type类：每个对象代表1个数据类型

|  |  |
| --- | --- |
| type(对象) |  |

|  |
| --- |
| import inspect  inspect.getmro(**有\_\_mro\_\_属性的对象**)  from pprint import pprint  pprint(inspect.getclasstree(inspect.getmro(**有\_\_mro\_\_属性的对象**))) |

容器（序列、映射、集合等）

Sequence和Iterable区别

<http://blog.axant.it/2010/08/25/python-sequence-vs-iterable-protocols/>

判断1个对象的类是否是抽象类Iterable的子类

|  |
| --- |
| import collections  isinstance(**表达式**,collections.Iterable) |

序列（list、tuple、range、str）

|  |  |  |  |
| --- | --- | --- | --- |
| 长度 | | len(**序列**) |  |
| 取元素 | 索引 | **序列**[**整数**] | 得到1个元素（的地址）。  1、下标是整数，最小-n，最大n-1。 |
| 分片 | **序列**[(**整数**)?:(**整数**)?(:**整数**)?] | 得到1个序列（由元素的地址组成）（当然不包括结束位置的元素）。  1、开始位置、结束位置、步长是整数。  2、步长不等于0。  3、若省略开始位置或结束位置，   |  |  |  | | --- | --- | --- | |  | 步长小于0 | 步长大于0 | | 开始位置 | 序列结尾 | 序列开头 | | 结束位置 | 序列开头之前 | 序列结尾之后 |   4、步长默认是1。  ！！！开始位置和结束位置可以超出范围，超出范围时算作序列开头之前或序列结尾之后而不是取模。 |
| 改 | | **索引**=**表达式** | （修改的是序列里存的地址，元素没动。） |
| **分片**=**Iterable** | （修改的是序列里存的地址，元素没动。下同。） |
| 删 | | del **索引** | （删的是序列里存的地址，元素没动。） |
| （**分片**=[]） | （删的是序列里存的地址，元素没动。下同。） |
| 追加 | | **序列**+**序列** | 得到1个新序列。  1、2个序列数据类型相同。 |
| 重复 | | **序列**\***整数** | **整数**不大于0时得到[]。 |
| 成员资格判断 | | **表达式** in **序列**  **表达式** not in **序列** | 1、比较的是值，不是地址。 |
| （元素可比较） | | min(**Iterable**)  max(**Iterable**) |  |
| （元素可相加） | | sum(**Iterable**,**初始值**=0) |  |

list

|  |  |  |
| --- | --- | --- |
| 构造函数 | list()  list(**Iterable**) |  |
| 插入 | **列表**.insert(**元素**) |  |
| （**分片**=**Iterable**） |  |
| 追加 | **列表**.append(**元素**) | 修改列表，得到None。 |
| **列表**.extend(**Iterable**) | 修改列表，得到None。 |
| 取元素+删元素 | **列表**.pop((**整数**)?) | 得到1个元素（的地址）。  1、下标是整数，最小-n，最大n-1。下标超出范围，抛出IndexError异常。  2、省略下标，取和删最后1个元素。 |
| 删元素 | **列表**.remove(**元素**) | 修改列表，得到None（只删第1个）。  1、若列表中没有该元素，抛出ValueError异常。 |
| 倒置 | **列表**.reverse() | 修改列表，得到None。 |
| 元素出现次数 | **列表**.count(**元素**) | 1、比较的是值，不是地址。  ！！！只算列表直接包含的元素。 |
| 元素位置 | **列表**.index(**元素**) | 1、若列表中没有该元素，抛出ValueError异常。 |
| （元素可比较） | **列表**.sort() | 修改列表，得到None。  ！！！是稳定排序。 |
| 随机排序 | from random import shuffle  shuffle(**列表**) | 修改列表，得到None。 |

|  |  |  |
| --- | --- | --- |
| 反向迭代器 | reversed(**列表**) |  |
|  |  |  |
|  |  |  |

tuple

|  |  |  |
| --- | --- | --- |
| 构造函数 | tuple()  tuple(**Iterable**) |  |

list相当于数组数据类型，元素数据类型一般相同，长度一般会改变；tuple相当于const结构体数据类型，元素数据类型一般不一定相同，创建后长度不改变。

set

|  |  |  |
| --- | --- | --- |
| 构造函数 | set()  set(**Iterable**) | 1、自动去除重复元素。 |
| 长度 | len(**集合**) |  |
| 增加 | **集合**.add(**元素**) | 修改集合，得到None。  1、若列表中有该元素，不造成任何影响。 |
|  |  |  |
| 删除 | **集合**.remove(**元素**) | 修改集合，得到None。  1、若列表中没有该元素，抛出KeyError异常。 |
|  |  |  |
| 包含关系 | **集合**.issubset(**Iterable**)  **集合**.issuperset(**Iterable**)  **集合**<**集合**  **集合**<=**集合**  **集合**>**集合**  **集合**>=**集合** |  |
| 并 | **集合**.union(**Iterable**)  **集合**|**集合** | 得到1个新集合。 |
| 差 | **集合**.intersection(**Iterable**)  **集合**&**集合** | 得到1个新集合。 |
| 交 | **集合**.difference(**Iterable**)  **集合**-**集合** | 得到1个新集合。 |
| 对称差 | **集合**.symmetric\_difference(**Iterable**)  **集合**^**集合** | 得到1个新集合。 |
| （元素可比较） | min(**Iterable**)  max(**Iterable**) |  |
| （元素可相加） | sum(**Iterable**,**初始值**=0) |  |

|  |
| --- |
| sorted(**Iterable**)  得到1个新列表。 |
|  |

类型对象

input repr

import time

|  |
| --- |
| time()  得到当前时间，单位是s。 |
|  |