Abjad Quick Reference

2017

Duration primitives

Pairs:	`(3, 8)`
Durations:	`abjad.Duration(3, 8)`
Offsets:	`abjad.Offset(1, 2)`
Multipliers:	`abjad.Multiplier(5, 7)`

Pitch primitives

Integers and "floats":	`3`
Strings: [1]	`"cs''"` *or* `"dqs''"` *or* `"C\#5"`
Named pitches: [2]	`abjad.NamedPitch("d'")`

[1] Lower-case pitch strings use LilyPond's syntax, while upper-case pitch strings — e.g `C#5`, `Db3` — use American pitch-class syntax.

[2] Named pitch objects can be instantiated from integers or floats representing numbered pitches, from any of the string variations, or from other pitch objects.

Leaves

Notes: [3]	`abjad.Note("c'4")`
Rests:	`abjad.Rest((1, 4))`
Chords:	`abjad.Chord((0, 1, 4), Duration(1, 4))`
Skips:	`abjad.Skip(Duration(1, 4))`
Multimeasure rests:	`abjad.MultimeasureRest(1)`

[3] All leaves can be instantiated from LilyPond strings, from valid duration and pitch inputs as described above, or from other leaf instances.

Non-context Containers

"Bare" containers: [4]	`abjad.Container("c'4 d'4 e'4 f'4")`
Measures:	`abjad.Measure((3, 4), [Note("c2.")])`
Tuplets:	`abjad.Tuplet((2, 3), "c'8 d'8 e'8")`

[4] All containers can be instantiated from LilyPond strings, or from lists of Abjad components.

Contexts

Voices:	`abjad.Voice("c'4 d'4 e'4 f'4")`
Staves:	`abjad.Staff("c'4 d'4 e'4 f'4")`
Staff groups:	`abjad.StaffGroup([staff_one, staff_two])`
Scores:	`abjad.Score([staff_group])`
Generic contexts:	`abjad.Context(..., context_name='FooStaff')`

Non-scoped indicators

Articulations: `abjad.Articulation('staccato')`

Markup: `abjad.Markup("honk car horn now")`

Indicators

Clefs: `abjad.Clef('bass')`

Dynamics: `abjad.Dynamic('f')`

Key signatures: `abjad.KeySignature('c', 'minor')`

Metronome marks: `abjad.MetronomeMark((1, 4), 60)`

Time signatures: `abjad.TimeSignature((6, 8))`

Spanners

Beams: `abjad.Beam()`

Slurs: `abjad.Slur()`

Glissandi: `abjad.Glissando()`

Hairpins: `abjad.Hairpin('p < f')`

Simple verbs

Formatting: `format(object)`

Illustrating: `abjad.show(component)`

Playing: `abjad.play(component)`

Attaching: [5] `abjad.attach(indicator, component)`

Templating: `abjad.new(object, key1=value1, key2=value2)`

[5] Use `attach()` to attach an indicator to a single leaf, or a spanner to one or more leaves. Use `detach()` to remove an indicator from a leaf. Some spanners have specific requirements about how many and what kinds of leaves they can attach to.

Complex verbs

Selecting: `abjad.select(component).by_<query>()`

Inspecting: `abjad.inspect(component).get_<property>()`

Typesetting: `abjad.override(...).<grob_name>.<property> = ...`

Mutating: `abjad.mutate(component).<do_something>()`