

暨南大学本科实验报告专用纸

课程名称 python 实验 成绩评定
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 实验项目类型 实验地点
学生姓名 王志涛 学号 2021102259
学院 智能科学与工程 系 专业 人工智能
实验时间 2022 年 3 月 9 日 下 午 ~ 4 月 20 日 下 午

一、 实验目的

通过自拟案例，初始化并测试各类的变量类型数据。练习常用数据操作与语言语法。熟练掌握运算符与其优先级。熟悉常用的函数

二、 实验原理

在 python 中掌握函数和熟练掌握运算符与其优先级

三、 实验过程

1. 初始化并测试：字符串，整数，浮点数，列表，字典，如下代码：

```
a="hello world"
b=10
c=1.00
d=[1,2]
e={1,2}
print(a)
print(str(b))
print(float(c))
print(d)
print(e)
```

输出：

```
hello world
10
1.0
[1, 2]
{1, 2}
```

2.练习内置函数

(1) `.split()` 可以将字符串分割，具体用法如以下代码：

```
a="hello world ohloveyou"
print(a.split())#空格分行
print(a.split('w'))#w分行
print(a.split('o',2))#o分割两次
print(a.split('o')[2])#取分割后的排列第二的字符串
```

输出为:

```
['hello', 'world', 'ohloveyou']
['hello ', 'orld ohloveyou']
['hell', ' w', 'rld ohloveyou']
rld
```

(2) `.title()`可以将字符串的首字母变成大写，具体用法如下代码：

```
b="hello world"
print(b.title())
```

输出为:

```
Hello World
```

(3) `.join()`的用法是在`.`前面加要插入的字符串，括号里面加入原来的字符串，具体用法如下代码：

```
c=', '
set="hello world"
print(c.join(set))
```

输出为:

```
h,e,l,l,o, ,w,o,r,l,d
```

(4) 列表的`.append()`具体用法如下代码：

```
d=['wo','ai','ni','ya']#先创建一个列表
d.append('123')#在列表加一个'123'
print(d)#打印d
```

输出为:

```
['wo', 'ai', 'ni', 'ya', '123']
```

(5) 列表的`.pop()`具体用法如下代码：

```
liste=['li','yan','fang']
lishei=liste.pop(2)#删除列表的第三个项
print("删除的项: "+lishei)
print("剩下的列表: "+str(liste))
```

输出：

```
删除的项: fang
剩下的列表: ['li', 'yan']
```

4. 练习使用各种运算符，熟悉优先级

(1) 算术运算符：

```
a=10
b=20
c=30
print(a*b)
print(a+b)
print(a-b)
print(a/b)
print(a%b)
print(a**2)
print(a//b)
```

输出：

```
200
30
-10
0.5
10
100
0
```

(2) 比较运算符

```
a=10
b=20
if(a==b):
    print("a等于b")
else:
    print("a不等于b")
if(a>b):
    print("a比b大")
else:
    print("a比b小")
if(a!=b):
    print("a不等于b")
else:
    print("a等于b")
if(a<b):
    print("b比a大")
else:
    print("b比a小")
if(a>=b):
    print("a大于等于b")
else:
    print("a小于b")
```

输出为:

```
a不等于b
a比b小
a不等于b
b比a大
a小于b
```

(3) 赋值运算符:

```
1  a=10
2  b=20
3  d=e=f=g=h=i=j=7d
4  c=a+b
5  d+=a
6  print(str(d))
7  e-=b
8  print(str(e))
9  f*=a
10 print(str(f))
11 g/=a
12 print(str(g))
13 h%=a
14 print(str(h))
15 i**=b
16 print(str(i))
17 j//=a
18 print(str(j))
```

输入为:

```
80
50
700
7.0
0
797922662976120010000000000000000000
7
```

(4) 位运算符:

```
a = 60          # 60 = 0011 1100
b = 13          # 13 = 0000 1101
c = 0
c = a & b       # 12 = 0000 1100
print ("1 - c 的值为: ", c)
c = a | b       # 61 = 0011 1101
print ("2 - c 的值为: ", c)
c = a ^ b       # 49 = 0011 0001
print ("3 - c 的值为: ", c)
c = ~a          # -61 = 1100 0011
print ("4 - c 的值为: ", c)
c = a << 2       # 240 = 1111 0000
print ("5 - c 的值为: ", c)
c = a >> 2       # 15 = 0000 1111
print ("6 - c 的值为: ", c)
```

输出为：

```
1 - c 的值为: 12
2 - c 的值为: 61
3 - c 的值为: 49
4 - c 的值为: -61
5 - c 的值为: 240
6 - c 的值为: 15
```

(5) 逻辑运算符：

```
a=60
b=40
print(a and b)
print(a or b)
print(not(a and b))
```

输出：

```
40
60
False
```

(6) 成员运算符：

```
a=10
b=20
c=[10,52,63,87]
if(a in c):
    print("a在c里面")
else:
    print("a不在c里面")
if(b not in c):
    print("b不在c里面")
else:
    print("b在c里面")
```

输出：

```
a在c里面
b不在c里面
```

(7) 熟悉优先级：查阅资料得：

以下表格列出了从最高到最低优先级的所有运算符：

运算符	描述
**	指数 (最高优先级)
~ + -	按位翻转, 一元加号和减号 (最后两个的方法名为 +@ 和 -@)
* / % //	乘, 除, 求余数和取整除
+ -	加法减法
>> <<	右移, 左移运算符
&	位 'AND'
^	位运算符
<= < > >=	比较运算符
== !=	等于运算符
= %= /= //= -= += *= **=	赋值运算符
is is not	身份运算符
in not in	成员运算符
not and or	逻辑运算符

4. 练习使用 id(),eval(),type(),zip()等函数.

(1) id () 函数用于获取对象得内存地址

```
a="hello world"
print(id(a))
```

输出为：

```
1749082111024
```

(2) eval()函数用来执行一个字符串表达式，并返回表达式的值：

```
print(eval('3*4'))
```

输出：

```
12
```

(3) `type()`函数用来判断类型:

```
a=10 #int类型
b="hello world"#str类型
c=1.0#float类型
print(type(a))
print(type(b))
print(type(c))
```

输出:

```
<class 'int'>
<class 'str'>
<class 'float'>
```

(4) `zip()` 函数具体用法如下:

```
a=[1,2,3]
b=[12,14,18]
c=[120,125,102,1587]
print(zip(a,b))
print(list(zip(a,b)))#打包为元组的列表
print(list(zip(a,c)))#长度不一时以短的为标准
```

输出为:

```
<zip object at 0x000001C6BA8EEF08>
[(1, 12), (2, 14), (3, 18)]
[(1, 120), (2, 125), (3, 102)]
```

5. 练习 f-string 和 `str.format()`方法

(1) **f-string** 格式化字符串以 **f** 开头, 后面跟着字符串, 字符串中的表达式用大括号`{}`包起来, 它会将变量或表达式计算后的值替换进去。

```
a='world'
print(f'hello {a}')#将a放进里面
print(f'{12356+125896}')#里面执行语句
```

输出

```
hello world
138252
```

6.

(1)实验比较字典和 OrderedDict 区别

```
from collections import OrderedDict#普通字典是不会排序的
d={}
d['a']='A'
d['b']='B'
d['d']='D'
for v,i in d.items():
    print(v,i)

a=OrderedDict()#OrderedDict实现了对字典对象中元素的排序
# import collections
# a=collections.OrderedDict()
a['a']='A'
a['d']='D'
a['f']='F'
a['c']='C'
a['b']='B'
a=sorted(a.items(),key=lambda x:x[0])
print(a)
```

输出：

```
a A
b B
d D
[('a', 'A'), ('b', 'Q'), ('c', 'C'), ('d', 'D'), ('f', 'F')]
```

(2) 了解字典的.get()和.setdefault()方法

```
wang={
    "color":"yellow",
    "year":"2005",
    "happy":"zhitao"
}
m=wang.get("egg",'black')
print(m)
print(wang)
n=wang.setdefault("egg","black")
print(n)
print(wang)
```

.get()和.setdefault()都能访问字典的值，若字典不存在这个值，这两个方法都可以有一个替补的值补上，不同的是，.setfault()补上的值会在原字典上补上，即修改了原字典。

输出:

```
{'color': 'yellow', 'year': '2005', 'happy': 'zhitao'}  
black  
{'color': 'yellow', 'year': '2005', 'happy': 'zhitao', 'egg': 'black'}
```

7.简易代码:

```
import random  
num=random.randint(1,100)  
for a in range(8):  
    b=int(input(f'请输入一个数:'))  
    if b>num:  
        print('你猜大了')  
        print('请继续')  
    elif b < num:  
        print('你猜小了')  
        print('请继续')  
    else:  
        print('你猜对了! ')  
        break  
else:  
    print(f'你的机会已经用完了: {num}')
```

```
print('游戏结束!!')
```

调试:

```
请输入一个数:40  
你猜大了  
请继续  
请输入一个数:30  
你猜大了  
请继续  
请输入一个数:20  
你猜大了  
请继续  
请输入一个数:10  
你猜小了  
请继续  
请输入一个数:15  
你猜小了  
请继续  
请输入一个数:18  
你猜小了  
请继续  
请输入一个数:19  
你猜对了!  
游戏结束!!
```

四、 实验结果

五、 实验总结

回顾了理论课的函数相关知识，熟悉掌握函数的用法及相似函数的区别。