BashMatic® Auto-Generated Function Index

Index

array.has-element()

Returns "true" if the first argument is a member of the array passed as the second argument:

Example

```
$ declare -a array=("a string" test2000 moo)
if [[ $(array.has-element "a string" "${array[@]}") == "true" ]]; then
  ...
fi
```

array.includes()

Similar to array.has-elements, but does not print anything, just returns 0 if includes, 1 if not.

array.join()

Joins a given array with a custom character

Example

```
$ declare -a array=(one two three)
$ array.join "," "${array[@]}"
one,two,three
```

array.sort()

Sorts the array alphanumerically and prints it to STDOUT

Example

```
declare -a unsorted=(hello begin again again)
local sorted="$(array.sort "${unsorted[@]}")"
```

array.sort-numeric()

Sorts the array numerically and prints it to STDOUT

Example

```
declare -a unsorted=(1 2 34 45 6)
local sorted="$(array.sort-numeric "${unsorted[@]}")"
```

array.min()

Returns a minimum integer from an array. Non-numeric elements are ignored and skipped over. Negative numbers are supported, but non-integers are not.

Example

```
$ declare -a array=(10 20 30 -5 5)
$ array.min "," "${array[@]}"
-5
```

array.max()

Returns a maximum integer from an array. Non-numeric elements are ignored and skipped over. Negative numbers are supported, but non-integers are not.

Example

```
$ declare -a array=(10 20 30 -5 5)
$ array.min "," "${array[@]}"
30
```

array.uniq()

Sorts and uniqs the array and prints it to STDOUT

Example

```
declare -a unsorted=(hello hello hello goodbye)
local uniqued="$(array.sort-numeric "${unsorted[@]}")"
```

Index

section()

Prints a "arrow-like" line using powerline characters

Arguments

- = @arg1 Width (optional) — only intepretered as width if the first argument is a number.
- = @args Text to print

is-dbg()

Checks if we have debug mode enabled

dbg()

Local debugging helper, activate it with DEBUG=1

Index

path.add()

Adds valid directories to those in the PATH and prints to the output. DOES NOT MODIFY $PATH

path.append()

Appends valid directories to those in the PATH, and exports the new value of the PATH

PATH_add()

This function exists within direnv, but since we are sourcing in .envrc we need to have this defined to avoid errors.

osx.sh

Overview

OSX Specific Helpers and Utilities

Index

osx.app.is-installed()

Checks if a given parameter matches any of the installed applications under /Applications and ~/Applications

By the default prints the matched application. Pass `-q` as a second argument to disable output.

Example

```
▯ osx.app.is-installed safari
Safari.app
▯ osx.app.is-installed safari -q && echo installed
installed
▯ osx.app.is-installed microsoft -c
6
```

Arguments

- **$1** (a): string value to match (case insentively) for an app name
- $2.. additional arguments to the last invocation of `grep`

Exit codes

- **0**: if match was found
- **1**: if not

Index

- db.psql.connect.db-settings-toml()

db.config.parse()

Returns a space-separated values of db host, db name, username and password

Example

```
db.config.set-file ~/.db/database.yml
db.config.parse development
##=> hostname dbname dbuser dbpass
declare -a params=($(db.config.parse development))
echo ${params[0]} # host
```

db.psql.connect()

Connect to one of the databases named in the YAML file, and optionally pass additional arguments to psql. Informational messages are sent to STDERR.

Example

```
db.psql.connect production
db.psql.connect production -c 'show all'
```

db.psql.connect.just-data()

Similar to the db.psql.connect, but outputs just the raw data with no headers.

Example

```
db.psql.connect.just-data production -c 'select datname from pg_database;'
```

db.psql.connect.table-settings-set()

Set per-table settings, such as autovacuum, eg:

Example

```
db.psql.connect.table-settings-set prod users autovacuum_analyze_threshold 1000000
db.psql.connect.table-settings-set prod users autovacuum_analyze_scale_factor 0
```

db.psql.db-settings()

Print out PostgreSQL settings for a connection specified by args

Example

```
db.psql.db-settings -h localhost -U postgres appdb
```

db.psql.connect.db-settings-pretty()

Print out PostgreSQL settings for a named connection

Example

```
db.psql.connect.db-settings-pretty primary
```

Arguments

- = @arg1 dbname database entry name in ~/.db/database.yml

db.psql.connect.db-settings-toml()

Print out PostgreSQL settings for a named connection using TOML/ini format.

Example

```
db.psql.connect.db-settings-toml primary > primary.ini
```

Arguments

- = @arg1 dbname database entry name in ~/.db/database.yml

lib/shdoc.sh

Helpers to install gawk and shdoc properly.0

Overview

see `${BASHMATIC_HOME}/lib/shdoc.md` for an example of how to use SHDOC. and also project's github page.

Index

gawk::install()

Installs gawk into /usr/local/bin/gawk

shdoc::install()

Installs shdoc unless already exists

shdoc::reinstall()

Reinstall shdoc completely

Bashmatic Utilities and aliases for Git revision control system.

Functions in this file manage git repos, including this one.

Overview

Reads the remote of a repo by name provided as an argument (or defaults to "origin") and opens it in the browser.

Index

git.open()

Reads the remote of a repo by name provided as an argument (or defaults to "origin") and opens it in the browser.

Example

```
git clone git@github.com:kigster/bashmatic.git
cd bashmatic
source init.sh
git.open
git.open origin # same thing
```

Arguments

- **$1** (optional): name of the remote to open, defaults to "orogin"

Index

pg.is-running()

Returns true if PostgreSQL is running locally

pg.running.server-binaries()

if one or more PostgreSQL instances is running locally, prints each server's binary postgres file path

pg.running.data-dirs()

For each running server prints the data directory

pg.server-in-path.version()

Grab the version from `postgres` binary in the PATH and remove fractional sub-version

Index

dir.short-home()

Replaces the first part of the directory that matches ${HOME} with '~/'

is.sh

Overview

Various validations and asserts that can be chained and be explicit in a DSL-like way.

Index

__is.validation.error()

Invoke a validation on the value, and process the invalid case using a customizable error handler.

Arguments

* = @arg1 func Validation function name to invoke

* = @arg2 var Value under the test

* = @arg4 error_func Error function to call when validation fails

Exit codes

* **0**: if validation passes

is-validations()

Returns the list of validation functions available

__is.validation.ignore-error()

Private function that ignores errors

__is.validation.report-error()

Private function that ignores errors

whenever()

a convenient DSL for validating things

Example

```
whenever /var/log/postgresql.log is.an-empty-file && {
   touch /var/log/postgresql.log
}
```

util.sh

Overview

Miscellaneous utilities.

Index

util.rot13-stdin()

Convert STDIN using rot13

Example

```
echo "test" | util.rot13-stdin
```

Bashmatic Utilities for PDF file handling

Overview

Install and uses GhostScript to manipulate PDFs.

Index

pdf.combine()

Combine multiple PDFs into a single one using ghostscript.

Example

```
pdf.combine ~/merged.pdf 'my-book-chapter*'
```

Arguments

- **$1** (pathname): to the merged file
- **...** (the): rest of the PDF files to combine