SINBAD - Quick Reference - PYTHON

http://cs.berry.edu/sinbad

Basic Template

```
from sinbad import *

ds = Data_Source.connect("<URL>")
# ... additional settings - see params, options below
ds.load()
x = ds.fetch(...)
```

Examining Available Data

```
ds.print_description()
```

Test if field paths valid:

```
ds.has_fields(".../...", ...)
```

List of available field names:

```
ds.field_list()    # OR   ds.field_list( ".../..." )
```

Number of data records (in a list) available:

```
ds.data_length()   # OR   ds.data_length(“.../...")
```

Other Connection Methods

Specify a data format ("CSV", "XML", "JSON"):

```
ds = Data_Source.connect_as("xml", "<URL>")
```

Connect using a data specification file:

```
ds = Data_Source.connect_using("<URL/Path>")
```

Use a GUI dialog to select local file:

```
ds = Data_Source.connect_gui()
# OR  ds = Data_Source.connect_gui_as(“xml")
```

Connection (URL) Parameters

Some data sources may require additional parameters to construct the URL. After the `connect` and before `load`:

```
ds.set_param("<name>", "<value>")
```

Data Format Options

Some data sources provide post-processing options to manipulate the data once it has been downloaded. The available options are format-specific and are listed in the *print_description()* information.

```
ds.set_option("<name>", "<value>")
```

For example (with a CSV data source):

```
ds.set_option("header", "ID,Name,Call
sign,Country,Active”)
```

Selecting from .zip archive

```
ds.set_option("file-entry", "FACTDATA_MAR2016.TXT")
```

Sampling Data

```
ds.load_sample(<amt>)
    # or
ds.load_sample(<amt>, <seed>)
```

Sampled data is cached and reloaded from cache if the same code is run again. To force a fresh sample to be generated, use **`ds.load_fresh_sample(...)`** instead of `load_sample`.

Cache Control

Control frequency of caching (or disable it):

```
ds.set_cache_timeout(300)
# OR ds.set_cache_timeout(NEVER-RELOAD)  -- always use cache
# OR ds.set_cache_timeout(NEVER-CACHE)
```

Show where files are cached:

```
print(ds.cache_directory())
```

Clear all cache files (for all data sources):

```
ds.clear_cache()
```

View Preferences

```
Data_Source.preferences()
```

When preferences are saved, the program will immediately terminate and exit. Comment out or delete the statement above to enable the program to continue running as usual.

Fetching Data

```
### GENERAL PURPOSE -----

ds.fetch()
     # fetches ALL available data (lists + dictionaries)

ds.fetch("path/to/field1", ...)
     # fetches (lists of, if appropriate) data
ds.fetch("path/to/field1", ..., base_path = "loans")
     # using optional base_path clause

### RANDOM -----

ds.fetch_random(...)
    # same patterns as for ds.fetch(...) above
    # note: always returns the same result until .load()
            called again

### POSITIONAL -----
     # same patterns as for ds.fetch(...) above
ds.fetch_first(...)
ds.fetch_second(...)
ds.fetch_third(...)
ds.fetch_ith(i, ...)     # i >= 0

### TYPE CONVERTING —

ds.fetch_int("path/to/field")
ds.fetch_first_int("path/to/field")
ds.fetch_ith_int(i, "path/to/field")   # i >= 0
ds.fetch_random_int("path/to/field")

ds.fetch_int("path/to/field")
ds.fetch_first_int("path/to/field")
ds.fetch_ith_int(i, "path/to/field")   # i >= 0
ds.fetch_random_int("path/to/field")
```