

Sounds, Sequences, Searching, and Sorting

- Physics of sounds
- Sounds as objects in Python
 - Getting information about sounds (samples, sampling rate, duration, length)
 - Playing and exploring sounds
- Looping over sounds
 - Getting individual samples
 - For a sample, getting value, left and right channel
 - Creating integer list of sample values
 - Creating a sound from a list of integer values
- Manipulations on sounds
 - splicing
 - scaling (frequency and amplitude)
 - merging
 - echo
 - others
- Sequences
 - Operations that work for all sequences
 - `for x in <sequence>`
 - `len(<sequence>)`
 - `<val> in <sequence>`
 - `<seq1> + <seq2>`
 - `<seq> * n`
 - `s[i]` -- accessing element
 - `s[i:j]` -- slicing
 - sequence methods: `s.index(<val>)`, `s.count(<val>)`
 - Operations that work for lists and change them
 - `s[i] = <val>` -- changing existing elements
 - `del s[i]` -- deleting an individual element
 - `del s[i:j]` -- deleting a sublist
 - list methods:
 - `s.append(<val>)`
 - `s.insert(i, <val>)`
 - `s.pop(i)`
 - `s.remove(<val>)`
 - `s.reverse()`
 - Operations that are string methods, typically returning new strings
 - `str.upper()`
 - `str.find(<substring>[,start])`
 - `str.lower()`
- Search
 - linear search -- and average case and worst case number of steps to find something
 - binary search
 - conceptual solution
 - given an example problem, performing algorithm "by hand"
 - coding a solution
 - worst case number of steps to find something
- Sorting
 - Insertion Sort

- conceptual solution
- given an example problem, performing algorithm "by hand"
- coding a solution (or at least really understanding the code we developed)