13. 前方高能-内置函数一

一. 本节主要内容:

1. 内置函数

什么是内置函数? 就是python给你提供的. 拿来直接用的函数, 比如print., input等等. 截止到python版本3.6.2 python一共提供了68个内置函数. 他们就是python直接提供给我们的. 有一些我们已经用过了. 有一些还没有用过. 还有一些需要学完了面向对象才能继续学习的. 今天我们就认识一下python的内置函数.

abs()	dict()	help()	min()	setattr()
all()	dir()	hex()	next()	slice()
any()	divmod()	id()	object()	sorted()
ascii()	enumerate()	input()	oct()	staticmethod()
bin()	eval()	int()	open()	str()
bool()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	
delattr()	hash()	memoryview()	set()	

作用域相关:

locals() 返回当前作用域中的名字

globals()　返回全局作用域中的名字

迭代器相关:

range()　生成数据

next()　迭代器向下执行一次, 内部实际使用了__next__()方法返回迭代器的下一个项目

iter()　获取迭代器, 内部实际使用的是__iter__()方法来获取迭代器

字符串类型代码的执行

eval()　执行字符串类型的代码. 并返回最终结果

```
print(eval("2+2"))  # 4
n = 8
print(eval("2+n"))  # 10

def func():
    print(666)
eval("func()")  # 666
```

exec()　执行字符串类型的代码

```
exec("""
for i in range(10):
    print(i)
""")

exec("""
def func():
    print("我是周杰伦")
func()
""")
```

compile()　将字符串类型的代码变异. 代码对象能够通过exec语句来执行或者eval()进行求值

```
'''
    参数说明:
        1. resource 要执行的代码, 动态代码片段
        2. 文件名, 代码存放的文件名, 当传入了第一个参数的时候, 这个参数给空就可以了
        3. 模式, 取值有3个,
            1. exec: 一般放一些流程语句的时候
            2. eval: resource只存放一个求值表达式.
            3. single: resource存放的代码有交互的时候. mode应为single
'''
code1 = "for i in range(10): print(i)"
c1 = compile(code1, "", mode="exec")
exec(c1)

code2 = "1+2+3"
c2 = compile(code2, "", mode="eval")
a = eval(c2)
```

```
print(a)

code3 = "name = input('请输入你的名字:')"
c3 = compile(code3, "", mode="single")
exec(c3)
print(name)
```

有返回值的字符串形式的代码用eval(). 没有返回值的字符串形式的代码用exec(). 一般很少用到compile()

输入和输出相关:
  input() 获取用户输入的内容
  print() 打印输出

内存相关:
  hash() 获取到对象的哈希值(int, str, bool, tuple)
  id() 获取到对象的内存地址

文件操作相关:
  open() 用于口打开一个文件, 创建一个文件句柄

模块相关:
  __import__() 用于动态加载类和函数

帮助:
  help() 函数用于查看函数或模块用途的详细说明
调用相关:
  callable() 用于检查一个对象是否是可调用的. 如果返回True, object有可能调用失败, 但如果返回False. 那调用绝对不会成功
查看内置属性:
  dir() 查看对象的内置属性, 方法. 访问的是对象中的__dir__()方法
基础数据类型相关:
  数字相关:
    bool() 将给定的数据转换成bool值. 如果不给值. 返回False
    int() 将给定的数据转换成int值. 如果不给值, 返回0
    float() 将给定的数据转换成float值. 也就是小数
    complex() 创建一个复数. 第一个参数为实部, 第二个参数为虚部. 或者第一个参数直接用字符串来描述复数
  进制转换:
    bin() 将给的参数转换成二进制
    otc() 将给的参数转换成八进制
    hex() 将给的参数转换成十六进制

数学运算:

abs()　　返回绝对值

divmode()　返回商和余数

round()　　四舍五入

pow(a, b)　求a的b次幂, 如果有三个参数. 则求完次幂后对第三个数取余

sum()　求和

min()　求最小值

max()　求最大值

和数据结构相关:

列表和元组:

list()　　将一个可迭代对象转换成列表

tuple()　将一个可迭代对象转换成元组

reversed()　将一个序列翻转, 返回翻转序列的迭代器

slice()　列表的切片

```
st = "大家好, 我是麻花藤"
s = slice(1, 5, 2)
print(st[s])
```

字符串相关:

str()　　将数据转化成字符串

format()　与具体数据相关, 用于计算各种小数, 精算等

```
# 字符串
print(format('test', '<20'))    # 左对齐
print(format('test', '>20'))    # 右对齐
print(format('test', '^20'))    # 居中
# 数值
print(format(3, 'b'))   # 二进制
print(format(97, 'c'))   # 转换成unicode字符
print(format(11, 'd'))   # 十进制
print(format(11, 'o'))   # 八进制
print(format(11, 'x'))   # 十六进制(小写字母)
print(format(11, 'X'))   # 十六进制(大写字母)
print(format(11, 'n'))   # 和d一样
print(format(11))   # 和d一样
# 浮点数
print(format(123456789, 'e'))   # 科学计数法. 默认保留6位小数
print(format(123456789, '0.2e'))   # 科学计数法. 保留2位小数(小写)
print(format(123456789, '0.2E'))   # 科学计数法. 保留2位小数(大写)
print(format(1.23456789, 'f'))   # 小数点计数法. 保留6位小数
print(format(1.23456789, '0.2f'))   # 小数点计数法. 保留2位小数
print(format(1.23456789, '0.10f'))   # 小数点计数法. 保留10位小数
print(format(1.23456789e+10000, 'F'))   # 小数点计数法.
```

bytes() 把字符串转化成bytes类型

```
s = "你好"
bs = s.encode("UTF-8")
print(bs)
s1 = bs.decode("UTF-8")
print(s1)

bs = bytes(s, encoding="utf-8")      # 把字符串编码成UTF-8
print(bs)
```

bytearray() 返回一个新字节数组. 这个数字里的元素是可变的, 并且每个元素的值得范围是[0,256)

```
ret = bytearray('alex',encoding='utf-8')
print(ret[0])
print(ret)
```

memoryview() 查看bytes在内存中的情况

```
# 查看bytes字节在内存中的情况
s = memoryview("麻花藤".encode("utf-8"))
print(s)
```

ord() 输入字符找带字符编码的位置

chr() 输入位置数字找出对应的字符

ascii() 是ascii码中的返回该值 不是就返回\u...

```
# 找到对应字符的编码位置
print(ord('a'))
print(ord('中'))

# 找到对应编码位置的字符
print(chr(97))
print(chr(20013))

# 在ascii中就返回这个值. 如果不在就返回\u...
print(ascii('a'))
print(ascii('好'))
```

repr() 返回一个对象的string形式

```
# repr 就是原封不动的输出, 引号和转义字符都不起作用
print(repr('大家好,\n \t我叫周杰伦'))
print('大家好我叫周杰伦')

# %r  原封不动的写出来
name = 'taibai'
print('我叫%r' % name)
```

数据集合:

dict() 创建一个字典

set()　创建一个集合

frozenset()　创建一个冻结的集合. 冻结的集合不能进行添加和删除操作

其他相关:

len()　返回一个对象中的元素的个数

sorted()　对可迭代对象进行排序操作(讲完lamda后再讲这个)

enumerate()　获取集合的枚举对象

```
lst = ["alex", "wusir", "taibai"]
for index, el in enumerate(lst):
    print(str(index)+"==>"+el)
```

all()　可迭代对象中全部是True, 结果才是True

any()　可迭代对象中有一个是True, 结果就是True

```
print(all([1,2,True,0]))
print(any([1,'',0]))
```

zip()　函数用于将可迭代的对象作为参数, 将对象中对应的元素打包成一个个元组, 然后返回由这些元组组成的开了表. 如果各个迭代器的元素个数不一致, 则返回列表长度与最短的对象相同.

```
l1 = [1,2,3,]
l2 = ['a','b','c',5]
l3 = ('*','**',(1,2,3))
for i in zip(l1,l2,l3):
    print(i)
```

filter()　过滤(讲完lamda)

map()　会根据提供的函数对指定序列做映射(lamda)

参考资料:

https://www.processon.com/view/link/5b4ee15be4b0edb750de96ac