

附录 IV. 内置函数

表IV-1. 内置类型构造函数

函数名	语法	说明
object	<code>class object()</code>	object对象的构造函数。
int	<code>class int([x])</code> <code>class int(x, base=10)</code>	整数的构造函数。
bool	<code>class bool([x])</code>	布尔值的构造函数。
float	<code>class float([x])</code>	浮点数的构造函数。
complex	<code>class complex([real[, imag]])</code>	复数的构造函数。
str	<code>class str(object='')</code> <code>class str(object='', encoding='utf-8', errors='strict')</code>	字符串的构造函数。
bytes	<code>class bytes([source[, encoding[, errors]]])</code>	字节串的构造函数。
bytearray	<code>class bytearray([source[, encoding[, errors]]])</code>	字节数组的构造函数。
memoryview	<code>class memoryview(object)</code>	内存视图的构造函数。
tuple	<code>class tuple([iterable])</code>	元组的构造函数。
list	<code>class list([iterable])</code>	列表的构造函数。
range	<code>class range(stop)</code> <code>class range(start, stop[, step])</code>	范围的构造函数。
dict	<code>class dict(**kwarg)</code> <code>class dict(mapping, **kwarg)</code> <code>class dict(iterable, **kwarg)</code>	字典的构造函数。
set	<code>class set([iterable])</code>	集合的构造函数。
frozenset	<code>class frozenset([iterable])</code>	凝固集合的构造函数。

表IV-2. 装饰器

函数名	语法	说明
classmethod	<code>@classmethod</code>	把方法转换为类方法。
staticmethod	<code>@staticmethod</code>	把方法转换为静态方法。

表IV-3. 基本操作

函数名	语法	说明
id	<code>id(object)</code>	返回指定对象的编号。

函数名	语法	说明
type	<code>class type(object)</code> <code>class type(name, bases, dict, **kwargs)</code>	第一种用法返回指定对象的类型。 第二种用法等价于类定义语句。
repr	<code>repr(object)</code>	返回指定对象的正式字符串表示。
ascii	<code>ascii(object)</code>	返回指定对象的ASCII字符串表示。
format	<code>format(object, format_spec)</code>	返回指定对象的格式化字符串表示。
hash	<code>hash(object)</code>	返回指定对象的哈希值。

表IV-4. 访问属性

函数名	语法	说明
getattr	<code>getattr(object, name[, default])</code>	返回指定对象的指定属性。
setattr	<code>setattr(object, name, value)</code>	设置指定对象的指定属性。
delattr	<code>delattr(object, name)</code>	删除指定对象的指定属性。
hasattr	<code>hasattr(object, name)</code>	判断指定对象是否有指定属性。
dir	<code>dir(object)</code> <code>dir()</code>	第一种用法返回指定对象的有效属性列表。 第二种用法返回当前作用域中的名字列表。
vars	<code>vars(object)</code> <code>vars()</code>	第一种用法返回指定对象的实例属性说明。 第二种用法返回当前作用域中的名字说明。

表IV-5. 类定义

函数名	语法	说明
property	<code>class property(fget=None, fset=None, fdel=None, doc=None)</code>	定义托管属性。
super	<code>class super([type[, object-or-type]])</code>	返回一个代理对象，以访问指定类的基类中的方法。
isinstance	<code>isinstance(object, classinfo)</code>	判断指定对象是否是指定类的实例。
issubclass	<code>issubclass(class, classinfo)</code>	判断指定类是否是其他指定类的子类。
callable	<code>callable(object)</code>	判断指定对象是否可调用。

表IV-6. 名字空间和作用域

函数名	语法	说明
globals	<code>globals()</code>	返回当前全局名字空间中的名字说明。
locals	<code>locals()</code>	返回当前作用域中的名字说明。
eval	<code>eval(expression[, globals[, locals]])</code>	在指定的名字空间下将指定的字符串或代码对象当成表达式来求值，并返回求值结果。

函数名	语法	说明
exec	<code>exec(object[, globals[, locals]])</code>	在指定的名字空间下将指定的字符串或代码对象当成Python脚本来执行。

表IV-7. 模块

函数名	语法	说明
__import__	<code>__import__(name, globals=None, locals=None, fromlist=(), level=0)</code>	被import语句自动调用，以加载指定的模块。

表IV-8. I/O

函数名	语法	说明
print	<code>print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False)</code>	将指定的对象输出至指定的文本流。
input	<code>input([prompt])</code>	将提示信息写入标准输出，然后从标准输入读取一行，将其转换为字符串并返回。
open	<code>open(file, mode='r', buffering=-1, encoding=None, errors=None, newline=None, closefd=True, opener=None)</code>	打开指定的文件，产生相应的流，并返回一个文件对象以对该流进行操作。

表IV-9. 数字和字符串操作

函数名	语法	说明
divmod	<code>divmod(x, y)</code>	带余除法。
pow	<code>pow(base, exp[, mod])</code>	幂或幂取余。
abs	<code>abs(x)</code>	绝对值。
round	<code>round(x[, ndigits])</code>	舍入。
bin	<code>bin(x)</code>	返回整数的二进制表示。
oct	<code>oct(x)</code>	返回整数的八进制表示。
hex	<code>hex(x)</code>	返回整数的十六进制表示。
chr	<code>chr(i)</code>	以字符串形式返回Unicode码位为指定整数的字符。
ord	<code>orc(c)</code>	以整数形式返回指定字符的Unicode码位。

表IV-10. 容器操作

函数名	语法	说明
len	<code>len(object)</code>	返回容器的长度（元素个数）。

函数名	语法	说明
slice	<code>class slice(stop)</code> <code>class slice(start, stop[, step])</code>	返回一个切片对象，以作为索引对序列进行切片。
iter	<code>iter(iterator)</code> <code>iter(callable, sentinel)</code>	第一种用法迭代一个可迭代对象。 第二种用法重复调用一个可调用对象，直到得到指定的返回值。
aiter	<code>aiter(async_iterable)</code>	迭代一个异步可迭代对象。
reversed	<code>reversed(iterable)</code>	反向迭代一个可迭代对象。
next	<code>next(iterator[, default])</code>	返回可迭代对象的下一项。
anext	<code>awaitable anext(async_iterator[, default])</code>	返回异步可迭代对象的下一项。
sorted	<code>sorted(iterable, *, key=None, reverse=False)</code>	对可迭代对象进行排序。
enumerate	<code>enumerate(iterable, start=0)</code>	枚举可迭代对象中的元素。
zip	<code>zip(*iterables, strict=False)</code>	并行迭代多个可迭代对象。
filter	<code>filter(function, iterable)</code>	按指定规则筛选可迭代对象中的元素。
map	<code>map(function, iterable, ...)</code>	将指定操作应用到可迭代对象中的每个元素。
all	<code>all(iterable)</code>	仅当可迭代对象中的每个元素的逻辑值检测结果都为True时才返回True。
any	<code>any(iterable)</code>	只要可迭代对象中有一个元素的逻辑值检测结果为True就返回True。
max	<code>max(iterable, *[, key, default])</code> <code>max(arg1, arg2, *args[, key])</code>	第一种用法返回可迭代对象中的最大元素。 第二种用法返回指定对象中的最大者。
min	<code>min(iterable, *[, key, default])</code> <code>min(arg1, arg2, *args[, key])</code>	第一种用法返回可迭代对象中的最小元素。 第二种用法返回指定对象中的最小者。
sum	<code>sum(iterable, /, start=0)</code>	从指定位置开始对可迭代对象的元素求和。

表IV-11. 调试

函数名	语法	说明
compile	<code>compile(source, filename, mode, flags=0, dont_inherit=False, optimize=-1)</code>	将指定代码编译成代码对象或AST对象。
breakpoint	<code>breakpoint(*args,**kws)</code>	将一个断点硬编码到脚本中。

表IV-12. 帮助

函数名	语法	说明
help	<code>help(object)</code> <code>help()</code>	第一种用法基于指定对象的文档和注解生成帮助页面并返回。 第二种用法启动交互式帮助系统。