micvbang Documentation

Michael Bang

Jul 23, 2018

Contents

`micvbang` is a collection of stuff often used by micvbang.

CHAPTER 1

Modules

1.1 io

micvbang.io.**here**(**ps*)
Return script execution path os.path.join'ed with the given arguments.

micvbang.io.**list_dir**(*path='.'*, *dirs=False*, *files=True*, *ext=None*, *recursive=False*)
List the contents of a directory. If recursive is set to True, list_dir will recurse through sub-dirs as well.

Returns directories if dirs is set to True. Returns files if files is set True. Returns only files with the given extension if ext is set. Note that the extension includes the dot, i.e. '.jpg'.

micvbang.io.**open**(*path*, *mode='r'*)
Open a file and return a stream.

If the file has .gz extension, *gzip.open* is used in place of *open*.

1.2 progress

class micvbang.progress.**ProgressTracker**(*it*, *get_id=None*, *f=None*, *flush_freq=0*, *print_skips_freq=0*)
Track and continue progress over iterators, saving state in a file-like object.

Arguments: it(iterable): Iterable used to to make the iterator to track the progress of. get_id(function): Function mapping values generated from :param it: to unique ids of type string.

Note:

param get_id must return a string value that does **not** contain newlines.

---

close()
Close *ProgressTracker*. This ensures that no iterators created from the instance will progress any further.

iter()
Return an iterator that iterates over the given input iterator and automatically tracks its progress. *processed()* will be called **before** each value is returned to the user.

Note: Potential off-by-one error here; all ids are marked as *processed* **before** they are returned and will therefore never be returned again. If the program crashes and the id was not in fact processed by user code, it will go unprocessed.

---

iter_ids()
Return an iterator that yields an (id, data)-tuple. In order to mark an iteration as processed, *processed()* must be called with the given id.

processed(*id*)
Mark an id as processed.

This means that values with the given id will **not** be returned when creating iterators using the same progress file.

class micvbang.progress.**ReadAppendFile**(*open_read*, *open_append*)
Create new instance of ReadAppendFile(open_read, open_append)

open_append
Alias for field number 1

open_read
Alias for field number 0

1.3 jsonutil

1.4 dictutil

micvbang.dictutil.**get_deep**(*obj*, *path*, *separator='.'*)
Retrieve the value denoted by path in obj.

This is done by recursively calling obj.get([path_head], None) until the path is traversed, until a value on the path does not have a callable *get* method, or until the next step in the path does not exist. In case the full path cannot be traversed, None is returned.

Python Module Index

m

Index