Python 3.1 快速導覽 - 內建序列型態

內建的序列型態 (sequence types) 共有六種，分別是

|  |  |
| --- | --- |
| **型態** | **描述** |
| [str](http://pydoing.blogspot.com/2011/03/python-stringtype.html) | 字串 (string) ，不可變 (immutable) |
| [bytes](http://pydoing.blogspot.com/2011/03/python-bytestype.html) | 字節 (byte) ，不可變 |
| [bytearray](http://pydoing.blogspot.com/2011/03/python-bytearraytype.html) | 字節陣列 (byte array) ，可變 (mutable) |
| [list](http://pydoing.blogspot.com/2011/03/python-listtype.html) | 串列 (list) ，可變 |
| [tuple](http://pydoing.blogspot.com/2011/03/python-tupletype.html) | 序對 (tuple) ，不可變 |
| [range](http://pydoing.blogspot.com/2011/03/python-rangetype.html) | 內建函數 range() 回傳的物件 (object) ，常用於 **for** 迴圈 (for loop) |

可進行以下的計算

|  |  |
| --- | --- |
| **計算** | **描述** |
| x in s | 判斷 x 是否在 s 中 |
| x not in s | 判斷 x 是否不在 s 中 |
| s + t | 連接 s 及 t |
| s \* n, n \* s | 將 s 重複 n 次連接 s 本身 |
| s[i] | 取得索引值 i 的元素 |
| s[i:j] | 取得索引值 i 到 j 的子序列 |
| s[i:j:k] | 取得索引值 i 到 j ，間隔 k 的子序列 |
| len(s) | 回傳 s 的元素個數 |
| min(s) | 回傳 s 中的最小值 |
| max(s) | 回傳 s 中的最大值 |
| s.index(i) | 取得 s 中第一次出現 i 的索引值 |
| s.count(i) | 累計 s 中 i 出現的個數 |

字串型態有以下的方法 (method)

|  |  |
| --- | --- |
| **方法** | **描述** |
| [str.capitalize()](http://pydoing.blogspot.com/2011/03/python-strcapitalize.html) | 回傳將 str 改成首字母大寫，其餘字母小寫的字串 |
| [str.center(width[, fillchar])](http://pydoing.blogspot.com/2011/03/python-strcenter.html) | 回傳一個將 str 設置字串中央，長度 width 的新字串， fillchar 為填充字元，預設為空格 |
| [str.count(sub[, start[, end]])](http://pydoing.blogspot.com/2011/03/python-strcount.html) | 計算 sub 出現的次數， start 為起始計算索引值， end 為結束索引值 |
| [str.encode(encoding="utf-8", errors="strict")](http://pydoing.blogspot.com/2011/03/python-strencode.html) | 回傳 encoding 版本的 bytes 物件 |
| [str.endswith(suffix[, start[, end]])](http://pydoing.blogspot.com/2011/03/python-strendswith.html) | 判斷 str 是否以 suffix 結尾 |
| [str.expandtabs([tabsize])](http://pydoing.blogspot.com/2011/03/python-strexpandtabs.html) | 將 tab 符號以 tabsize 的空格數替換 |
| [str.find(sub[, start[, end]])](http://pydoing.blogspot.com/2011/03/python-strfind.html) | 回傳 sub 第一次出現的索引值 |
| [str.format(\*args, \*\*kwargs)](http://pydoing.blogspot.com/2011/03/python-strformat.html) | 進行格式化字串運算 |
| [str.index(sub[, start[, end]])](http://pydoing.blogspot.com/2011/03/python-strindex.html) | 回傳 sub 第一次出現的索引值 |
| [str.isalnum()](http://pydoing.blogspot.com/2011/03/python-strisalnum.html) | 判斷字串中的字元是否至少一個是字母或數字 |
| [str.isalpha()](http://pydoing.blogspot.com/2011/03/python-strisalpha.html) | 判斷字串中的字元是否至少一個是字母 |
| [str.isdecimal()](http://pydoing.blogspot.com/2011/03/python-strisdecimal.html) | 判斷字串中所有字元是否是十進位數字 |
| [str.isdigit()](http://pydoing.blogspot.com/2011/03/python-strisdigit.html) | 判斷字串中所有字元是否是數字 |
| [str.isidentifier()](http://pydoing.blogspot.com/2011/03/python-strisidentifier.html) | 判斷字串是否可作為合法的識別字 |
| [str.islower()](http://pydoing.blogspot.com/2011/03/python-strislower.html) | 判斷字串中所有字母字元是否都是小寫字母 |
| [str.isnumeric()](http://pydoing.blogspot.com/2011/03/python-strisnumeric.html) | 判斷字串中所有字元是否是數字 |
| [str.isprintable()](http://pydoing.blogspot.com/2011/03/python-strisprintable.html) | 判斷字串中所有字元是否都屬於可見字元 |
| [str.isspace()](http://pydoing.blogspot.com/2011/03/python-strisspace.html) | 判斷字串是否為空格字元 |
| [str.istitle()](http://pydoing.blogspot.com/2011/03/python-stristitle.html) | 判斷字串是否適合當作標題 |
| [str.isupper()](http://pydoing.blogspot.com/2011/03/python-strisupper.html) | 判斷字串中所有字母字元是否都是大寫字母 |
| [str.join(iterable)](http://pydoing.blogspot.com/2011/03/python-strjoin.html) | 回傳將 str 連結 iterable 各元素的字串 |
| [str.ljust(width[, fillchar])](http://pydoing.blogspot.com/2011/03/python-strljust.html) | 回傳將 str 在寬度 width 向左對齊的字串， fillchar 為填充字元，預設為空格 |
| [str.lower()](http://pydoing.blogspot.com/2011/03/python-strlower.html) | 將 str 的英文字母都改成小寫 |
| [str.lstrip([chars])](http://pydoing.blogspot.com/2011/03/python-strlstrip.html) | 回傳將 str 左邊具有 chars 字元去除的拷貝版本， chars 預設為空格符號 |
| [static str.maketrans(x[, y[, z]])](http://pydoing.blogspot.com/2011/03/python-strmaketrans.html) | 回傳 x 與 y 配對的 Unicode 編碼字典，若有提供 z ， z 中的字元會跟 None 配對 |
| [str.partition(sep)](http://pydoing.blogspot.com/2011/03/python-strpartition.html) | 以 sep 分割 str 為三個部份，結果回傳具有三個子字串的序對 |
| [str.replace(old, new[, count])](http://pydoing.blogspot.com/2011/03/python-strreplace.html) | 將 str 中的 old 子字串以 new 代換 |
| [str.rfind(sub[, start[, end]])](http://pydoing.blogspot.com/2011/03/python-strrfind.html) | 尋找最右邊的 sub ，也就是索引值最大的 sub |
| [str.rindex(sub[, start[, end]])](http://pydoing.blogspot.com/2011/03/python-strrindex.html) | 尋找最右邊的 sub ，也就是索引值最大的 sub |
| [str.rjust(width[, fillchar])](http://pydoing.blogspot.com/2011/03/python-strrjust.html) | 回傳將 str 在寬度 width 向右對齊的字串， fillchar 為填充字元，預設為空格 |
| [str.rpartition(sep)](http://pydoing.blogspot.com/2011/03/python-strrpartition.html) | 以 sep 從最右端分割 str 為三個部份，結果回傳具有三個子字串的序對 |
| [str.rsplit([sep[, maxsplit]])](http://pydoing.blogspot.com/2011/03/python-strrsplit.html) | 將 str 從最右端以 sep 分割成子字串，回傳儲存子字串的串列， maxsplit 為子字串最多的數量 |
| [str.rstrip([chars])](http://pydoing.blogspot.com/2011/03/python-strrstrip.html) | 從 str 的最右端中移除 chars 字元，預設為空白字元 |
| [str.split([sep[, maxsplit]])](http://pydoing.blogspot.com/2011/03/python-strsplit.html) | 將 str 以 sep 分割成子字串，回傳儲存子字串的串列， maxsplit 為子字串最多的數量 |
| [str.splitlines([keepends])](http://pydoing.blogspot.com/2011/03/python-strsplitlines.html) | 將 str 以新行符號分割成子字串，回傳儲存子字串的串列 |
| [str.startswith(prefix[, start[, end]])](http://pydoing.blogspot.com/2011/03/python-strstartswith.html) | 判斷 str 是否以 prefix 開頭 |
| [str.strip([chars])](http://pydoing.blogspot.com/2011/03/python-strstrip.html) | 從 str 中移除 chars 字元，預設為空白字元 |
| [str.swapcase()](http://pydoing.blogspot.com/2011/03/python-strswapcase.html) | 將 str 中的英文字母進行大小寫轉換 |
| [str.title()](http://pydoing.blogspot.com/2011/03/python-strtitle.html) | 將 str 轉換成作為標題的字串 |
| [str.translate(map)](http://pydoing.blogspot.com/2011/03/python-strtranslate.html) | 將 str 中的字元以 map 中配對的字元轉換 |
| [str.upper()](http://pydoing.blogspot.com/2011/03/python-strupper.html) | 將 str 的英文字母都改成大寫 |
| [str.zfill(width)](http://pydoing.blogspot.com/2011/03/python-strzfill.html) | 回傳以 0 塞滿 width 的新字串 |

字節型態與字節陣列為相對的資料型態，前者是不可變的，後者是可變的。兩者有以下的方法

|  |  |
| --- | --- |
| **方法** | **描述** |
| [bytes.decode(encoding="utf-8", errors="strict")](http://pydoing.blogspot.com/2011/03/python-bytesdecode.html) | 將 bytes 解譯為指定編碼的字串 |
| [bytearray.decode(encoding="utf-8", errors="strict")](http://pydoing.blogspot.com/2011/03/python-bytearraydecode.html) | 將 bytearray 解譯為指定編碼的字串 |
| [classmethod bytes.fromhex(string)](http://pydoing.blogspot.com/2011/03/python-bytesfromhex.html) | 將具有十六進位數字的字串 string 轉換為字節型態 |
| [classmethod bytearray.fromhex(string)](http://pydoing.blogspot.com/2011/03/python-bytearrayfromhex.html) | 將具有十六進位數字的字串 string 轉換為字節陣列 |
| [bytes.translate(table[, delete])](http://pydoing.blogspot.com/2011/03/python-bytestranslate.html) | 將 bytes 依 table 替換字元， delete 為所要刪除的字元 |
| [bytearray.translate(table[, delete])](http://pydoing.blogspot.com/2011/03/python-bytearraytranslate.html) | 將 bytearray 依 table 替換字元， delete 為所要刪除的字元 |
| [static bytes.maketrans(from, to)](http://pydoing.blogspot.com/2011/03/python-bytesmaketrans.html) | 回傳字節型態的轉換表，使之用於 translate() 方法 |
| [static bytearray.maketrans(from, to)](http://pydoing.blogspot.com/2011/03/python-bytearraymaketrans.html) | 回傳字節陣列的轉換表，使之用於 translate() 方法 |

串列與序對也是相對的資料型態，由於串列也是可變的，也是 Python 中大量運用的工作型態種類，因此有額外以下的計算

|  |  |
| --- | --- |
| **計算** | **描述** |
| s[i] = x | 將索引值 i 的元素指派為 x |
| s[i:j] = t | 將索引值 i 到 j 的元素指派為 t ， t 為迭代器 |
| del s[i:j] | 刪除索引值 i 到 j 的元素 |
| s[i:j:k] = t | 將索引值 i 到 j ，間隔 k 的元素指派為 t ， t 為迭代器 |
| del s[i:j:k] | 刪除索引值 i 到 j ，間隔 k 的元素 |
| [串列綜合運算 list comprehension](http://pydoing.blogspot.com/2011/03/python-listcomprehension.html) | 運用運算式生成新的串列 |

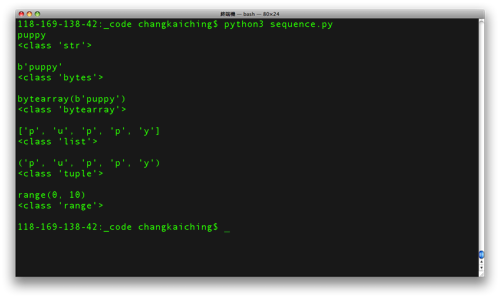
串列型態有以下的方法

|  |  |
| --- | --- |
| **方法** | **描述** |
| [list.append(x)](http://pydoing.blogspot.com/2011/03/python-listappend.html) | 將 x 附加到 list 的最後 |
| [list.extend(x)](http://pydoing.blogspot.com/2011/03/python-listextend.html) | 將 x 中的元素附加到 list 的最後 |
| [list.count(x)](http://pydoing.blogspot.com/2011/03/python-listcount.html) | 計算 list 中 x 出現的次數 |
| [list.index(x[, i[, j]])](http://pydoing.blogspot.com/2011/03/python-listindex.html) | 回傳 x 在 list 最小的索引值 |
| [list.insert(i, x)](http://pydoing.blogspot.com/2011/03/python-listinsert.html) | 將 x 插入 list 索引值 i 的地方 |
| [list.pop([i])](http://pydoing.blogspot.com/2011/03/python-listpop.html) | 取出 list 中索引值為 i 的元素，預設是最後一個 |
| [list.remove(x)](http://pydoing.blogspot.com/2011/03/python-listremove.html) | 移除 list 中第一個 x 元素 |
| [list.reverse()](http://pydoing.blogspot.com/2011/03/python-listreverse.html) | 倒轉 list 中元素的順序 |
| [list.sort([key[, reverse]])](http://pydoing.blogspot.com/2011/03/python-listsort.html) | 排序 list 中的元素 |

舉例示範如下

[?](http://pydoing.blogspot.tw/2011/03/python-sequencetype.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36 | a = "puppy"  print(a)  print(type(a))  print()    b = b"puppy"  print(b)  print(type(b))  print()    c = bytearray(b)  print(c)  print(type(c))  print()    d = list(a)  print(d)  print(type(d))  print()    e = tuple(a)  print(e)  print(type(e))  print()    f = range(10)  print(f)  print(type(f))  print()    # 《程式語言教學誌》的範例程式  # <http://pydoing.blogspot.com/>  # 檔名：sequence.py  # 功能：示範 Python 程式  # 作者：張凱慶  # 時間：西元 2010 年 12 月 |

執行結果如下  


|  |  |
| --- | --- |
| **中英文術語對照** | |
| 序列型態 | sequence types |
| 字串 | string |
| 不可變 | immutable |
| 字節 | byte |
| 可變 | mutable |
| 字節陣列 | byte array |
| 串列 | list |
| 序對 | tuple |
| 物件 | object |
| **for** 迴圈 | for loop |
| 方法 | method |

Python 3.1 快速導覽 - 內建集合型態 (set)

內建的集合型態 (set types) 共有兩種，分別是

|  |  |
| --- | --- |
| **型態** | **描述** |
| set | 集合 (set) ，可變 (mutable) |
| forzenset | 原封集合 (forzenset) ，建立後變不新增或刪除元素 (element) ，因此為不可變 (immutable) |

集合型態的字面常數使用大括弧圍起來，其物件屬於複合資料型態 (compound data type) ，也就是說單一集合型態物件可以包含多個元素，但沒有重複的元素。例如以下 s1

|  |
| --- |
| s1 = {1, 1, 1, 2, 2, 3, 3, 4, 5} |

s1 實際等於 s2

|  |
| --- |
| s2 = {1, 2, 3, 4, 5} |

集合型態的物件可進行以下的運算

|  |  |
| --- | --- |
| **計算** | **描述** |
| x in s | 判斷 x 是否在 s 中 |
| x not in s | 判斷 x 是否不在 s 中 |
| s1 & s2 | 且運算，取得 s1 與 s2 的交集 |
| s2 | s2 | 或運算，取得 s1 與 s2 的聯集 |
| s1 ^ s2 | 對稱差運算，取得 s1 與 s2 的對稱差集 |
| s1 - s2 | 差運算，取得 s1 與 s2 的差集 |
| s1 < s2 | 判斷 s1 是否為 s2 的真子集 |
| s1 <= s2 | 判斷 s1 是否為 s2 的子集 |
| s1 > s2 | 判斷 s2 是否為 s1 的真子集 |
| s1 >= s2 | 判斷 s2 是否為 s1 的子集 |
| len(s) | 回傳 s 的元素個數 |
| min(s) | 回傳 s 中的最小值， s 中的元素必須是相同型態 |
| max(s) | 回傳 s 中的最大值， s 中的元素必須是相同型態 |

集合型態物件的運算大都有相對應的方法 (method)

|  |  |
| --- | --- |
| **方法** | **描述** |
| [s1.intersection(s2)](http://pydoing.blogspot.com/2011/03/python-setintersection.html) | 等於 s1 & s2 |
| [s1.union(s2)](http://pydoing.blogspot.com/2011/03/python-setunion.html) | 等於 s1 | s2 |
| [s1.symmetric\_difference(s2)](http://pydoing.blogspot.com/2011/03/python-setsymmetricdifference.html) | 等於 s1 ^ s2 |
| [s1.difference(s2)](http://pydoing.blogspot.com/2011/03/python-setdifference.html) | 等於 s1 - s2 |
| [s1.issubset(s2)](http://pydoing.blogspot.com/2011/03/python-setissubset.html) | 等於 s1 <= s2 |
| [s1.issuperset(s2)](http://pydoing.blogspot.com/2011/03/python-setissuperset.html) | 等於 s1 >= s2 |
| [s1.isdisjoint(s2)](http://pydoing.blogspot.com/2011/03/python-setisdisjoint.html) | 判斷 s1 與 s2 是否無交集，若無交集，回傳 True |
| [s.copy()](http://pydoing.blogspot.com/2011/03/python-setcopy.html) | 回傳 s 的拷貝 |

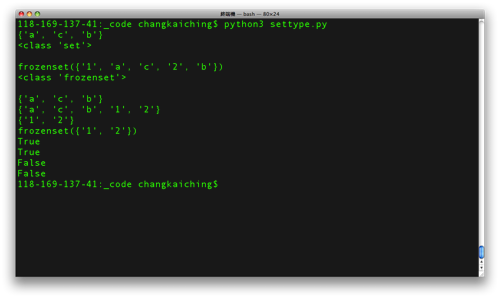
由於 set 型態是可變的，因此有額外兩個新增與刪除元素的方法

|  |  |
| --- | --- |
| **方法** | **描述** |
| [s.add(e)](http://pydoing.blogspot.com/2011/03/python-setadd.html) | 增加 e 為 s 的元素 |
| [s.remove(e)](http://pydoing.blogspot.com/2011/03/python-setremove.html) | 從 s 中刪除元素 e |

舉例示範如下

[?](http://pydoing.blogspot.tw/2011/03/python-settype.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | a = {"a", "b", "c"}  print(a)  print(type(a))  print()    b = frozenset({"a", "b", "c", "1", "2"})  print(b)  print(type(b))  print()    print(a & b)  print(a | b)  print(a ^ b)  print(b - a)  print(a < b)  print(a <= b)  print(a > b)  print(a >= b)    # 《程式語言教學誌》的範例程式  # <http://pydoing.blogspot.com/>  # 檔名：settype.py  # 功能：示範 Python 程式  # 作者：張凱慶  # 時間：西元 2010 年 12 月 |

執行結果如下  


|  |  |
| --- | --- |
| **中英文術語對照** | |
| 集合型態 | set types |
| 集合 | set |
| 可變 | mutable |
| 原封集合 | forzenset |
| 元素 | element |
| 不可變 | immutable |
| 複合資料型態 | compound data type |
| 方法 | method |

Python 3.1 快速導覽 - 內建字典型態 (dict)

內建的字典型態 (dictionary type) 只有一種，由 key:value 配對的複合資料型態 (compound data type)

|  |  |
| --- | --- |
| **型態** | **描述** |
| dict | 由 key:value 配對的複合資料型態 |

建立字典變數可利用大括弧，裡頭以 key:value 為配對的資料項目，每一筆資料再以逗號區隔開，例如

|  |
| --- |
| d1 = {1:"a", 2:"b"} |

上述字典型態的變數 d1 有兩筆資料，第一筆資料的 key 為 1 ， value 為 "a" ，第二筆資料的 key 為 2 ， value 為 "b" 。使用字典須注意， key 必須是不可變的 (immutable) 資料型態，如數字、字串 (string) 等， value 沒有限制，因此有需要的話，使用串列 (list) 或字典皆可。   
  
  
也可以利用字典型態的建構子 (constructor) 建立物件，如下

|  |
| --- |
| d2 = dict(1="a", 2="b")  d3 = dict({1:"a", 2:"b"})  d4 = dict(zip((1, 2), ("a", "a")))  d5 = dict([[2, "b"], [1, "a"]]) |

以上 d2 、 d3 、 d4 、 d5 所得到的字典物件，全都會與 d1 相同。  
  
  
字典物件可進行以下的運算

|  |  |
| --- | --- |
| **計算** | **描述** |
| d[key] | 從 d 中取得 key 的 value |
| d[key] = value | 將 d 的 key 指定為 value |
| del d[key] | 刪除 d 中 key 所指定的 value |
| key in d | 判斷 key 是否在 d 中 |
| key not in d | 判斷 key 是否不在 d 中 |
| iter(d) | 回傳由 d 的 key 建立的迭代器 |
| len(d) | 回傳 d 的配對資料個數 |

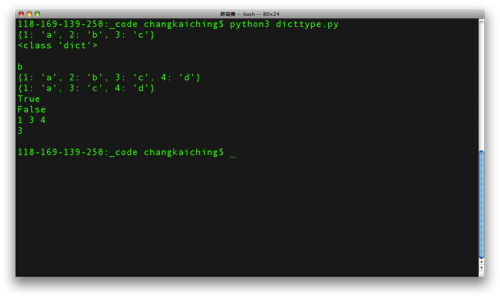
字典物件有以下的方法 (method)

|  |  |
| --- | --- |
| **方法** | **描述** |
| [dict.clear()](http://pydoing.blogspot.com/2011/03/python-dictclear.html) | 清空 dict 的所有配對資料 |
| [dict.copy()](http://pydoing.blogspot.com/2011/03/python-dictcopy.html) | 回傳 dict 的拷貝 |
| [classmethod dict.fromkeys(seq[, value])](http://pydoing.blogspot.com/2011/03/python-dictfromkeys.html) | 由 seq 中的元素構成 key ，每個 key 都給相同的 value 值 |
| [dict.get(key[, default])](http://pydoing.blogspot.com/2011/03/python-dictget.html) | 從 dict 中取得 key 的 value ，若無此 key 則回傳 default ， default 預設為 None |
| [dict.items()](http://pydoing.blogspot.com/2011/03/python-dictitems.html) | 回傳 dict\_items 物件，使 key:value 儲存為序對，然後依序儲存在 dict\_items 物件中 |
| [dict.keys()](http://pydoing.blogspot.com/2011/03/python-dictkeys.html) | 回傳 dict\_items 物件，使 key 依序儲存在 dict\_items 物件中 |
| [dict.pop(key[, default])](http://pydoing.blogspot.com/2011/03/python-dictpop.html) | 將 key 的 value 從 dict 移除，若無此 kay ，回傳 default |
| [dict.popitem()](http://pydoing.blogspot.com/2011/03/python-dictpopitem.html) | 從 dict 移除任意一組 key:value |
| [dict.setdefault(key[, default])](http://pydoing.blogspot.com/2011/03/python-dictsetdefault.html) | 如果 key 在 dict 中，回傳 value 值，反之，將 key:default 加入 dict 之中 |
| [dict.update([other])](http://pydoing.blogspot.com/2011/03/python-dictupdate.html) | 將 dict 以 other 更新 |
| [dict.values()](http://pydoing.blogspot.com/2011/03/python-dictvalues.html) | 回傳 dict\_items 物件，使 value 依序儲存在 dict\_items 物件中 |

舉例示範如下

[?](http://pydoing.blogspot.tw/2011/03/python-dicttype.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | d = {1:"a", 2:"b", 3:"c"}  print(d)  print(type(d))  print()    print(d[2])  d[4] = "d"  print(d)  del d[2]  print(d)  print(3 in d)  print(3 not in d)  for i in iter(d):      print(i, end=" ")  print()  print(len(d))  print()    # 《程式語言教學誌》的範例程式  # <http://pydoing.blogspot.com/>  # 檔名：dicttype.py  # 功能：示範 Python 程式  # 作者：張凱慶  # 時間：西元 2010 年 12 月 |

執行結果如下  


|  |  |
| --- | --- |
| **中英文術語對照** | |
| 字典型態 | dictionary type |
| 複合資料型態 | compound data type |
| 不可變的 | immutable |
| 字串 | string |
| 串列 | list |
| 建構子 | constructor |
| 方法 | method |

float([x])

Convert a string or a number to floating point.

If the argument is a string, it should contain a decimal number, optionally preceded by a sign, and optionally embedded in whitespace. The optional sign may be '+' or '-'; a '+' sign has no effect on the value produced. The argument may also be a string representing a NaN (not-a-number), or a positive or negative infinity. More precisely, the input must conform to the following grammar after leading and trailing whitespace characters are removed:

sign ::= "+" | "-"

infinity ::= "Infinity" | "inf"

nan ::= "nan"

numeric\_value ::= floatnumber | infinity | nan

numeric\_string ::= [sign] numeric\_value

Here floatnumber is the form of a Python floating-point literal, described in Floating point literals. Case is not significant, so, for example, “inf”, “Inf”, “INFINITY” and “iNfINity” are all acceptable spellings for positive infinity.

Otherwise, if the argument is an integer or a floating point number, a floating point number with the same value (within Python’s floating point precision) is returned. If the argument is outside the range of a Python float, an OverflowError will be raised.

For a general Python object x, float(x) delegates to x.\_\_float\_\_().

If no argument is given, 0.0 is returned.

Examples:

>>>>>> float('+1.23')

1.23

>>> float(' -12345\n')

-12345.0

>>> float('1e-003')

0.001

>>> float('+1E6')

1000000.0

>>> float('-Infinity')

-inf

The float type is described in Numeric Types — int, float, complex.

format(value[, format\_spec])

Convert a value to a “formatted” representation, as controlled by format\_spec. The interpretation of format\_spec will depend on the type of the value argument, however there is a standard formatting syntax that is used by most built-in types: Format Specification Mini-Language.

The default format\_spec is an empty string which usually gives the same effect as calling str(value).

A call to format(value, format\_spec) is translated to type(value).\_\_format\_\_(format\_spec) which bypasses the instance dictionary when searching for the value’s \_\_format\_\_() method. A TypeError exception is raised if the method search reaches object and the format\_spec is non-empty, or if either the format\_spec or the return value are not strings.

Changed in version 3.4: object().\_\_format\_\_(format\_spec) raises TypeError if format\_spec is not an empty string.

frozenset([iterable])

Return a new frozenset object, optionally with elements taken from iterable. frozenset is a built-in class. See frozenset and Set Types — set, frozenset for documentation about this class.

For other containers see the built-in set, list, tuple, and dict classes, as well as the collections module.

getattr(object, name[, default])

Return the value of the named attribute of object. name must be a string. If the string is the name of one of the object’s attributes, the result is the value of that attribute. For example, getattr(x, 'foobar') is equivalent to x.foobar. If the named attribute does not exist, default is returned if provided, otherwise AttributeError is raised.

字典的len()函式

len(d)

Return the number of items in the dictionary d.

**zip()結合dict,set,list,tuple用法**

可參考: 精通Python3程式設計 (第二版) p. 145

**群集的複製**

可參考: 精通Python3程式設計 (第二版) pp. 146-148