

unc ByteToReadableSize

```
func ByteToReadableSize(bigNum int64) string
```

ByteToReadableSize transform a byte size into human readable form sizes (kb, Mb, Gb, Tb, Pb). Takes 1 argument and returns a HR string for size

1: bigNum int64 (size in bytes)

Returns:

1: string (size in human readable form: Pb, Tb, Gb, etc)

func CleanData

```
func CleanData(rawData [][]string) ([][]string, []string)
```

CleanData removes first column for [][]string matrix. Ideally the format returned by EncapData() function in second position

1: [][]string (Raw data from Encap(), including dirs conditional y/n in first colum

Returns:

1: [][]string (Same matrix without first colum)

2: []string (Folder y/n confirmation string obtained from argument to this function)

func EncapData

```
func EncapData(fileInfo []fs.FileInfo, root string) ([][]int, [][]string, error, int64, int)
```

EncapData extracts data from a []fs.FileInfo dataset in a given path (string).

1: fileInfo []fs.FileInfo (obtained from os.Open File -> Readdir()) 2: root string (Path where files are located)

Returns:

1: [][]int (File sizes matrix)

2: [][]string (File info -as in [n_files]{isDir, lastM, fName, size_HR_Format})

3: error (Returns this error when trying to obtain os.Stat(/path/to/file/name/) for each file

4: int64 (Sum of total file sizes in given path)

5: int (total number of files in given path)

func FastSwitchSli

```
func FastSwitchSli(strUnordered [][]string, orderedSli [][]int, origPos int) [][]string
```

FastSwitchSli sorts a [n_files][5]string dataset obtained from <- GetPathInfo() <- EncapData().

Takes 3 arguments: 1: [][]string (Unordered string matrix with folder files data) 2: [][]int (Sorted slice with file size and original position in primitive raw data slice) 3: int (original position of files, in ordered fileSize slice's rows. Basically its col_index)

Returns:

1: [][]string (Fully formatted array with file data. Ordered by size. later derived to RenderData() function for CLI display purpose)

func FileSizeSort

func FileSizeSort(sli [][]int, sizePos int)

FileSizeSort sorts a [][]int slice matrix of file data, by size.

Takes 2 arguments:

1: sli [][]int (size matrix with size and original position as column values in every row) 2: sizePort int (as first argument (Bigger first, smaller last) by calling Swap() function

<No return>

func GetPath

func GetPath(args []string) (string, bool)

GetPath extracts path from CLI argument, if not given it returns current directory path

Takes 1 argument: 1: args []string (os.Args)

Returns:

1: string (argument path or current working directory)
2: bool (Yes for argument with path from CLI call to blit program)

func GetPathInfo

func GetPathInfo(root string, cli_ON bool) ([][]int, [][]string, error, int64, int)

GetPathInfo extracts a info from files in a given dir path and returns in cascade from EncapData() as -> Matrix [][]int for Sizes; [][]string for File Data as [n_files]{isDir, lastM, fName, size_HR_Format}; error(err); total file_size(int64); total files(int)

Takes 2 arguments: 1: root string (Given path from GetPath()) 2: cli_ON bool (True when path was given from command line to blit program)

Returns (same as EncapData() :

```
1: [][]int          (File sizes matrix)
2: [][]string       (Contains file info -> [n_files]{isDir, lastM, fName, size_HR_Format} )
3: error            (Flagged error obtained from os.Stat(/path/to/file/name/) for each file in
2nd argument dataset)
4: int64            (Sum of total file sizes in given path)
5: int              (total number of files in given path)
```

func RenderData

```
func RenderData(dirs []string, data [][]string, totSize int64, totFiles int)
```

RenderData renders a table in CLI. Takes 4 arguments with information from Files in path given as first argument to the program

1: []string (Slice with y/n values for Directory) 2: [][]string (Sorted Slice from biggest file to lowest size) 3: int64 (Total scanned file size combined) 4: int (Total files in given path)//

<No return>

func Swap ¶

```
func Swap(sli [][]int, i int)
```

Swap switches positions of 2 rows from [][]int slice. Rows swapped are i and i+1 index (Takes i int as second argument)

Takes 2 arguments:

```
1: sli [][]int (Slice containing file size information in 2 columns)
2: i int        (i and i+1 positions where rows are going to be swapped)
```

<No return>