

入门 | 30个常用python实现

1、冒泡排序

```
lis = [56,12,1,8,354,10,100,34,56,7,23,456,234,-58]

def sortport():
    for i in range(len(lis)-1):
        for j in range(len(lis)-1-i):
            if lis[j] > lis[j+1]:
                lis[j],lis[j+1] = lis[j+1],lis[j]
    return lis
```

2、计算x的n次方的方法

```
def power(x, n):
    s = 1
    while n > 0:
        n = n - 1
        s = s * x
    return s
```

3、计算 $a*a + b*b + c*c + \dots$

```
def calc(*numbers):
    sum = 0
    for n in numbers:
        sum = sum + n * n
    return sum
```

4、计算阶乘 $n!$

```
def fac():
    num = int(input("请输入一个数字: "))
    factorial = 1

# 查看数字是负数, 0 或 正数
    if num < 0:
        print("抱歉, 负数没有阶乘")
    elif num == 0:
        print("0 的阶乘为 1")
    else:
        for i in range(1, num + 1):
            factorial = factorial * i
        print("%d 的阶乘为 %d" % (num, factorial))
```

```
def factorial(n):
    result = n
    for i in range(1, n):
        result *= i
    return result
```

```
def fact(n):
    if n==1:
        return 1
    return n * fact(n - 1)
```

5、列出当前目录下的所有文件和目录名

```
[d for d in os.listdir('.')]
```

6、把一个list中所有的字符串变成小写:

```
L = ['Hello', 'World', 'IBM', 'Apple']
[s.lower() for s in L]
```

7、输出某个路径下的所有文件和文件夹的路径

```
def print_dir():
    filepath = input("请输入一个路径: ")
    if filepath == "":
        print("请输入正确的路径")
    else:
        for i in os.listdir(filepath):
            #获取目录中的文件及子目录列表
            print(os.path.join(filepath,i)) #把路径组合起来

print(print_dir())
```

8、输出某个路径及其子目录下的所有文件路径

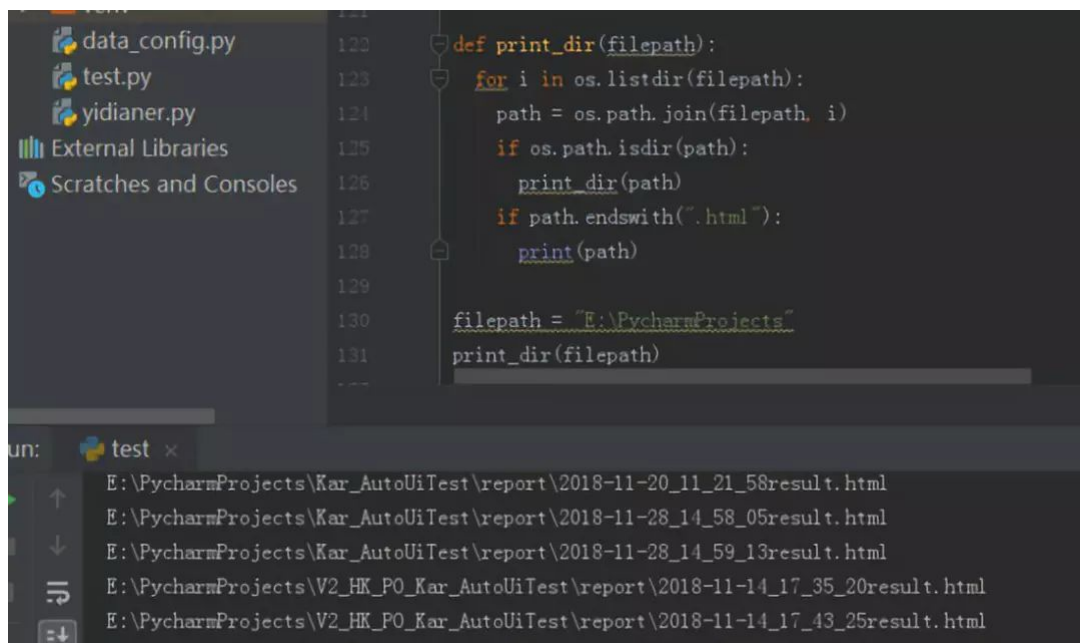
```
def show_dir(filepath):
    for i in os.listdir(filepath):
        path = (os.path.join(filepath, i))
        print(path)
        if os.path.isdir(path):      #isdir()判断是否是目录
            show_dir(path)          #如果是目录，使用递归方法

filepath = "C:\Program Files\Internet Explorer"
show_dir(filepath)
```

9、输出某个路径及其子目录下所有以.html为后缀的文件

```
def print_dir(filepath):
    for i in os.listdir(filepath):
        path = os.path.join(filepath, i)
        if os.path.isdir(path):
            print_dir(path)
        if path.endswith(".html"):
            print(path)

filepath = "E:\PycharmProjects"
print_dir(filepath)
```



10、把原字典的键值对颠倒并生产新的字典

```
dict1 = {"A": "a", "B": "b", "C": "c"}
dict2 = {y: x for x, y in dict1.items()}
print(dict2)
```

```
{ 'a': 'A', 'b': 'B', 'c': 'C' }
```

11、打印九九乘法表

```
for i in range(1, 10):
    for j in range(1, i+1):
        # print('{ }x{ }={ } \t'.format(j, i, i*j), end='')
        print('%d x %d = %d \t'%(i, j, i*j),end='')
    print()
```

通过指定end参数的值，可以取消在末尾输出回车符，实现不换行。

12、替换列表中所有的3为3a

```
num = ["harden", "lampard", 3, 34, 45, 56, 76, 87, 78, 45, 3, 3, 3, 87686, 98, 76]
# print(num.count(3))
# print(num.index(3))
for i in range(num.count(3)):      #获取3出现的次数
    ele_index = num.index(3)      #获取首次3出现的坐标
    num[ele_index]="3a"          #修改3为3a
    print(num)
```

13、打印每个名字

```
L = ["James", "Meng", "Xin"]
for i in range(len(L)):
    print("Hello,%s"%L[i])
```

** 善于使用 rang()，会使问题变得简单

14、合并去重

```
list1 = [2, 3, 8, 4, 9, 5, 6]
list2 = [5, 6, 10, 17, 11, 2]

list3 = list1 + list2
print(list3)                # 不去重只进行两个列表的组合
print(set(list3))           # 去重,类型为set需要转换成list
print(list(set(list3)))
```

15、随机生成验证码的两种方式

```

import random
list1=[]
for i in range(65,91):
    list1.append(chr(i))    #通过for循环遍历ascii追加到空列表中
for j in range(97,123):
    list1.append(chr(j))
for k in range(48,58):
    list1.append(chr(k))
ma = random.sample(list1,6)
print(ma)                  #获取到的为列表
ma = ''.join(ma)           #将列表转化为字符串
print(ma)

```

```

import random,string
str1 = "0123456789"
str2 = string.ascii_letters    # string.ascii_letters 包含所有字母（大写或小写）的字符串
str3 = str1+str2
ma1 = random.sample(str3,6)    #多个字符中选取特定数量的字符
ma1 = ''.join(ma1)             #使用join拼接转换为字符串
print(ma1)                     #通过引入string模块和random模块使用现有的方法

```

```

#随机数字小游戏
import random
i = 1
a = random.randint(0,100)
b = int( input('请输入0-100中的一个数字\n然后查看是否与电脑一样：'))
while a != b:
    if a > b:
        print('你第%d输入的数字小于电脑随机数字'%i)
        b = int(input('请再次输入数字：'))
    else:
        print('你第%d输入的数字大于电脑随机数字'%i)
        b = int(input('请再次输入数字：'))
    i+=1
else:
    print('恭喜你，你第%d次输入的数字与电脑的随机数字%d一样'%(i,b))

```

```
test_case.py
test_return.py
venv
data_config.py
test.py
yidianer.py
Internal Libraries
atches and Consoles

204 #随机数字小游戏
205 import random
206 i = 1
207 a = random.randint(0,100)
208 b = int(input('请输入0-100中的一个数字\n然后查看是否与电脑一样:'))
209 while a != b:
210     if a > b:
211         print('你第%d输入的数字小于电脑随机数字'%i)
212         b = int(input('请再次输入数字:'))
213     else:
214         print('你第%d输入的数字大于电脑随机数字'%i)
215         b = int(input('请再次输入数字:'))
216     i+=1
217 else:
218     print('恭喜你,你第%d次输入的数字与电脑的随机数字%d一样'%(i,b))

while a != b : if a > b

test x
D:\Python36\python.exe E:/PycharmProjects/xiezhu_V3/test.py
请输入0-100中的一个数字
然后查看是否与电脑一样: 8
你第1输入的数字小于电脑随机数字
请再次输入数字: 89
你第2输入的数字大于电脑随机数字
请再次输入数字: 50
你第3输入的数字大于电脑随机数字
请再次输入数字: 39
你第4输入的数字大于电脑随机数字
请再次输入数字: 23
你第5输入的数字大于电脑随机数字
请再次输入数字: 16
恭喜你,你第6次输入的数字与电脑的随机数字16一样
```

16、计算平方根

```
num = float(input('请输入一个数字: '))
num_sqrt = num ** 0.5
print(' %.2f 的平方根为 %.2f'%(num ,num_sqrt))
```

17、判断字符串是否只由数字组成

```
def is_number(s):
    try:
        float(s)
        return True
    except ValueError:
        pass

    try:
        import unicodedata
        unicodedata.numeric(s)
        return True
    except (TypeError, ValueError):
        pass

    return False
```



```
print(chri.isdigit())      #检测字符串是否只由数字组成
print(chri.isnumeric())    #检测字符串是否只由数字组成，这种方法是只针对unicode对象
```

18、判断奇偶数

```
num = int(input("输入一个数字: "))
if (num % 2) == 0:
    print("{0} 是偶数".format(num))
else:
    print("{0} 是奇数".format(num))
```

```
while True:
    try:
        num = int(input('输入一个整数: '))    #判断输入是否为整数
    except ValueError:                          #不是纯数字需要重新输入
        print("输入的不是整数!")
        continue
    if num % 2 == 0:
        print('偶数')
    else:
        print('奇数')
    break
```

19、判断闰年

```
year = int(input("输入一个年份: "))
if (year % 4) == 0:
    if (year % 100) == 0:
        if (year % 400) == 0:
            print("{0} 是闰年".format(year))    # 整百年能被400整除的是闰年
        else:
            print("{0} 不是闰年".format(year))
    else:
        print("{0} 是闰年".format(year))        # 非整百年能被4整除的为闰年
else:
    print("{0} 不是闰年".format(year))
```

```
year = int(input("请输入一个年份: "))
if (year % 4) == 0 and (year % 100) != 0 or (year % 400) == 0:
    print("{0}是闰年".format(year))
else:
    print("{0}不是闰年".format(year))
```

```
import calendar

year = int(input("请输入年份: "))
check_year=calendar.isleap(year)
if check_year == True:
    print ("%d是闰年"% year)
else:
    print ("%d是平年"% year)
```

20、获取最大值

```
N = int(input('输入需要对比大小数字的个数: '))
print("请输入需要对比的数字: ")
num = []
for i in range(1,N+1):
    temp = int(input('输入第 %d 个数字:' % i))
    num.append (temp)

print('您输入的数字为: ',num)
print('最大值为: ',max(num))
```

```
N = int(input('输入需要对比大小数字的个数: \n'))

num = [ int(input('请输入第 %d 个对比数字: \n'%i))for i in range(1,N+1)]

print('您输入的数字为:',num)
print('最大值为: ',max(num))
```

21、斐波那契数列

斐波那契数列指的是这样一个数列 0, 1, 1, 2, 3, 5, 8, 13; 特别指出：第0项是0，第1项是第一个1。从第三项开始，每一项都等于前两项之和。

```
# 判断输入的值是否合法
if nterms <= 0:
    print("请输入一个正整数。")
elif nterms == 1:
    print("斐波那契数列: ")
    print(n1)
else:
    print("斐波那契数列: ")
    print(n1, ", ", n2, end=" , ")
    while count < nterms:
        nth = n1 + n2
        print(n1+n2, end=" , ")
        # 更新值
        n1 = n2
        n2 = nth
        count += 1
```


22、十进制转二进制、八进制、十六进制

```
# 获取输入十进制数
dec = int(input("输入数字: "))

print("十进制数为: ", dec)
print("转换为二进制为: ", bin(dec))
print("转换为八进制为: ", oct(dec))
print("转换为十六进制为: ", hex(dec))
```

23、最大公约数

```
def hcf(x, y):
    """该函数返回两个数的最大公约数"""

    # 获取最小值
    if x > y:
        smaller = y
    else:
        smaller = x

    for i in range(1, smaller + 1):
        if ((x % i == 0) and (y % i == 0)):
            hcf = i

    return hcf

# 用户输入两个数字
num1 = int(input("输入第一个数字: "))
num2 = int(input("输入第二个数字: "))

print(num1, "和", num2, "的最大公约数为", hcf(num1, num2))
```

23、最小公倍数

```

# 定义函数
def lcm(x, y):

    # 获取最大的数
    if x > y:
        greater = x
    else:
        greater = y

    while(True):
        if((greater % x == 0) and (greater % y == 0)):
            lcm = greater
            break
        greater += 1

    return lcm

# 获取用户输入
num1 = int(input("输入第一个数字: "))
num2 = int(input("输入第二个数字: "))

print( num1,"和", num2,"的最小公倍数为", lcm(num1, num2))

```

24、简单计算器

```

# 定义函数
def add(x, y):
    """相加"""
    return x + y

def subtract(x, y):
    """相减"""
    return x - y

def multiply(x, y):
    """相乘"""
    return x * y

def divide(x, y):
    """相除"""
    return x / y

# 用户输入
print("选择运算:")
print("1、相加")
print("2、相减")
print("3、相乘")
print("4、相除")

choice = input("输入你的选择(1/2/3/4):")

num1 = int(input("输入第一个数字: "))
num2 = int(input("输入第二个数字: "))

if choice == '1':
    print(num1, "+", num2, "=", add(num1, num2))

elif choice == '2':
    print(num1, "-", num2, "=", subtract(num1, num2))

elif choice == '3':
    print(num1, "*", num2, "=", multiply(num1, num2))

elif choice == '4':
    if num2 != 0:
        print(num1, "/", num2, "=", divide(num1, num2))
    else:
        print("分母不能为0")
else:
    print("非法输入")

```

25、生成日历

```

# 引入日历模块
import calendar

# 输入指定年月
yy = int(input("输入年份: "))
mm = int(input("输入月份: "))

# 显示日历
print(calendar.month(yy, mm))

```

26、文件IO

```
# 写文件
with open("test.txt", "wt") as out_file:
    out_file.write("该文本会写入到文件中\n看到我了吧！")

# Read a file
with open("test.txt", "rt") as in_file:
    text = in_file.read()

print(text)
```

27、字符串判断

```
# 测试实例一
print("测试实例一")
str = "runoob.com"
print(str.isalnum()) # 判断所有字符都是数字或者字母
print(str.isalpha()) # 判断所有字符都是字母
print(str.isdigit()) # 判断所有字符都是数字
print(str.islower()) # 判断所有字符都是小写
print(str.isupper()) # 判断所有字符都是大写
print(str.istitle()) # 判断所有单词都是首字母大写，像标题
print(str.isspace()) # 判断所有字符都是空白字符、\t、\n、\r

print("-----")

# 测试实例二
print("测试实例二")
str = "Bake corN"
print(str.isalnum())
print(str.isalpha())
print(str.isdigit())
print(str.islower())
print(str.isupper())
print(str.istitle())
print(str.isspace())
```

28、字符串大小写转换

```
str = "https://www.cnblogs.com/ailiailan/"
print(str.upper()) # 把所有字符中的小写字母转换成大写字母
print(str.lower()) # 把所有字符中的大写字母转换成小写字母
print(str.capitalize()) # 把第一个字母转化为大写字母，其余小写
print(str.title()) # 把每个单词的第一个字母转化为大写，其余小写
```

29、计算每个月天数

```
import calendar
monthRange = calendar.monthrange(2016,9)
print(monthRange)
```

30、获取昨天的日期

```
# 引入 datetime 模块
import datetime
def getYesterday():
    today=datetime.date.today()
    oneday=datetime.timedelta(days=1)
    yesterday=today-oneday
    return yesterday

# 输出
print(getYesterday())
```