

一 常用模块

1.1 time

```
import time

print(time.time())    # 2023/11/4 19:33:00

print(time.localtime())

# time.sleep()    # 接收的是一个整数 也可以接收浮点数

while True:
    # time.sleep(1)
    print('hello world')

time.strftime()  有两个参数，第一个参数是格式，第二个参数是要转换的时间

localtime()  转换格式
%Y  年
%m  月
%d  日
%H  时
%M  分
%S  秒

print(time.strftime('%Y/%m/%d %H:%M:%S',time.localtime()))
```

1.2 random

```
import random
需要导包
生成一个大于0，小于1的随机数
a = random.random()
print(a)

random.randint(start,end)  # 左闭右闭区间
start ==> 起点
end ==> 终点
a = random.randint(1,100)  # 在1~100之间随机一个整数
print(a)

随机生成元素
a = random.choice(['剪刀','石头','布'])
print(a)
```

随机生成多个元素，参数大于指定的列表元素数量也不会报错

```
a = random.choices(['剪刀', '石头', '布'], k=2)
print(a)
```

```
a = random.choices(['剪刀', '石头', '布'], k=4)
print(a)
```

随机生成指定范围的小数

```
a = random.uniform(1,10)
print(a)
```

打乱序列类型的顺序

```
li = [1,2,3,4,5,6,7,8,9,10]
random.shuffle(li)
print(li)
```

不可变类型不可以被打乱顺序

```
s = (123,234,345)
random.shuffle(s)
print(s)
```

4位数字的随机验证码

```
def yzm():
    s = ''
    for i in range(4):
        s += str(random.randint(0,9))
    return s
print(yzm())
```

1.3 json

json是文件的存储形式

json是一种兼容性很强的文件格式

跨语言跨平台

序列数据：机器码(底层的0和1) 用来存储和传输的

反序列数据：程序员写的看的代码 使用和观看

```
json_data = json.dumps([1,2, '左asdasdasda'])
print(json_data)
print(type(json_data))
```

序列化之后它存储的是一个字符串类型

```
json_data = json.dumps([1,2, '左手asdasdasda'])
data = json.loads(json_data) # 反序列化
print(json_data)
print(data)
print(type(data))
```

二 文件操作

操作文件步骤:

1. 打开文件
2. 对文件进行操作, 保存
3. 关闭文件

2.1 打开

```
open(file, mode='r', buffering=-1, encoding=None, errors=None, newline=None,
closefd=True, opener=None)
```

file: 要打开的文件的名字(或者路径)

```
fileName = '../day_14/demo.txt' # 相对路径
demo = open(fileName)
print(demo)
```

python的绝对路径快捷键: ctrl + shift + c

.. 表示上一级

相对路径 : ../day_14/demo.txt

在路径前面加一个r作用是规避转义

```
r'../day_14/demo.txt'
```

2.2 关闭

read 读取

demo.close() 调用关闭方法

```
fileName = r'demo.txt' # 相对路径
demo = open(fileName)
content = demo.read()
print(content)
print(type(content))
```

with open ... as 语句

```
fileName = r'demo.txt' # 相对路径
with open(fileName) as f:
    print(f.read())
```

注意 f只能在with当中使用, with结束了就自动close关闭

2.3 读取

我们在调用open()来打开文件的时候可以将文件分成两类

1. 纯文本文件(使用utf-8等编码编写的文本文件)
2. 二进制文件(音乐, 视频)

```
fileName = r'demo1.txt' # 相对路径
with open(fileName, encoding='utf-8') as f:
    print(f.read())
```

read有个参数，用来指定要读取的字符数量，默认值为-1就是读取文件中所有的内容

```
fileName = r'demo1.txt' # 相对路径
with open(fileName,encoding='utf-8') as f:
    content = f.read(4)
    content = f.read(4)
    content = f.read(4)
    content = f.read(4)
    print(content)
```

跟着上次的结果继续往下面读取，如果最后读取完了，就会返回一个空字符串

```
fileName = r'demo1.txt' # 相对路径
# 读取大文件的方式
with open(fileName,encoding='utf-8') as f:
    # 定义变量，指定每次读取的大小
    c = 3
    # 创建循环读取文件内容
    while 1:
        # 读取指定大小的内容
        content = f.read(c)
        # 检测内容是否为空
        if content == '':
            # 读取完毕，退出循环
            break
        print(content,end='')
```

```
# 或者
fileName = r'demo1.txt' # 相对路径
# 读取大文件的方式
with open(fileName,encoding='utf-8') as f:
    # 用来存储
    all_content = ''
    # 定义变量，指定每次读取的大小
    c = 3
    # 创建循环读取文件内容
    while 1:
        # 读取指定大小的内容
        content = f.read(c)
        # 检测内容是否为空
        if content == '':
            # 读取完毕，退出循环
            break
        all_content+=content
    print(all_content)
```

作业

制作一个跟电脑对战的剪刀锤子布小游戏（电脑随机出招），进入游戏时候提示当前时间,并需要登录密码。

密码规则：

总共是11位，全部由数字组成，第1位是1，第二位可以是数字3458其中之一，后面9位任意数字

例如：18601012345、13912367890则满足。

可不用函数

1.执行程序，进行友好的提示，需要提示当前时间

2.判断密码正确性

3.密码错误则重新输入

4.密码正确进入游戏循环，游戏具有友好的提示

5.出招处理胜负！0退出游戏