
Parker Documentation

Release 0.0.1

Chris Gilmer

June 24, 2016

1	Tutorial	3
1.1	Features	3
1.2	Tutorial 01 - Notes	3
1.3	Tutorial 02 - Intervals	6
1.4	Tutorial 03 - Chords	6
1.5	Tutorial 04 - Scales	6
1.6	Tutorial 05 - Keys	6
1.7	Tutorial 06 - Progressions	6
2	API Docs	7
2.1	parker.notes	7
2.2	parker.chords	13
2.3	parker.scales	15
2.4	parker.keys	20
2.5	parker.progressions	20
3	Indices and tables	21

Parker is a music theory library for python. It's primary use is for music theory education and tooling. It is not a music composition tool, though it can be used for that purpose.

Tutorial

1.1 Features

- Work with notes, chords, scales, and keys.
- Generate all chords given a root note or generate chords from shorthand notation.
- Generate scales using a root note using natural class names (Diatonic, Major, Minor, Dorian, Mixolydian, MajorPentatonic, etc)

[Back to Index](#)

1.2 Tutorial 01 - Notes

The basic building block of any music is the note. A note is defined by its name, octave, and accidentals following Scientific Pitch Notation. Let's play with notes:

```
>>> from parker.notes import Note
```

1.2.1 Note objects

A note can be easily constructed from a valid note name. Let's try a few:

```
>>> n = Note('C')
>>> print(n)
C4
>>> repr(n)
"Note('C4')"
```

You'll notice immediately that the note that was returned says *C4* instead of *C*. This is because the default octave will be the middle or fourth (4) octave of the scale. It also has no accidentals (sharps or flats).

There are a lot of valid notes that you can construct:

```
>>> note_list = ['C', 'C4', 'C#4', 'Cbb4']
>>> print([Note(note) for note in note_list])
[Note('C4'), Note('C4'), Note('C#4'), Note('Cbb4')]
```

Suprisingly you can use some very interesting notes:

```
>>> Note('C#####bb')
Note('C####4')
>>> Note('C#b#bb##b##bb')
Note('C4')
```

You can use a convenience method to check if the format of the note is valid before constructing a note:

```
>>> from parker.notes import note_is_valid
>>> note_is_valid('C')
True
>>> note_is_valid('c')
False
```

1.2.2 Note Properties

As mentioned previously every Note object is constructed from a name, octave, and accidentals. You can get to these properties directly if desired:

```
>>> n = Note('C#4')
>>> n.base_name
>>> 'C'
>>> n.octave
>>> 4
>>> n.accidentals
>>> 1
```

The accidentals are captured as integer steps, or semitones, above the base note. The accidentals are changed internally into regognizable strings using a private method:

```
>>> n = Note('C#4')
>>> str(n.accidentals)
'#'
```

1.2.3 Notes as Numbers

Notes can also be represented as numbers:

```
>>> float(Note('C4'))
60.0
```

Or constructed from numbers:

```
>>> Note(60.0)
Note('C4')
```

1.2.4 Constructing Notes

You can construct notes using strings, integers, and other Note objects. As demonstrated all of these are valid:

```
>>> Note('C')
Note('C4')
>>> Note(60)
Note('C4')
```



```
>>> Note(Note('C'))
Note('C4')
```

1.2.5 Augment and Diminish Notes

Should you want to augment or diminish a note you can use easy convenience methods:

```
>>> n = Note('C4')
>>> n.augment()
Note('C#4')
>>> n.diminish()
Note('Cb4')
>>> n
Note('C4')
```

The methods to augment and diminish return new note objects instead of modifying the note in place. To modify in place you must use a different method:

```
>>> n = Note('C4')
>>> n.set_augment()
Note('C#4')
>>> n
Note('C#4')
>>> n.set_diminish()
Note('C4')
>>> n
Note('C4')
```

1.2.6 Other Reprerentations

There are different use cases for representing a note. In some cases the note is needed without the octave in the string. This is called Generalization:

```
>>> n = Note('C####4')
>>> n.generalize()
'C####'
```

In other cases what is needed is the note in its most succinct form. This is called Normalization:

```
>>> n = Note('C####4')
>>> n.normalize()
'E'
```

1.2.7 Note Frequencies

Every note has a frequency. To access the frequency of a note simply ask:

```
>>> n = Note('C4')
>>> n.get_frequency()
261.6255653005985
```

This is a bit cumbersome so it can be managed by setting the desired digits:

```
>>> n = Note('C4')
>>> n.get_frequency(ndigits=3)
261.626
```

You can get a note object from a frequency by using a convenience method:

```
>>> from parker.notes import note_from_frequency
>>> note_from_frequency(261.626)
Note('C4')
```

This method will only get the closest note but will not tell how far out of tune the note is given the frequency.

1.3 Tutorial 02 - Intervals

Intervals can be described as the difference between two notes. You may ask for the minor third above a note or the octave below a note. Fortunately the Note object comes with a number of built-in methods for getting the interval you want using transposition features. Let's get started:

```
>>> from parker.notes import Note
```

1.4 Tutorial 03 - Chords

```
>>> from parker.chords import *
```

1.5 Tutorial 04 - Scales

```
>>> from parker.scales import *
```

1.6 Tutorial 05 - Keys

```
>>> from parker.keys import *
```

1.7 Tutorial 06 - Progressions

```
>>> from parker.progressions import *
```

2.1 parker.notes

`notes.is_valid_note (note)`

Determine if a note is valid from a given string representation.

`notes.note_from_frequency (freq)`

Return the closest note given a frequency value in Hz

This uses the forumula $f = f_0 * (a ** n)$

f_0 - the reference frequency, which is A4 at 440 Hz a - the twelfth root of 2, or $2 ** (1/12)$ n - the number of half steps between notes

Here we want the value of n , or the integer value of half steps distance from the reference note.

$n = \log(f / f_0) / \log(a)$

This does not take into account out of tune notes. In the future it might make sense to return the cents above or below the note.

`class parker.notes.Accidental (acc='')`

`alter`

`name`

`set_augment ()`

`set_diminish ()`

`set_from_num (alter)`

`set_from_str (acc)`

`class parker.notes.Note (note=None, use_sharps=True)`

Representation of a single note.

`accidentals`

`all_transpositions ()`

Create all available named transpositions

`augment ()`

`augmented_eleventh_down ()`

`augmented_eleventh_up ()`

`augmented_ninth_down()`
`augmented_ninth_up()`
`base_name`
`clone()`
`compound_augmented_fourth_down()`
`compound_augmented_fourth_up()`
`compound_augmented_second_down()`
`compound_augmented_second_up()`
`compound_major_second_down()`
`compound_major_second_up()`
`compound_major_sixth_down()`
`compound_major_sixth_up()`
`compound_major_third_down()`
`compound_major_third_up()`
`compound_minor_second_down()`
`compound_minor_second_up()`
`compound_minor_sixth_down()`
`compound_minor_sixth_up()`
`compound_minor_third_down()`
`compound_minor_third_up()`
`compound_perfect_fourth_down()`
`compound_perfect_fourth_up()`
`diminish()`

`generalize()`

Return the note without the octave component.

Example: C4 -> C Cbb4 -> Cbb C####4 -> C###

`get_frequency(ndigits=None)`

Return the frequency of the note.

This uses the formula $f = f_0 * (a ** n)$

f_0 - the reference frequency, which is A4 at 440 Hz
 a - the twelfth root of 2, or $2 ** (1/12)$
 n - the number of half steps between notes

Should rounding be required you can set the number of digits to round to in the method.

Reference:

- <http://www.phy.mtu.edu/~suits/NoteFreqCalcs.html>
- https://en.wikipedia.org/wiki/Scientific_pitch_notation
- https://en.wikipedia.org/wiki/Music_and_mathematics

`major_eleventh_down()`

```
major_eleventh_up()
major_fifth_down()
major_fifth_up()
major_fourth_down()
major_fourth_up()
major_ninth_down()
major_ninth_up()
major_second_down()
major_second_up()
major_seventh_down()
major_seventh_up()
major_sixth_down()
major_sixth_up()
major_tenth_down()
major_tenth_up()
major_third_down()
major_third_up()
major_thirteenth_down()
major_thirteenth_up()
minor_fifth_down()
minor_fifth_up()
minor_ninth_down()
minor_ninth_up()
minor_second_down()
minor_second_up()
minor_seventh_down()
minor_seventh_up()
minor_sixth_down()
minor_sixth_up()
minor_tenth_down()
minor_tenth_up()
minor_third_down()
minor_third_up()
minor_thirteenth_down()
minor_thirteenth_up()
```

normalize (*use_sharps=None*)

Return the note normalized and without the octave component. Set *use_sharps* to control the output.

Example: C4 -> C Cbb4 -> Bb Cbb4 -> A# (*use_sharps=True*) C###4 -> D# C###4 => Eb (*use_sharps=False*)

octave

octave_down ()

octave_up ()

perfect_fifth_down ()

perfect_fifth_up ()

perfect_fourth_down ()

perfect_fourth_up ()

set_augment ()

set_diminish ()

set_transpose (*amount*)

Modify the note by a given number of semitones.

In some instances the letters ‘t’ or ‘A’ may be used to designate a change of 10 pitch classes. Similarly ‘e’ or ‘B’ may be used to designate a change of 11 pitch classes.

References:

- https://en.wikipedia.org/wiki/Pitch_class
- https://en.wikipedia.org/wiki/List_of_chords

transpose (*amount*)

transpose_list (*lst*)

class `parker.notes`.**NotesParser**

Parse notes of any type into a list of notes. Valid notes are: Note, NoteGroup, int, str, list, tuple, set.

static parse (*notes*)

class `parker.notes`.**NoteGroupBase**

Representation of a set of notes to be played at the same time. An example of a NoteGroup would be a chord (1, 3, 5) played on a piano.

The base class does not let you add notes.

all_transpositions ()

Create all available named transpositions

augment ()

augmented_eleventh_down ()

augmented_eleventh_up ()

augmented_ninth_down ()

augmented_ninth_up ()

clone ()

compound_augmented_fourth_down ()

compound_augmented_fourth_up ()

```
compound_augmented_second_down()  
compound_augmented_second_up()  
compound_major_second_down()  
compound_major_second_up()  
compound_major_sixth_down()  
compound_major_sixth_up()  
compound_major_third_down()  
compound_major_third_up()  
compound_minor_second_down()  
compound_minor_second_up()  
compound_minor_sixth_down()  
compound_minor_sixth_up()  
compound_minor_third_down()  
compound_minor_third_up()  
compound_perfect_fourth_down()  
compound_perfect_fourth_up()  
diminish()  
get_notes()  
highest_note()  
lowest_note()  
major_eleventh_down()  
major_eleventh_up()  
major_fifth_down()  
major_fifth_up()  
major_fourth_down()  
major_fourth_up()  
major_ninth_down()  
major_ninth_up()  
major_second_down()  
major_second_up()  
major_seventh_down()  
major_seventh_up()  
major_sixth_down()  
major_sixth_up()  
major_tenth_down()  
major_tenth_up()
```

```
major_third_down()
major_third_up()
major_thirteenth_down()
major_thirteenth_up()
minor_fifth_down()
minor_fifth_up()
minor_ninth_down()
minor_ninth_up()
minor_second_down()
minor_second_up()
minor_seventh_down()
minor_seventh_up()
minor_sixth_down()
minor_sixth_up()
minor_tenth_down()
minor_tenth_up()
minor_third_down()
minor_third_up()
minor_thirteenth_down()
minor_thirteenth_up()
notes = []
octave_down()
octave_up()
perfect_fifth_down()
perfect_fifth_up()
perfect_fourth_down()
perfect_fourth_up()
root = None
set_augment()
set_diminish()
set_transpose(amount)
transpose(amount)
transpose_list(lst)
walk(func)
```

```
class parker.notes.NoteGroup(notes=None)
    A mutable set of notes to be played at the same time.
```


`add (notes)`
`append (item)`

2.2 parker.chords

`class parker.chords.Chord (chord=None, octave=None)`
Source Material: [https://en.wikipedia.org/wiki/Chord_\(music\)](https://en.wikipedia.org/wiki/Chord_(music))

`all_transpositions ()`
Create all available named transpositions

`augment ()`

`augmented_eleventh_down ()`

`augmented_eleventh_up ()`

`augmented_ninth_down ()`

`augmented_ninth_up ()`

`clone ()`

`compound_augmented_fourth_down ()`

`compound_augmented_fourth_up ()`

`compound_augmented_second_down ()`

`compound_augmented_second_up ()`

`compound_major_second_down ()`

`compound_major_second_up ()`

`compound_major_sixth_down ()`

`compound_major_sixth_up ()`

`compound_major_third_down ()`

`compound_major_third_up ()`

`compound_minor_second_down ()`

`compound_minor_second_up ()`

`compound_minor_sixth_down ()`

`compound_minor_sixth_up ()`

`compound_minor_third_down ()`

`compound_minor_third_up ()`

`compound_perfect_fourth_down ()`

`compound_perfect_fourth_up ()`

`diminish ()`

`get_notes ()`

`get_octave_construction ()`

`get_scale ()`

```
highest_note()
lowest_note()
major_eleventh_down()
major_eleventh_up()
major_fifth_down()
major_fifth_up()
major_fourth_down()
major_fourth_up()
major_ninth_down()
major_ninth_up()
major_second_down()
major_second_up()
major_seventh_down()
major_seventh_up()
major_sixth_down()
major_sixth_up()
major_tenth_down()
major_tenth_up()
major_third_down()
major_third_up()
major_thirteenth_down()
major_thirteenth_up()
minor_fifth_down()
minor_fifth_up()
minor_ninth_down()
minor_ninth_up()
minor_second_down()
minor_second_up()
minor_seventh_down()
minor_seventh_up()
minor_sixth_down()
minor_sixth_up()
minor_tenth_down()
minor_tenth_up()
minor_third_down()
minor_third_up()
```

```

minor_thirteenth_down()
minor_thirteenth_up()
notes = []
octave_down()
octave_up()
perfect_fifth_down()
perfect_fifth_up()
perfect_fourth_down()
perfect_fourth_up()
root = None
set_augment()
set_diminish()
set_transpose(amount)
transpose(amount)
transpose_list(lst)
walk(func)

```

`chords.produce_all_chords(root)`
 Produce all chords given a root note.
 Returns a dictionary of chord names as the key and the chord object as the value.

2.3 parker.scales

```

class parker.scales.Scale(root, order=None)
    Source: https://en.wikipedia.org/wiki/Scale\_\(music\)
    ASCENDING = 'ascending'
    DESCENDING = 'descending'
    ORDER_CHOICES = ['ascending', 'descending']
    all_transpositions()
        Create all available named transpositions
    augment()
    augmented_eleventh_down()
    augmented_eleventh_up()
    augmented_ninth_down()
    augmented_ninth_up()
    build_scale()
    clone()
    compound_augmented_fourth_down()

```

```
compound_augmented_fourth_up()
compound_augmented_second_down()
compound_augmented_second_up()
compound_major_second_down()
compound_major_second_up()
compound_major_sixth_down()
compound_major_sixth_up()
compound_major_third_down()
compound_major_third_up()
compound_minor_second_down()
compound_minor_second_up()
compound_minor_sixth_down()
compound_minor_sixth_up()
compound_minor_third_down()
compound_minor_third_up()
compound_perfect_fourth_down()
compound_perfect_fourth_up()
diminish()
generic_notes = []
get_notes()
get_octave_construction()
get_tone_semitone_construction()
get_whole_half_construction()
highest_note()
intervals
is_generic_note_in_scale(note)
lowest_note()
major_eleventh_down()
major_eleventh_up()
major_fifth_down()
major_fifth_up()
major_fourth_down()
major_fourth_up()
major_ninth_down()
major_ninth_up()
major_second_down()
```

```
major_second_up()
major_seventh_down()
major_seventh_up()
major_sixth_down()
major_sixth_up()
major_tenth_down()
major_tenth_up()
major_third_down()
major_third_up()
major_thirteenth_down()
major_thirteenth_up()
minor_fifth_down()
minor_fifth_up()
minor_ninth_down()
minor_ninth_up()
minor_second_down()
minor_second_up()
minor_seventh_down()
minor_seventh_up()
minor_sixth_down()
minor_sixth_up()
minor_tenth_down()
minor_tenth_up()
minor_third_down()
minor_third_up()
minor_thirteenth_down()
minor_thirteenth_up()
notes = []
octave_down()
octave_up()
perfect_fifth_down()
perfect_fifth_up()
perfect_fourth_down()
perfect_fourth_up()
root = None
set_augment()
```

```
    set_diminish()  
    set_transpose(amount)  
    transpose(amount)  
    transpose_list(lst)  
    walk(func)
```

```
class parker.scales.Diatonic(root, order=None)
```

```
class parker.scales.Ionian(root, order=None)
```

```
    intervals
```

```
class parker.scales.Major(root, order=None)
```

```
class parker.scales.HarmonicMajor(root, order=None)
```

```
    intervals
```

```
class parker.scales.Dorian(root, order=None)
```

```
    intervals
```

```
class parker.scales.Phygian(root, order=None)
```

```
    intervals
```

```
class parker.scales.MedievalLydian(root, order=None)
```

```
    intervals
```

```
class parker.scales.Lydian(root, order=None)
```

The Lydian scale can be described as a major scale with the fourth scale degree raised a semitone, e.g., a C-major scale with an F# rather than F-natural.

Source: https://en.wikipedia.org/wiki/Lydian_mode

```
    intervals
```

```
class parker.scales.Mixolydian(root, order=None)
```

```
    intervals
```

```
class parker.scales.Dominant(root, order=None)
```

```
class parker.scales.Aeolian(root, order=None)
```

```
    intervals
```

```
class parker.scales.Minor(root, order=None)
```

```
class parker.scales.HarmonicMinor(root, order=None)
```

```
    intervals
```

```
class parker.scales.Locrian(root, order=None)
```

intervals

```
class parker.scales.SuperLocrian (root, order=None)
```

intervals

```
class parker.scales.MajorPentatonic (root, order=None)
```

Major Pentatonic drops 4th and 7th from the Diatonic Major and consists of only 5 notes.

intervals

```
class parker.scales.MinorPentatonic (root, order=None)
```

Minor Pentatonic drops 2nd and 6th from the Diatonic Minor and consists of only 5 notes.

intervals

```
class parker.scales.MajorBlues (root, order=None)
```

Major Blues is the same as the Major Pentatonic but it adds a diminished 4th and consists of 6 notes.

intervals

```
class parker.scales.MinorBlues (root, order=None)
```

Minor Blues is the same as the Minor Pentatonic but it adds an augmented 5th and consists of 6 notes.

intervals

```
class parker.scales.Altered (root, order=None)
```

intervals

```
class parker.scales.DiminishedWholeTone (root, order=None)
```

```
class parker.scales.Chromatic (root, order=None)
```

intervals

```
class parker.scales.Octatonic (root, order=None)
```

```
class parker.scales.OctatonicModeOne (root, order=None)
```

intervals

```
class parker.scales.OctatonicModeTwo (root, order=None)
```

intervals

```
scales.circle_of_fifths ()
```

```
scales.circle_of_fourths ()
```

```
scales._scale_creator (scale_cls, root='C4')
```

```
scales.dorian_scales (root='C4')
```

```
scales.mixolydian_scales (root='C4')
```

```
scales.major_scales (root='C4')
```

```
scales.minor_scales (root='C4')
```

```
scales.major_pentatonic_scales (root='C4')
```

```
scales.minor_pentatonic_scales (root='C4')
```

```
scales.major_blues_scales (root='C4')
```

```
scales.minor_blues_scales (root='C4')
```

2.4 parker.keys

```
class parker.keys.Key (key='C')
```

Reference: [https://en.wikipedia.org/wiki/Key_\(music\)](https://en.wikipedia.org/wiki/Key_(music))

```
MAJOR = 'major'
```

```
MINOR = 'minor'
```

```
get_accidental_notes ()
```

```
get_scale ()
```

```
is_major ()
```

```
is_minor ()
```

2.5 parker.progressions

```
class parker.progressions.Progression (root, scale_cls=<class 'parker.scales.Major'>)
```

Reference: https://en.wikipedia.org/wiki/Roman_numeral_analysis

```
all_progressions ()
```

```
from_list (prog_list, as_map=False)
```

Take a list of progressions and return the list of chords.

Setting *as_map* to True will return the progressions as a dictionary where the keys are the progressions passed in and the values are the chords.

```
from_string (progression)
```

Take a string representation of a progression and return the chord that it represents.

```
standard_sevenths ()
```

```
standard_triads ()
```

Indices and tables

- `genindex`
- `modindex`
- `search`

Symbols

`_scale_creator()` (parker.scales method), 19

A

Accidental (class in parker.notes), 7

accidentals (parker.notes.Note attribute), 7

add() (parker.notes.NoteGroup method), 12

Aeolian (class in parker.scales), 18

all_progressions() (parker.progressions.Progression method), 20

all_transpositions() (parker.chords.Chord method), 13

all_transpositions() (parker.notes.Note method), 7

all_transpositions() (parker.notes.NoteGroupBase method), 10

all_transpositions() (parker.scales.Scale method), 15

alter (parker.notes.Accidental attribute), 7

Altered (class in parker.scales), 19

append() (parker.notes.NoteGroup method), 13

ASCENDING (parker.scales.Scale attribute), 15

augment() (parker.chords.Chord method), 13

augment() (parker.notes.Note method), 7

augment() (parker.notes.NoteGroupBase method), 10

augment() (parker.scales.Scale method), 15

augmented_eleventh_down() (parker.chords.Chord method), 13

augmented_eleventh_down() (parker.notes.Note method), 7

augmented_eleventh_down() (parker.notes.NoteGroupBase method), 10

augmented_eleventh_down() (parker.scales.Scale method), 15

augmented_eleventh_up() (parker.chords.Chord method), 13

augmented_eleventh_up() (parker.notes.Note method), 7

augmented_eleventh_up() (parker.notes.NoteGroupBase method), 10

augmented_eleventh_up() (parker.scales.Scale method), 15

augmented_ninth_down() (parker.chords.Chord method), 13

augmented_ninth_down() (parker.notes.Note method), 7

augmented_ninth_down() (parker.notes.NoteGroupBase method), 10

augmented_ninth_down() (parker.scales.Scale method), 15

augmented_ninth_up() (parker.chords.Chord method), 13

augmented_ninth_up() (parker.notes.Note method), 8

augmented_ninth_up() (parker.notes.NoteGroupBase method), 10

augmented_ninth_up() (parker.scales.Scale method), 15

B

base_name (parker.notes.Note attribute), 8

build_scale() (parker.scales.Scale method), 15

C

Chord (class in parker.chords), 13

Chromatic (class in parker.scales), 19

circle_of_fifths() (parker.scales method), 19

circle_of_fourths() (parker.scales method), 19

clone() (parker.chords.Chord method), 13

clone() (parker.notes.Note method), 8

clone() (parker.notes.NoteGroupBase method), 10

clone() (parker.scales.Scale method), 15

compound_augmented_fourth_down() (parker.chords.Chord method), 13

compound_augmented_fourth_down() (parker.notes.Note method), 8

compound_augmented_fourth_down() (parker.notes.NoteGroupBase method), 10

compound_augmented_fourth_down() (parker.scales.Scale method), 15

compound_augmented_fourth_up() (parker.chords.Chord method), 13

compound_augmented_fourth_up() (parker.notes.Note method), 8

compound_augmented_fourth_up() (parker.notes.NoteGroupBase method), 10

compound_augmented_fourth_up() (parker.scales.Scale method), 15

compound_augmented_second_down() (parker.chords.Chord method), 13	compound_major_third_down() (parker.scales.Scale method), 16
compound_augmented_second_down() (parker.notes.Note method), 8	compound_major_third_up() (parker.chords.Chord method), 13
compound_augmented_second_down() (parker.notes.NoteGroupBase method), 10	compound_major_third_up() (parker.notes.Note method), 8
compound_augmented_second_down() (parker.scales.Scale method), 16	compound_major_third_up() (parker.notes.NoteGroupBase method), 11
compound_augmented_second_up() (parker.chords.Chord method), 13	compound_major_third_up() (parker.scales.Scale method), 16
compound_augmented_second_up() (parker.notes.Note method), 8	compound_minor_second_down() (parker.chords.Chord method), 13
compound_augmented_second_up() (parker.notes.NoteGroupBase method), 11	compound_minor_second_down() (parker.notes.Note method), 8
compound_augmented_second_up() (parker.scales.Scale method), 16	compound_minor_second_down() (parker.notes.NoteGroupBase method), 11
compound_major_second_down() (parker.chords.Chord method), 13	compound_minor_second_down() (parker.scales.Scale method), 16
compound_major_second_down() (parker.notes.Note method), 8	compound_minor_second_up() (parker.chords.Chord method), 13
compound_major_second_down() (parker.notes.NoteGroupBase method), 11	compound_minor_second_up() (parker.notes.Note method), 8
compound_major_second_down() (parker.scales.Scale method), 16	compound_minor_second_up() (parker.notes.NoteGroupBase method), 11
compound_major_second_up() (parker.chords.Chord method), 13	compound_minor_second_up() (parker.scales.Scale method), 16
compound_major_second_up() (parker.notes.Note method), 8	compound_minor_sixth_down() (parker.chords.Chord method), 13
compound_major_second_up() (parker.notes.NoteGroupBase method), 11	compound_minor_sixth_down() (parker.notes.Note method), 8
compound_major_second_up() (parker.scales.Scale method), 16	compound_minor_sixth_down() (parker.notes.NoteGroupBase method), 11
compound_major_sixth_down() (parker.chords.Chord method), 13	compound_minor_sixth_down() (parker.scales.Scale method), 16
compound_major_sixth_down() (parker.notes.Note method), 8	compound_minor_sixth_up() (parker.chords.Chord method), 13
compound_major_sixth_down() (parker.notes.NoteGroupBase method), 11	compound_minor_sixth_up() (parker.notes.Note method), 8
compound_major_sixth_down() (parker.scales.Scale method), 16	compound_minor_sixth_up() (parker.notes.NoteGroupBase method), 11
compound_major_sixth_up() (parker.chords.Chord method), 13	compound_minor_sixth_up() (parker.scales.Scale method), 16
compound_major_sixth_up() (parker.notes.Note method), 8	compound_minor_third_down() (parker.chords.Chord method), 13
compound_major_sixth_up() (parker.notes.NoteGroupBase method), 11	compound_minor_third_down() (parker.notes.Note method), 8
compound_major_sixth_up() (parker.scales.Scale method), 16	compound_minor_third_down() (parker.notes.NoteGroupBase method), 11
compound_major_third_down() (parker.chords.Chord method), 13	compound_minor_third_down() (parker.scales.Scale method), 16
compound_major_third_down() (parker.notes.Note method), 8	compound_minor_third_up() (parker.chords.Chord method), 13
compound_major_third_down() (parker.notes.NoteGroupBase method), 11	compound_minor_third_up() (parker.notes.Note method), 8

compound_minor_third_up() (parker.notes.NoteGroupBase method), 11
 compound_minor_third_up() (parker.scales.Scale method), 16
 compound_perfect_fourth_down() (parker.chords.Chord method), 13
 compound_perfect_fourth_down() (parker.notes.Note method), 8
 compound_perfect_fourth_down() (parker.notes.NoteGroupBase method), 11
 compound_perfect_fourth_down() (parker.scales.Scale method), 16
 compound_perfect_fourth_up() (parker.chords.Chord method), 13
 compound_perfect_fourth_up() (parker.notes.Note method), 8
 compound_perfect_fourth_up() (parker.notes.NoteGroupBase method), 11
 compound_perfect_fourth_up() (parker.scales.Scale method), 16

D

DESCENDING (parker.scales.Scale attribute), 15
 Diatonic (class in parker.scales), 18
 diminish() (parker.chords.Chord method), 13
 diminish() (parker.notes.Note method), 8
 diminish() (parker.notes.NoteGroupBase method), 11
 diminish() (parker.scales.Scale method), 16
 DiminishedWholeTone (class in parker.scales), 19
 Dominant (class in parker.scales), 18
 Dorian (class in parker.scales), 18
 dorian_scales() (parker.scales method), 19

F

from_list() (parker.progressions.Progression method), 20
 from_string() (parker.progressions.Progression method), 20

G

generalize() (parker.notes.Note method), 8
 generic_notes (parker.scales.Scale attribute), 16
 get_accidental_notes() (parker.keys.Key method), 20
 get_frequency() (parker.notes.Note method), 8
 get_notes() (parker.chords.Chord method), 13
 get_notes() (parker.notes.NoteGroupBase method), 11
 get_notes() (parker.scales.Scale method), 16
 get_octave_construction() (parker.chords.Chord method), 13
 get_octave_construction() (parker.scales.Scale method), 16
 get_scale() (parker.chords.Chord method), 13
 get_scale() (parker.keys.Key method), 20
 get_tone_semitone_construction() (parker.scales.Scale method), 16

get_whole_half_construction() (parker.scales.Scale method), 16

H

HarmonicMajor (class in parker.scales), 18
 HarmonicMinor (class in parker.scales), 18
 highest_note() (parker.chords.Chord method), 13
 highest_note() (parker.notes.NoteGroupBase method), 11
 highest_note() (parker.scales.Scale method), 16

I

intervals (parker.scales.Aeolian attribute), 18
 intervals (parker.scales.Altered attribute), 19
 intervals (parker.scales.Chromatic attribute), 19
 intervals (parker.scales.Dorian attribute), 18
 intervals (parker.scales.HarmonicMajor attribute), 18
 intervals (parker.scales.HarmonicMinor attribute), 18
 intervals (parker.scales.Ionian attribute), 18
 intervals (parker.scales.Locrian attribute), 18
 intervals (parker.scales.Lydian attribute), 18
 intervals (parker.scales.MajorBlues attribute), 19
 intervals (parker.scales.MajorPentatonic attribute), 19
 intervals (parker.scales.MedievalLydian attribute), 18
 intervals (parker.scales.MinorBlues attribute), 19
 intervals (parker.scales.MinorPentatonic attribute), 19
 intervals (parker.scales.Mixolydian attribute), 18
 intervals (parker.scales.OctatonicModeOne attribute), 19
 intervals (parker.scales.OctatonicModeTwo attribute), 19
 intervals (parker.scales.Phyrgian attribute), 18
 intervals (parker.scales.Scale attribute), 16
 intervals (parker.scales.SuperLocrian attribute), 19
 Ionian (class in parker.scales), 18
 is_generic_note_in_scale() (parker.scales.Scale method), 16

is_major() (parker.keys.Key method), 20
 is_minor() (parker.keys.Key method), 20
 is_valid_note() (parker.notes method), 7

K

Key (class in parker.keys), 20

L

Locrian (class in parker.scales), 18
 lowest_note() (parker.chords.Chord method), 14
 lowest_note() (parker.notes.NoteGroupBase method), 11
 lowest_note() (parker.scales.Scale method), 16
 Lydian (class in parker.scales), 18

M

Major (class in parker.scales), 18
 MAJOR (parker.keys.Key attribute), 20
 major_blues_scales() (parker.scales method), 19
 major_eleventh_down() (parker.chords.Chord method), 14

major_eleventh_down() (parker.notes.Note method), 8
 major_eleventh_down() (parker.notes.NoteGroupBase method), 11
 major_eleventh_down() (parker.scales.Scale method), 16
 major_eleventh_up() (parker.chords.Chord method), 14
 major_eleventh_up() (parker.notes.Note method), 8
 major_eleventh_up() (parker.notes.NoteGroupBase method), 11
 major_eleventh_up() (parker.scales.Scale method), 16
 major_fifth_down() (parker.chords.Chord method), 14
 major_fifth_down() (parker.notes.Note method), 9
 major_fifth_down() (parker.notes.NoteGroupBase method), 11
 major_fifth_down() (parker.scales.Scale method), 16
 major_fifth_up() (parker.chords.Chord method), 14
 major_fifth_up() (parker.notes.Note method), 9
 major_fifth_up() (parker.notes.NoteGroupBase method), 11
 major_fifth_up() (parker.scales.Scale method), 16
 major_fourth_down() (parker.chords.Chord method), 14
 major_fourth_down() (parker.notes.Note method), 9
 major_fourth_down() (parker.notes.NoteGroupBase method), 11
 major_fourth_down() (parker.scales.Scale method), 16
 major_fourth_up() (parker.chords.Chord method), 14
 major_fourth_up() (parker.notes.Note method), 9
 major_fourth_up() (parker.notes.NoteGroupBase method), 11
 major_fourth_up() (parker.scales.Scale method), 16
 major_ninth_down() (parker.chords.Chord method), 14
 major_ninth_down() (parker.notes.Note method), 9
 major_ninth_down() (parker.notes.NoteGroupBase method), 11
 major_ninth_down() (parker.scales.Scale method), 16
 major_ninth_up() (parker.chords.Chord method), 14
 major_ninth_up() (parker.notes.Note method), 9
 major_ninth_up() (parker.notes.NoteGroupBase method), 11
 major_ninth_up() (parker.scales.Scale method), 16
 major_pentatonic_scales() (parker.scales method), 19
 major_scales() (parker.scales method), 19
 major_second_down() (parker.chords.Chord method), 14
 major_second_down() (parker.notes.Note method), 9
 major_second_down() (parker.notes.NoteGroupBase method), 11
 major_second_down() (parker.scales.Scale method), 16
 major_second_up() (parker.chords.Chord method), 14
 major_second_up() (parker.notes.Note method), 9
 major_second_up() (parker.notes.NoteGroupBase method), 11
 major_second_up() (parker.scales.Scale method), 16
 major_seventh_down() (parker.chords.Chord method), 14
 major_seventh_down() (parker.notes.Note method), 9
 major_seventh_down() (parker.notes.NoteGroupBase method), 11
 major_seventh_down() (parker.scales.Scale method), 17
 major_seventh_up() (parker.chords.Chord method), 14
 major_seventh_up() (parker.notes.Note method), 9
 major_seventh_up() (parker.notes.NoteGroupBase method), 11
 major_seventh_up() (parker.scales.Scale method), 17
 major_sixth_down() (parker.chords.Chord method), 14
 major_sixth_down() (parker.notes.Note method), 9
 major_sixth_down() (parker.notes.NoteGroupBase method), 11
 major_sixth_down() (parker.scales.Scale method), 17
 major_sixth_up() (parker.chords.Chord method), 14
 major_sixth_up() (parker.notes.Note method), 9
 major_sixth_up() (parker.notes.NoteGroupBase method), 11
 major_sixth_up() (parker.scales.Scale method), 17
 major_tenth_down() (parker.chords.Chord method), 14
 major_tenth_down() (parker.notes.Note method), 9
 major_tenth_down() (parker.notes.NoteGroupBase method), 11
 major_tenth_down() (parker.scales.Scale method), 17
 major_tenth_up() (parker.chords.Chord method), 14
 major_tenth_up() (parker.notes.Note method), 9
 major_tenth_up() (parker.notes.NoteGroupBase method), 11
 major_tenth_up() (parker.scales.Scale method), 17
 major_third_down() (parker.chords.Chord method), 14
 major_third_down() (parker.notes.Note method), 9
 major_third_down() (parker.notes.NoteGroupBase method), 11
 major_third_down() (parker.scales.Scale method), 17
 major_third_up() (parker.chords.Chord method), 14
 major_third_up() (parker.notes.Note method), 9
 major_third_up() (parker.notes.NoteGroupBase method), 12
 major_third_up() (parker.scales.Scale method), 17
 major_thirteenth_down() (parker.chords.Chord method), 14
 major_thirteenth_down() (parker.notes.Note method), 9
 major_thirteenth_down() (parker.notes.NoteGroupBase method), 12
 major_thirteenth_down() (parker.scales.Scale method), 17
 major_thirteenth_up() (parker.chords.Chord method), 14
 major_thirteenth_up() (parker.notes.Note method), 9
 major_thirteenth_up() (parker.notes.NoteGroupBase method), 12
 major_thirteenth_up() (parker.scales.Scale method), 17
 MajorBlues (class in parker.scales), 19
 MajorPentatonic (class in parker.scales), 19
 MedieallLydian (class in parker.scales), 18
 Minor (class in parker.scales), 18

MINOR (parker.keys.Key attribute), 20
 minor_blues_scales() (parker.scales method), 20
 minor_fifth_down() (parker.chords.Chord method), 14
 minor_fifth_down() (parker.notes.Note method), 9
 minor_fifth_down() (parker.notes.NoteGroupBase method), 12
 minor_fifth_down() (parker.scales.Scale method), 17
 minor_fifth_up() (parker.chords.Chord method), 14
 minor_fifth_up() (parker.notes.Note method), 9
 minor_fifth_up() (parker.notes.NoteGroupBase method), 12
 minor_fifth_up() (parker.scales.Scale method), 17
 minor_ninth_down() (parker.chords.Chord method), 14
 minor_ninth_down() (parker.notes.Note method), 9
 minor_ninth_down() (parker.notes.NoteGroupBase method), 12
 minor_ninth_down() (parker.scales.Scale method), 17
 minor_ninth_up() (parker.chords.Chord method), 14
 minor_ninth_up() (parker.notes.Note method), 9
 minor_ninth_up() (parker.notes.NoteGroupBase method), 12
 minor_ninth_up() (parker.scales.Scale method), 17
 minor_pentatonic_scales() (parker.scales method), 19
 minor_scales() (parker.scales method), 19
 minor_second_down() (parker.chords.Chord method), 14
 minor_second_down() (parker.notes.Note method), 9
 minor_second_down() (parker.notes.NoteGroupBase method), 12
 minor_second_down() (parker.scales.Scale method), 17
 minor_second_up() (parker.chords.Chord method), 14
 minor_second_up() (parker.notes.Note method), 9
 minor_second_up() (parker.notes.NoteGroupBase method), 12
 minor_second_up() (parker.scales.Scale method), 17
 minor_seventh_down() (parker.chords.Chord method), 14
 minor_seventh_down() (parker.notes.Note method), 9
 minor_seventh_down() (parker.notes.NoteGroupBase method), 12
 minor_seventh_down() (parker.scales.Scale method), 17
 minor_seventh_up() (parker.chords.Chord method), 14
 minor_seventh_up() (parker.notes.Note method), 9
 minor_seventh_up() (parker.notes.NoteGroupBase method), 12
 minor_seventh_up() (parker.scales.Scale method), 17
 minor_sixth_down() (parker.chords.Chord method), 14
 minor_sixth_down() (parker.notes.Note method), 9
 minor_sixth_down() (parker.notes.NoteGroupBase method), 12
 minor_sixth_down() (parker.scales.Scale method), 17
 minor_sixth_up() (parker.chords.Chord method), 14
 minor_sixth_up() (parker.notes.Note method), 9
 minor_sixth_up() (parker.notes.NoteGroupBase method), 12
 minor_sixth_up() (parker.scales.Scale method), 17
 minor_tenth_down() (parker.chords.Chord method), 14
 minor_tenth_down() (parker.notes.Note method), 9
 minor_tenth_down() (parker.notes.NoteGroupBase method), 12
 minor_tenth_down() (parker.scales.Scale method), 17
 minor_tenth_up() (parker.chords.Chord method), 14
 minor_tenth_up() (parker.notes.Note method), 9
 minor_tenth_up() (parker.notes.NoteGroupBase method), 12
 minor_tenth_up() (parker.scales.Scale method), 17
 minor_third_down() (parker.chords.Chord method), 14
 minor_third_down() (parker.notes.Note method), 9
 minor_third_down() (parker.notes.NoteGroupBase method), 12
 minor_third_down() (parker.scales.Scale method), 17
 minor_third_up() (parker.chords.Chord method), 14
 minor_third_up() (parker.notes.Note method), 9
 minor_third_up() (parker.notes.NoteGroupBase method), 12
 minor_third_up() (parker.scales.Scale method), 17
 minor_thirteenth_down() (parker.chords.Chord method), 14
 minor_thirteenth_down() (parker.notes.Note method), 9
 minor_thirteenth_down() (parker.notes.NoteGroupBase method), 12
 minor_thirteenth_down() (parker.scales.Scale method), 17
 minor_thirteenth_up() (parker.chords.Chord method), 15
 minor_thirteenth_up() (parker.notes.Note method), 9
 minor_thirteenth_up() (parker.notes.NoteGroupBase method), 12
 minor_thirteenth_up() (parker.scales.Scale method), 17
 MinorBlues (class in parker.scales), 19
 MinorPentatonic (class in parker.scales), 19
 Mixolydian (class in parker.scales), 18
 mixolydian_scales() (parker.scales method), 19

N

name (parker.notes.Accidental attribute), 7
 normalize() (parker.notes.Note method), 9
 Note (class in parker.notes), 7
 note_from_frequency() (parker.notes method), 7
 NoteGroup (class in parker.notes), 12
 NoteGroupBase (class in parker.notes), 10
 notes (parker.chords.Chord attribute), 15
 notes (parker.notes.NoteGroupBase attribute), 12
 notes (parker.scales.Scale attribute), 17
 NotesParser (class in parker.notes), 10

O

Octatonic (class in parker.scales), 19
 OctatonicModeOne (class in parker.scales), 19
 OctatonicModeTwo (class in parker.scales), 19
 octave (parker.notes.Note attribute), 10

octave_down() (parker.chords.Chord method), 15
octave_down() (parker.notes.Note method), 10
octave_down() (parker.notes.NoteGroupBase method), 12
octave_down() (parker.scales.Scale method), 17
octave_up() (parker.chords.Chord method), 15
octave_up() (parker.notes.Note method), 10
octave_up() (parker.notes.NoteGroupBase method), 12
octave_up() (parker.scales.Scale method), 17
ORDER_CHOICES (parker.scales.Scale attribute), 15

P

parse() (parker.notes.NotesParser static method), 10
perfect_fifth_down() (parker.chords.Chord method), 15
perfect_fifth_down() (parker.notes.Note method), 10
perfect_fifth_down() (parker.notes.NoteGroupBase method), 12
perfect_fifth_down() (parker.scales.Scale method), 17
perfect_fifth_up() (parker.chords.Chord method), 15
perfect_fifth_up() (parker.notes.Note method), 10
perfect_fifth_up() (parker.notes.NoteGroupBase method), 12
perfect_fifth_up() (parker.scales.Scale method), 17
perfect_fourth_down() (parker.chords.Chord method), 15
perfect_fourth_down() (parker.notes.Note method), 10
perfect_fourth_down() (parker.notes.NoteGroupBase method), 12
perfect_fourth_down() (parker.scales.Scale method), 17
perfect_fourth_up() (parker.chords.Chord method), 15
perfect_fourth_up() (parker.notes.Note method), 10
perfect_fourth_up() (parker.notes.NoteGroupBase method), 12
perfect_fourth_up() (parker.scales.Scale method), 17
Phrygian (class in parker.scales), 18
produce_all_chords() (parker.chords method), 15
Progression (class in parker.progressions), 20

R

root (parker.chords.Chord attribute), 15
root (parker.notes.NoteGroupBase attribute), 12
root (parker.scales.Scale attribute), 17

S

Scale (class in parker.scales), 15
set_augment() (parker.chords.Chord method), 15
set_augment() (parker.notes.Accidental method), 7
set_augment() (parker.notes.Note method), 10
set_augment() (parker.notes.NoteGroupBase method), 12
set_augment() (parker.scales.Scale method), 17
set_diminish() (parker.chords.Chord method), 15
set_diminish() (parker.notes.Accidental method), 7
set_diminish() (parker.notes.Note method), 10
set_diminish() (parker.notes.NoteGroupBase method), 12
set_diminish() (parker.scales.Scale method), 17
set_from_num() (parker.notes.Accidental method), 7

set_from_str() (parker.notes.Accidental method), 7
set_transpose() (parker.chords.Chord method), 15
set_transpose() (parker.notes.Note method), 10
set_transpose() (parker.notes.NoteGroupBase method), 12
set_transpose() (parker.scales.Scale method), 18
standard_sevenths() (parker.progressions.Progression method), 20
standard_triads() (parker.progressions.Progression method), 20
SuperLocrian (class in parker.scales), 19

T

transpose() (parker.chords.Chord method), 15
transpose() (parker.notes.Note method), 10
transpose() (parker.notes.NoteGroupBase method), 12
transpose() (parker.scales.Scale method), 18
transpose_list() (parker.chords.Chord method), 15
transpose_list() (parker.notes.Note method), 10
transpose_list() (parker.notes.NoteGroupBase method), 12
transpose_list() (parker.scales.Scale method), 18

W

walk() (parker.chords.Chord method), 15
walk() (parker.notes.NoteGroupBase method), 12
walk() (parker.scales.Scale method), 18