

# Python内建函数

讲师：Wayne

从业十余载，漫漫求知路

# 内建函数

## □ 标识 id

- 返回对象的唯一标识，CPython返回内存地址

## □ 哈希 hash()

- 返回一个对象的哈希值

## □ 类型 type()

- 返回对象的类型

## □ 类型转换

- float() int() bin() hex() oct() bool() list() tuple() dict() set() complex() bytes() bytearray()

## □ 输入 input([prompt])

- 接收用户输入，返回一个字符串

## □ 打印 print(\*objects, sep=' ', end='\n', file=sys.stdout, flush=False)

- 打印输出，默认使用空格分割、换行结尾，输出到控制台

# 内建函数

- ❑ 对象长度 len(s)
  - ❑ 返回一个集合类型的元素个数
- ❑ isinstance(obj, class\_or\_tuple)
  - ❑ 判断对象obj是否属于某种类型或者元组中列出的某个类型
  - ❑ isinstance(True, int)
- ❑ issubclass(cls, class\_or\_tuple)
  - ❑ 判断类型cls是否是某种类型的子类或元组中列出的某个类型的子类
  - ❑ issubclass(bool, int)

# 内建函数

- ❑ 绝对值abs(x) x为数值
- ❑ 最大值max() 最小值min()
  - ❑ 返回可迭代对象中最大或最小值
  - ❑ 返回多个参数中最大或最小值
- ❑ round(x) 四舍六入五取偶，round(-0.5)
- ❑ pow(x, y) 等价于  $x^y$
- ❑ range(stop) 从0开始到stop-1的可迭代对象；range(start, stop[, step])从start开始到stop-1结束步长为step的可迭代对象
- ❑ divmod(x, y) 等价于 tuple (x//y, x%y)
- ❑ sum(iterable[, start]) 对可迭代对象的所有数值元素求和
  - ❑ sum(range(1,100,2))

# 内建函数

- `chr(i)` 给一个一定范围的整数返回对应的字符

- `chr(97)`   `chr(20013)`

- `ord(c)` 返回字符对应的整数

- `ord('a')`   `ord('中')`

- `str()` 、 `repr()`、 `ascii()` 后面说

# 内建函数

▣ `sorted(iterable[, key][, reverse])` 排序

▣ 返回一个新的列表，默认升序

▣ `reverse` 是反转

`sorted([1, 3, 5])`

`sorted([1, 3, 5], reverse=True)`

`sorted({'c':1, 'b':2, 'a':1})`

# 内建函数

## □ 翻转 reversed(seq)

- 返回一个翻转元素的迭代器

```
list(reversed("13579"))
```

```
{ reversed((2, 4)) } # 有几个元素？
```

```
for x in reversed(['c','b','a']):
```

```
    print(x)
```

```
reversed(sorted({1, 5, 9}))
```

# 内建函数

- 枚举 `enumerate(seq, start=0)`
  - 迭代一个序列，返回索引数字和元素构成的二元组
  - `start`表示索引开始的数字，默认是0

```
for x in enumerate([2,4,6,8]):  
    print(x)
```

```
for x in enumerate("abcde"):  
    print(x,end=" ")
```



# 内建函数

▣ 迭代器和取元素 `iter(iterable)`、`next(iterator[, default])`

▣ `iter` 将一个可迭代对象封装成一个迭代器

▣ `next` 对一个迭代器取下一个元素。如果全部元素都取过了，再次`next`会抛`StopIteration`异常

```
it = iter(range(5))
```

```
next(it)
```

```
it = reversed([1,3,5])
```

```
next(it)
```

# 可迭代对象

## □ 可迭代对象

- 能够通过迭代一次次返回不同的元素的对象。

- 所谓相同，不是指值是否相同，而是元素在容器中是否是同一个，例如列表中值可以重复的，['a', 'a']，虽然这个列表有2个元素，值一样，但是两个'a'是不同的元素不同的索引

- 可以迭代，但是未必有序，未必可索引

- 可迭代对象有：list、tuple、string、bytes、bytearray、range、set、dict、生成器等

- 可以使用成员操作符in、not in，in本质上就是在遍历对象

3 in range(10)

3 in (x for x in range(10))

3 in {x:y for x,y in zip(range(4),range(4,10))}

# 迭代器

## □ 迭代器

- 特殊的对象，一定是可迭代对象，具备可迭代对象的特征
- 通过iter方法把一个可迭代对象封装成迭代器
- 通过next方法，迭代 迭代器对象
- 生成器对象，就是迭代器对象

```
for x in iter(range(10)):  
    print(x)
```

```
g = (x for x in range(10))  
print(type(g))  
print(next(g))  
print(next(g))
```

# 内建函数

## □ 拉链函数zip(\*iterables)

- 像拉链一样，把多个可迭代对象合并在一起，返回一个迭代器
- 将每次从不同对象中取到的元素合并成一个元组

```
list(zip(range(10),range(10)))
```

```
list(zip(range(10),range(10),range(5),range(10)))
```

```
dict(zip(range(10),range(10)))
```

```
{str(x):y for x,y in zip(range(10),range(10))}
```

# 谢谢

咨询热线 400-080-6560