

暨南大学本科实验报告专用纸

课程名称：Python 程序设计实验

实验项目名称：语法与变量类型 指导教师：林聪

实验项目编号：实验 02 实验项目类型：验证性实验

实验地点：实 C302

学生姓名：陈旭天 学号：2021100733

学院：智能科学与工程学院 专业：人工智能

实验时间：2022 年 3 月 9 日下午--2022 年 3 月 9 日下午



一、实验目的

深化对 python 内部数据类型的理解，区分不同数据类型的变量的不同之处，同时上手实操一些可以鉴定变量数据类型的方法、函数。熟练使用一些数据特有的方法来操作数据。

二、实验原理

三、实验结果

初始化并测试：字符串，整数，浮点数，列表，字典

由图 1 声明变量，然后用 print () 函数和.isnumeric () 和.isdecimal () 方法判断 a 是整数，b 是浮点数

```
message="This is a string."
a=3
b=4/2
list1=["x","y","z"]
dictionary={"jen":["python','ruby']",
            "sarah":["c']"}
print(message)
a.isnumeric()
b.isdecimal()
```

```
PS C:\Users\36126\c语言> python -u "C:\Users\36126\AppData\Local\Temp\tempCodeRunnerFile.python"
This is a string.
Traceback (most recent call last):
  File "C:\Users\36126\AppData\Local\Temp\tempCodeRunnerFile.python", line 11, in <module>
    b.isdecimal()
AttributeError: 'float' object has no attribute 'isdecimal'
PS C:\Users\36126\c语言> python -u "C:\Users\36126\AppData\Local\Temp\tempCodeRunnerFile.python"
This is a string.
Traceback (most recent call last):
  File "C:\Users\36126\AppData\Local\Temp\tempCodeRunnerFile.python", line 10, in <module>
    a.isnumeric()
AttributeError: 'int' object has no attribute 'isnumeric'
```

熟悉字符串的内建函数，练习字符串的一些方法

练习使用 f“string”和.format () 方法

```

from site import check_enableusersite
message="    This is a string."
a=3
b=4/2
list1=["x","y","z"]
dictionary={"jen":["python','ruby']",
            "sarah":["c']}
first_name="chen"
last_name="xutian"
full_name=f"{first_name}{last_name}"
print(f"Hello,{full_name.title()}!")
print("Hello,{}".format(full_name))
print(message)
print(message.title())
print(message.split())
print(message.join(first_name))

```

Split 函数将字符串输出成一个可迭代对象

```

PS C:\Users\36126\c语言> python -u "C:\Users\36126\AppData\Local\Temp\tempCodeRunnerFile.python"
Hello,Chenxutian!
Hello,chenxutian!
    This is a string.
    This Is A String.
['This', 'is', 'a', 'string.']
c    This is a string.h    This is a string.e    This is a string.n

```

列表的.append()和.pop()方法使用

```

list1.append("w")
print(list1)
list1.pop()
print(list1)

```

```

PS C:\Users\36126\c语言> python -u "C:\Users\36126\AppData\Local\Temp\tempCodeRunnerFile.python"
['x', 'y', 'z', 'w']
['x', 'y', 'z']
PS C:\Users\36126\c语言>

```

然后，定义一个函数将可迭代对象 list1 的内容依次访问输出

```

def shuchu(list):
    n=len(list)
    for i in range(n):
        xinxi=f"My favorite character is {list[i].upper()}.\n"
        print(xinxi)
shuchu(list1)

```

输出如图所示：

```

PS C:\Users\36126\c语言> python -u "C:\Users\36126\AppData\Local\Temp\tempCodeRunnerFile.python"
This is a string.
My favorite character is X.

My favorite character is Y.

My favorite character is Z.

```

练习各种运算符的优先级

```
>>> 3+2
5
>>> 3+2*5
13
>>> 3+2/5
3.4
>>> 3+2%5
5
>>> (3+2)/5
1.0
>>> (3+2)%5
0
>>> 3+2==5
True
>>> 1&0==0
True
>>> 1|0==0
False
>>> x=2
>>> y=3
>>> x+y is 5
True
>>> list1=["a",3,["b","c"]]
>>> 3 in list1
True
>>> x or y is 2
2
>>> x and y is 2
False
>>> x!=y
True
```

优先级	运算符	描述
1	lambda	Lambda表达式
2	or	布尔“或”
3	and	布尔“与”
4	not x	布尔“非”
5	in, not in	成员测试
6	is, is not	同一性测试
7	<, <=, >, >=, !=, ==	比较
8		按位或
9	^	按位异或
10	&	按位与
11	<<, >>	移位
12	+, -	加法与减法
13	*, /, %	乘法、除法与取余
14	+x, -x	正负号
15	~x	按位翻转
16	**	指数
17	x.attribute	属性参考
18	x[index]	下标

熟悉 id (), eval (), type (), zip () 函数

```
>>> x=(2,3,5)
>>> y=[2,3,6]
>>> id(x)
46935057512992
>>> id(y)
46935057549064
```

Id()函数用于查看变量的存储地址

```
>>> n=81
>>> eval('n+5')
86
>>> eval('n*2')
162
>>> eval('n**2')
6561
>>> eval('pow(2,2)')
4
>>> eval('pow(n,n)')
386621969787156332734047587900743169602142130961783196218569342598075309373218614851925085428734706375011609800817940355
19670238407078788135931371782481
>>> █
```

type() 函数属于 [Python](#) 内置函数，通常用来查看某个变量的具体类型

```
1 string = "I love you."
2 a = 3
3 list1 = ["kira","python",
4          "jack","java"]
5 dictionary = {"a":1,"b":2,"c":3}
6
7 print(type(string))
8 print(type(a))
9 print(type(list1))
10 print(type(dictionary))
```

```
PS C:\Users\36126> python -u "c:\Users\36126\tempCodeRunnerFile.python"
<class 'str'>
<class 'int'>
<class 'list'>
<class 'dict'>
```

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列表

```
a = [1,1,1]
b = [2,3,4]
c = [6,7,8,9]
zipped = zip(a,b,c)
print(list(zip(a,b,c)))
```

```
PS C:\Users\36126> python -u "c:\Users\36126\tempCodeRunnerFile.python"
[(1, 2, 6), (1, 3, 7), (1, 4, 8)]
PS C:\Users\36126>
```

字典一般指普通的无序字典，order 字典指的是有序字典

遍历一个普通字典，返回的数据和定义字典时的字段顺序是不一致的。

有序字典可以按字典中元素的插入顺序来输出。

```

my_dict = dict()
my_dict["name"] = "lowman"
my_dict["age"] = 26
my_dict["girl"] = "Tailand"
my_dict["money"] = 80
my_dict["hourse"] = None

for key, value in my_dict.items():
    print(key, value)

```

```

PS C:\Users\36126> python -u "c:\Users\36126\tempCodeRunnerFile.python"
name lowman
age 26
girl Tailand
money 80
hourse None

```

```

import collections

my_order_dict = collections.OrderedDict()

my_order_dict["name"] = "lowman"
my_order_dict["age"] = 45
my_order_dict["money"] = 998
my_order_dict["hourse"] = None

for key, value in my_order_dict.items():
    print(key, value)

```

```

PS C:\Users\36126> python -u "c:\Users\36126\tempCodeRunnerFile.python"
name lowman
age 45
money 998
hourse None

```

.get()方法用于返回指定键的值

```

my_dict = dict()
my_dict["name"] = "lowman"
my_dict["age"] = 26
my_dict["girl"] = "Tailand"
my_dict["money"] = 80
my_dict["hourse"] = None

print(str(my_dict.get("age")))

```

```

PS C:\Users\36126> python -u "c:\Users\36126\tempCodeRunnerFile.python"
26

```

setdefault() 方法使用指定的键返回项目的值。
如果键不存在，则插入这个具有指定值的键。

```
my_dict = dict()
my_dict["name"] = "lowman"
my_dict["age"] = 26
my_dict["girl"] = "Tailand"
my_dict["money"] = 80
my_dict["hourse"] = None

# print(str(my_dict.get("age")))
x = my_dict.setdefault("girl")
print(x)
```

```
PS C:\Users\36126> python -u "c:\Users\36126\tempCodeRunnerFile.python"
Tailand
_
```

实验总结

Python 作为一门解释型语言其对变量的类型设定有着严格的要求，所以有必要去了解不同数据类型的变量的区别之处。因为不同的数据类型修改、访问、调用的方法也不同，所以熟悉数据和变量类型有助于写出更稳定和简洁的代码。