

全员debug





1. 探索新知

1.1

小试牛刀



试着根据报错信息，将各练习的错误代码修改正确吧！

练习1:

```
print("Hello, world!")
```

练习2:

```
a = 0
for i in range:
    a += 1
print(a)
```

练习3:

```
a = 0
for i in range:
    a += 1
print(a)
```

练习4:

```
num_str = "hello"
num = int(num_str)
```

1.1

小试牛刀

练习5:

```
my_list = [1, 2, 3]
print(my_list[3])
```

练习6:

```
my_dict = {'a': 1, 'b': 2, 'c': 3}
print(my_dict['d'])
```

练习7:

```
with open('missing_file.txt', 'r') as file:
    content = file.read()
    print(content)
```

练习8:

```
def ave (nums):
    total = sum(nums)
    average = total / len(nums)
    return average
result = ave ([])
print(result)
```

练习9:

```
str1 = 'hello'
str2 = 123
result = str1 + str2
print(result)
```

练习10:

```
from turtle import randint
a = randint(1,100)
print(a)
```

拓展 - split() 函数

split()函数用来将一个字符串根据指定的分隔符进行分割，并返回分割后的子字符串列表。

- 默认情况下，split() 函数使用空格作为分隔符

```
s = "hello world"  
result = s.split()  
print(result) # 输出 ['hello', 'world']
```

- split()函数还可以接受一个参数，用来指定分隔符，例如逗号、分号等

```
s = "hello world"  
result = s.split()  
print(result) # 输出 ['hello', 'world']
```

1.2

全员 debug

这里有一个代码，代码中有10个错误，请运行作品，挨个找到并解决所有的问题，使代码正确运行。



```
# 进行累加操作
def sum_list(lst):
    sum = 0
    for num in lst
        sum += num
    return sum

# 找出列表中的最大值
def max_list(lst):
    return max(lst)

# 把输入的字符串转换为整数列表
def str_to_list(s):
    for i in s.split(","):
        s1.append(i)
    return s1

# 代码主体部分
while True:
    print("1: 求和")
    print("2: 求最大值")
    print("3: 退出")
    choice = int(input("请输入操作编号: "))
    # 根据用户的选择执行不同任务
    if choice = 1:
        s = input("请输入一串以空格分隔的数字: ")
        lst = str_to_list(s)
        print("和为", sum_list(lst))
    elif choice = 2:
        s = input("请输入一串以空格分隔的数字: ")
        lst = str_to_list(s)
        print("最大值为", mx_list(lst))
    elif choice = 3:
        break
    else:
        print("输入无效")
```



2. 强化练习

1. 初学者常采用异常处理解决程序中出现的意外情况，下面语句的执行结果是？（ ）

- A. 我爱中国 2023
程序结束
- B. 错误
程序结束
- C. 我爱中国 2023
- D. 错误

```
try:  
    s1 = '我爱中国'  
    s2 = 2023  
    s = s1 + s2  
    print(s)  
except:  
    print('错误')  
finally:  
    print('程序结束')
```

2. 当发生异常时，下列描述正确的是？（ ）

- A. 需要捕获异常，然后进行相应的处理
- B. 需要把可能发生错误的语句放在except模块里
- C. 需要把处理异常的语句放在try模块里
- D. 需要为每一个try模块设定且只能设定一个except模块

3. 异常是指？（ ）

- A. 程序设计时的错误
- B. 程序编写时的错误
- C. 程序编译时的错误
- D. 程序运行时的错误



2. 强化练习

4. 请看下列代码:

代码中包含几处错误? ()

- A. 1
- B. 2
- C. 3
- D. 4

```
x = str(input('请输入购买数量:'))
if x < 5:
    print('不打折')
else if x < 10:
    print('打 8 折')
elif x < 15:
    print('打 7 折')
else:
    print('打 5 折')
```

5. 在Python异常处理结构中, 如果程序执行没有遇到异常(错误), 不会执行什么关键字后面的语句体? ()

- A. try
- B. except
- C. else
- D. finally

3. 术语箱

split 使分离，劈开

4. 课后挑战

请找出下列代码中的3处错误，并改正

```
def check(s):  
    if "!" in s  
        print("字符串中含有感叹号!")  
    else:  
        print("字符串中没有感叹号!")  
string = "Hello world"  
check(s)
```

① _____

② _____

③ _____