暨南大学本科实验报告专用纸

课程名称 python 程序设计实验 成绩评定

实验项目名称 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 4 月 6 日 上 午～ 4 月 24 日 午

一、 实验目的

通过自拟案例，初始化相关变量，设置简易的代码框架，测试使用 while 循环, for 循环, if 等选择结构和相关的关键字与函数.

二、 实验原理

在 python 中能实现代码的运行。

三、 实验过程

1. 初始化变量并测试：测试使用 while 和 for 循环，实现 for/while 循环的(两层)嵌套使用。

```
i=1
while i<9:
    print(i)
    i+=1
for x in range(1,9):
    print(x)

for h in range(1,5):
    while h<4:
        print(h)
        h+=1
```

2.break 与 continue，break 直接结束循环，break 是跳过当前循环，代码如下：

```
i=1
h=1
while i<9:
    i+=1
    if(i==3):
        continue
    print(i)
while h<9:
    h+=1
    if(h==8):
        break
    print(h)
```

实现设置"while True:"循环，设置退出循环的测试条件

```
d={
    "wang":"123456",
    "zhi":"123456789",
}
while True:
    name=input('输入用户名')
    if name in d:
        break
    else:
        print("没有你这个人")
        continue
while True:
    passwork=input("输入密码")
    if d[name]==passwork:
        print("你的密码正确")
        break
    else:
        print("你的密码错误")
        continue
```

3. 练习使用 enumerate(), zip(), range(), input()等函数

Enumerate()函数会在循环前加个索引值：

```
d=["wang","zhi","tao","shi","shuaige"]
for i,h in enumerate(d):
    print(i,h)
```

zip()将列表打包为元组的列表：

```
d=[4,5,6,8,9,12]
h=[1,2,3,4,5,6]
print(list(zip(d,h)))
```

Range()函数用于：

```
for i in range(20):
    print(i)
```

input()函数：

```
d={
    "wang":"123456",
    "zhi":"123456789",
}
while True:
    name=input('输入用户名')
    if name in d:
        break
    else:
        print("没有你这个人")
        continue
while True:
    password=input("输入密码")
    if d[name]==password:
        print("你的密码正确")
        break
    else:
        print("你的密码错误")
        continue
```

实现分别用 **enumerate()**和 **range()**，在循环中打印出"目前迭代第 **i** 个元素"字符串，字符串中"**i**"为当前迭代元素的索引值或迭代序号(字典没有索引).

```
for i in range(1,10):
    print("目前迭代到"+str(i)+"个元素")
```

```
h=[1,2,3,4,5,6]
for i,t in enumerate(h):
    print("目前迭代到"+str(i+1)+"个元素")
```

4. 测试使用 **if** 结构, **if-else** 结构, **if-elif-else** 结构

If 结构：

```
for k in range(2,80):
    print(k)
    k+=1
    if(k==5):
        break
```

if-else 结构：

```
h=input("输入整数")
if h<"30":
    print("这个数字还是太小了")
else:
    print("差不多够了")
```

if-elif-else 结构：

```
a = 200
b = 66
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
else:
  print("a is greater than b")
```

4.1 实现一个具有 if-else 结构的 Leaky ReLU 函数，命名和参数为 "leaky_relu_v1(x, a=0.2)"，这里 a 为 x 为负值时的系数.：

```
def leaky_relu_v1(x):
    a=0.2
    if(x<=0):
        y=a*x
    else:
        y=x
    return y
h=leaky_relu_v1(-10)
print(h)
```

4.2 实现一个没有 if-else 结构的 Leaky ReLU 函数，命名和参数为 "leaky_relu_v2(x, a=0.2)".

```
def leaky_relu_v2(x):
    a=0.2
    if(x<=0):
        y=a*x
    if(x>0):
        y=x
    return y
h=leaky_relu_v2(-12)
print(h)
```

4.3 通过 for 循环计算-10000000 到 10000000 的 Leaky ReLU 函数值,并记录总计算时间(这里可用 time 模块中的 time()函数).

```
import time
a=time.time()
def leaky_relu_v2(x):
    a=0.2
    if(x<=0):
        return a*x
    else:
        return x
for i in range(-10000000,10000000):
    leaky_relu_v2(i)
b=time.time()
print(b-a)
```

输出：`1.9640982151031494`

4.4：第一个： `1.9640982151031494`

第二个: `3.6567680835723877`

思考题：

1. 为什么循环结构中不应改变正在迭代的序列的长度?

迭代开始的时候，迭代的对象是原来的，当序列改变时，迭代的对象没有同步变化，所以迭代继续的时候，会找不到继续迭代的对象。

```
a="hello world"
for i in a:
    print(i)
    a="hello world nihao"
```

上面输出的仍然是 **hello world**。

2. 为什么程序设计语言中需要有循环结构和选择结构**?**

循环结构可以加快代码的速率，选择结构可以让代码按照我们的意愿运行。

四、实验结果

五、　实验总结