
FPBox Documentation

Release 0.7.0

AN3223

Sep 24, 2019

Contents

Python Module Index	3
Index	5

`fpbox.funcs.binmap(f, xs)`
Strict version of `lazy_binmap`

`fpbox.funcs.c(f, g)`
Function composition

`fpbox.funcs.collect(items, convert=(<class 'list'>, <class 'tuple'>), convert_to=<class 'tuple'>)`
Converts a nested list/tuple/generator into a tuple. If no nested list/tuple/generator is found (or if multiple are found) then “items” is returned unchanged to the caller. Useful for generic functions.

Parameters

- **items** – Target sequence
- **convert** – Tuple of types to convert into target type
- **convert_to** – Target type

Returns The collected sequence

`fpbox.funcs.compose(*fs)`
Function composition over a list of functions

`fpbox.funcs.curry(f, args_supplied=())`
Takes a function, and then returns a function that takes each argument for the original function through `__call__`. You probably shouldn't use this with builtins! Even if it seems to work with a builtin, it might not work properly in previous versions of Python.

`fpbox.funcs.filter(f, xs)`
Strict version of `filter`

`fpbox.funcs.head(xs)`

`fpbox.funcs.init(xs)`

`fpbox.funcs.last(xs)`

`fpbox.funcs.lazy(f)`
A decorator to simply yield the result of a function

`fpbox.funcs.lazy_binmap(f, xs)`
Maps a binary function over a sequence. The function is applied to each item and the item after it until the last item is reached.

`fpbox.funcs.lazy_reduce(f, xs)`
Lazy version of `functools.reduce`

`fpbox.funcs.lazy_reverse_binmap(f, xs)`
Same as `lazy_binmap`, except the parameters are flipped for the binary function

`fpbox.funcs.map(f, xs)`
Strict version of `map`

`fpbox.funcs.partition(f, xs)`
Works similar to `filter`, except it returns a two-item tuple where the first item is the sequence of items that passed the filter and the second is a sequence of items that didn't pass the filter

`fpbox.funcs.reverse(xs)`
Returns a reversed sequence

`fpbox.funcs.reverse_binmap(f, xs)`
Strict version of `lazy_reverse_binmap`

`fpbox.funcs.sum(xs)`
A “sum” implementation that can take advantage of operator overloading

`fpbox.funcs.tail(xs)`

class `fpbox.types.Array(*items)`

Immutable homogenous collection. It can be initialized with either a single list/tuple/generator (which will return an Array consisting of the contents of said list/tuple/generator) or it can just be given multiple arguments to initialize the Array with

class `fpbox.types.Char(char)`

Holds a single character

exception `fpbox.types.FPboxException`

class `fpbox.types.Stream(xs)`

Takes any iterable, returns a Stream object that gives access to a set of lazy (FP-related) methods. Some things to note: no methods mutate the iterable, most methods return a Stream object, and the Stream objects themselves are generators that yield the contents of the original iterable

dropwhile (*f*)

filter (*f*)

list ()

Packs the stream up into a list

map (*f*)

reduce (*f*)

takewhile (*f*)

tuple ()

Packs the stream up into a tuple

`fpbox.types.chars(string)`

Helper function that returns an array of characters from a string

f

`fpbox.funcs`, [??](#)
`fpbox.types`, [2](#)

A

`Array` (*class in fpbox.types*), 2

B

`binmap()` (*in module fpbox.funcs*), 1

C

`c()` (*in module fpbox.funcs*), 1

`Char` (*class in fpbox.types*), 2

`chars()` (*in module fpbox.types*), 2

`collect()` (*in module fpbox.funcs*), 1

`compose()` (*in module fpbox.funcs*), 1

`curry()` (*in module fpbox.funcs*), 1

D

`dropwhile()` (*fpbox.types.Stream method*), 2

F

`filter()` (*fpbox.types.Stream method*), 2

`filter()` (*in module fpbox.funcs*), 1

`fpbox.funcs` (*module*), 1

`fpbox.types` (*module*), 2

`FPboxException`, 2

H

`head()` (*in module fpbox.funcs*), 1

I

`init()` (*in module fpbox.funcs*), 1

L

`last()` (*in module fpbox.funcs*), 1

`lazy()` (*in module fpbox.funcs*), 1

`lazy_binmap()` (*in module fpbox.funcs*), 1

`lazy_reduce()` (*in module fpbox.funcs*), 1

`lazy_reverse_binmap()` (*in module fpbox.funcs*),
1

`list()` (*fpbox.types.Stream method*), 2

M

`map()` (*fpbox.types.Stream method*), 2

`map()` (*in module fpbox.funcs*), 1

P

`partition()` (*in module fpbox.funcs*), 1

R

`reduce()` (*fpbox.types.Stream method*), 2

`reverse()` (*in module fpbox.funcs*), 1

`reverse_binmap()` (*in module fpbox.funcs*), 1

S

`Stream` (*class in fpbox.types*), 2

`sum()` (*in module fpbox.funcs*), 1

T

`tail()` (*in module fpbox.funcs*), 1

`takewhile()` (*fpbox.types.Stream method*), 2

`tuple()` (*fpbox.types.Stream method*), 2