My Project

Generated by Doxygen 1.8.11

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 4

Class Documentation

4.1 it::help::Iterable< T > Struct Template Reference

Public Types

- using **iterator** = typename T::iterator

The documentation for this struct was generated from the following file:

- /home/matheus/Algoritmos/ZipIter/ZipIter/ZipIter.h

4.2 it::help::Iterable< T * > Struct Template Reference

Public Types

- using **iterator** = T *

The documentation for this struct was generated from the following file:

- /home/matheus/Algoritmos/ZipIter/ZipIter/ZipIter.h

4.3 it::help::Iterable< T[N]> Struct Template Reference

Public Types

- using **iterator** = T[N]

The documentation for this struct was generated from the following file:

- /home/matheus/Algoritmos/ZipIter/ZipIter/ZipIter.h

4.4 it::help::IterTagOrder< class > Struct Template Reference

The documentation for this struct was generated from the following file:

- /home/matheus/Algoritmos/ZipIter/ZipIter/ZipIter.h

4.5 it::help::IterTagOrder< std::bidirectional_iterator_tag > Struct Template Reference

Static Public Attributes

- static constexpr int **value** = IterTagOrder < std::forward_iterator_tag >::value + 1

The documentation for this struct was generated from the following file:

- /home/matheus/Algoritmos/ZipIter/ZipIter/ZipIter.h

4.6 it::help::IterTagOrder< std::forward_iterator_tag > Struct Template Reference

Static Public Attributes

- static constexpr int **value** = 0

The documentation for this struct was generated from the following file:

- /home/matheus/Algoritmos/ZipIter/ZipIter/ZipIter.h

4.7 it::help::IterTagOrder< std::random_access_iterator_tag > Struct Template Reference

Static Public Attributes

- static constexpr int **value** = IterTagOrder < std::bidirectional_iterator_tag >::value + 1

The documentation for this struct was generated from the following file:

- /home/matheus/Algoritmos/ZipIter/ZipIter/ZipIter.h

4.8 it::help::SelectIterTag<... > Struct Template Reference

The documentation for this struct was generated from the following file:

- /home/matheus/Algoritmos/ZipIter/ZipIter/ZipIter.h

4.9 it::help::SelectIterTag< Iter > Struct Template Reference

Public Types

- using **type** = Iter

The documentation for this struct was generated from the following file:

- /home/matheus/Algoritmos/ZipIter/ZipIter/ZipIter.h

4.10 it::help::SelectIterTag< Iter, Iters... > Struct Template Reference

Inheritance diagram for it::help::SelectIterTag< Iter, Iters... >:

Collaboration diagram for it::help::SelectIterTag< Iter, Iters... >:

The documentation for this struct was generated from the following file:

- /home/matheus/Algoritmos/ZipIter/ZipIter/ZipIter.h

4.11 it::help::SelectIterTag< Iter1, Iter2 > Struct Template Reference

Public Types

- using **type** = std::conditional_t< ((IterTagOrder_v< Iter1 >)< (IterTagOrder_v< Iter2 >)), Iter1, Iter2 >

The documentation for this struct was generated from the following file:

- /home/matheus/Algoritmos/ZipIter/ZipIter/ZipIter.h

4.12 it::UnZip< F > Struct Template Reference

Public Member Functions

- **UnZip** (const F &f=F())
- template<typename... Args, typename Indexes = std::make_index_sequence<sizeof...(Args)>>
  void **operator()** (std::tuple< Args... > &&tup)
- template<typename... Args, std::size_t... Is>
  void **unpack** (std::tuple< Args... > &tup, std::index_sequence< Is... >)

Public Attributes

- F **f**

The documentation for this struct was generated from the following file:

- /home/matheus/Algoritmos/ZipIter/ZipIter/ZipIter.h

4.13 it::Zip< Containers > Class Template Reference

Public Types

- using **value_type** = std::tuple< Containers... >
- using **iterator** = ZipIter< typename help::Iterable< std::decay_t< Containers > >::iterator... >
- using **const_iterator** = iterator

Public Member Functions

- **Zip** (Containers...conts)
- iterator **begin** ()
- const_iterator **begin** () const
- iterator **end** ()
- const_iterator **end** () const
- template< class Tag = typename iterator::iterator_category, std::enable_if_t< std::is_same< Tag, std::random_access_iterator_tag >↩ ::value, int > = 0>
  decltype(auto) **operator[ ]** (std::size_t pos)
- template< class Tag = typename iterator::iterator_category, std::enable_if_t< std::is_same< Tag, std::random_access_iterator_tag >↩ ::value, int > = 0>
  decltype(auto) **operator[ ]** (std::size_t pos) const
- std::size_t **size** () const

Static Public Attributes

- static constexpr std::size_t **containersSize** = sizeof... (Containers)

The documentation for this class was generated from the following file:

- /home/matheus/Algoritmos/ZipIter/ZipIter/ZipIter.h

4.14 it::ZipIter< Iters > Class Template Reference

Inheritance diagram for it::ZipIter< Iters >:

Collaboration diagram for it::ZipIter< Iters >:

Public Types

- using **Base** = std::iterator< help::SelectIterTag_t< typename std::iterator_traits< Iters >::iterator_category... >, std::tuple< Iters... > >
- using **value_type** = typename Base::value_type
- using **reference** = typename Base::reference
- using **difference_type** = typename Base::difference_type
- using **iterator_category** = typename Base::iterator_category

Public Member Functions

- **ZipIter** (Iters...iterators)
- **ZipIter** (std::tuple< Iters... > iters)
- ZipIter & **operator++** ()
- ZipIter **operator++** (int)
- template<class Tag = iterator_category, std::enable_if_t< std::is_same< Tag, std::bidirectional_iterator_tag >::value, int > = 0>
  ZipIter & **operator--** ()
- template<class Tag = iterator_category, std::enable_if_t< std::is_same< Tag, std::bidirectional_iterator_tag >::value, int > = 0>
  ZipIter **operator--** (int)
- template<class Tag = iterator_category, std::enable_if_t< std::is_same< Tag, std::random_access_iterator_tag >::value, int > = 0>
  ZipIter **operator+** (int inc) const
- template<class Tag = iterator_category, std::enable_if_t< std::is_same< Tag, std::random_access_iterator_tag >::value, int > = 0>
  ZipIter **operator-** (int inc) const
- difference_type **operator+** (const ZipIter &zp) const
- difference_type **operator-** (const ZipIter &zp) const
- template<typename Iter >
  bool **operator==** (const Iter &iter) const
- template<typename Iter >
  bool **operator!=** (const Iter &iter) const
- decltype(auto) **operator*** ()
- decltype(auto) **operator*** () const

The documentation for this class was generated from the following file:

- /home/matheus/Algoritmos/ZipIter/ZipIter/ZipIter.h

Chapter 5

File Documentation

5.1 /home/matheus/Algoritmos/ZipIter/ZipIter/ZipIter.h File Reference

Simple facilities to iterate through multiple containers at the same time similar to Python's zip.

```
#include <type_traits>
#include <tuple>
#include <iterator>
```

Include dependency graph for ZipIter.h:

Index