with other notation software.

x.) context concatenation

a.) bach in max

b.) openmusic & PWGL

3.) compare lilypond syntax with LaTeX or HTML

B: Python

1.) types of programming languages (Procedural, Functional, Object-oriented)

2.) python fundamentals

a.) integers, lists, strings (and to a lesser extent, dictionaries)

b.) slicing (as told by trevor in the email list)

c.) list comprehensions and loops

d.) concatenate, append, extend

e.) functions