

循环语句

while 语句

1. 作用:
可以让一段代码满足条件，重复执行。
2. 语法:
while 条件:
 满足条件执行的语句
else:
 不满足条件执行的语句
3. 说明:
 else 子句可以省略。
 在循环体内用 break 终止循环时,else 子句不执行。

"""

猜数字 2.0 版本

如果猜错 3 次，退出循环，提示 次数到达 3 次。

练习：exercise04.py

"""

生成随机数的工具

import random

产生一个随机数

random_number = random.randint(1, 100)

print(random_number)

count = 0

while count < 3:

 count += 1

 input_number = int(input("请输入："))

 if input_number > random_number:

 print("大了")

 elif input_number < random_number:

 print("小了")

 else:

 print("恭喜猜对了，总共猜了" + str(count) + "次。")

 break

else: # 可以判断循环是从条件离开的，还是从循环体中离开(break)的。

 print("次数到达 3 次")

print("后续逻辑")

for 语句

1. 作用:

用来遍历可迭代对象的数据元素。

可迭代对象是指能依次获取数据元素的对象，例如：容器类型。

2. 语法:

for 变量列表 in 可迭代对象:

语句块 1

else:

语句块 2

3. 说明:

else 子句可以省略。

在循环体内用 break 终止循环时,else 子句不执行。

"""

for

for 变量 in 可迭代对象:

循环体

练习: exercise05.py

exercise06.py

"""

for item in "我爱Python":

print(item)

range(开始值, 结束值, 步长) --> 整数生成器

不包含结束值

for item in range(0,5,1):# 0 1 2 3 4

print(item)

range(结束值)

for item in range(5):# 0 1 2 3 4

print(item)

range(开始值, 结束值)

for item in range(2,6):# 2 3 4 5

print(item)

倒序

for item in range(6,3,-1):#6 5 4

print(item)

跳着

for item in range(2,11,2):#2 4 6 8 10

print(item)

for + range 特别适合做预定次数的循环.

for item in range(50):

print(item)

range 函数

1. 作用:
用来创建一个生成一系列整数的可迭代对象(也叫整数序列生成器)。
2. 语法:
`range(开始点, 结束点, 间隔)`
3. 说明:
函数返回的可迭代对象可以用 for 取出其中的元素
返回的数字不包含结束点
开始点默认为 0
间隔默认值为 1

跳转语句

break 语句

1. 跳出循环体，后面的代码不再执行。
2. 可以让 while 语句的 else 部分不执行。

continue 语句

跳过本次，继续下次循环。

```
"""
    continue
    练习: exercise07.py
"""
# 累加 1 -- 50 之间，能被 6 整除的数字.
# sum_value = 0
# for item in range(1,51):
#     # 如果能被 6 整除则累加
#     if item % 6 == 0:
#         sum_value += item
#
# print(sum_value)
sum_value = 0
for item in range(5,51):
    # 如果不能被 6 整除则跳过
    if item % 6 != 0:
        continue
    sum_value += item
```

```
print(sum_value)
```

容器类型

通用操作

数学运算符

1. `+`: 用于拼接两个容器
2. `+=`: 用原容器与右侧容器拼接,并重新绑定变量
3. `*`: 重复生成容器元素
4. `*=`: 用原容器生成重复元素, 并重新绑定变量
5. `< <= > >= == !=`: 依次比较两个容器中元素,一但不同则返回比较结果。

成员运算符

1. 语法:
 数据 in 序列
 数据 not in 序列
2. 作用:
 如果在指定的序列中找到值, 返回 bool 类型。

索引 index

1. 作用: 访问容器元素
2. 语法: 容器[整数]
3. 说明:
 正向索引从 0 开始, 第二个索引为 1, 最后一个为 len(s)-1。
 反向索引从-1 开始,-1 代表最后一个,-2 代表倒数第二个,以此类推,第一个是-len(s)。

切片 slice

1. 作用:

从容器中取出相应的元素重新组成一个容器。

2. 语法:

容器[(开始索引):(结束索引):(步长)]

3. 说明:

小括号()括起的部分代表可省略

结束索引不包含该位置元素

步长是切片每次获取完当前元素后移动的偏移量

"""

```
    str 通用操作
"""
#1. 数学运算符
# 拼接 + +=
str01 = "悟空" + "八戒"
print(id(str01)) # 139637893711808
# 产生了新的字符串"悟空八戒唐僧"
str01 += "唐僧" # 139637893711984
print(id(str01))
# 乘法: 重复产生
str02 = "白雪公主"
str02 *= 10
print(str02)
# 比较运算: 依次比较两个容器中元素, 一但不同则返回比较结果
# 字符串比较的是编码值
str03 = "ac 悟空"
str04 = "b 唐僧"
print(str03 > str04)
# 2. 成员运算
re = "八戒" in str01
print(re)
# 3. 索引: 定位单个元素
message = "我叫齐天大圣。"
# 获取第一个字符
print(message[0])
print(message[-len(message)])
# 获取最后一个字符
print(message[-1])
print(message[len(message)-1])
# IndexError: string index out of range
# 索引越界
# print(message[500])
# 4. 切片: 定位多个元素
print(message[2:6]) # 齐天大圣
print(message[2:6:2]) # 齐大
# 开始索引, 默认从头开始。
print(message[:6]) # 我叫齐天大圣
```

```
# 结束索引，默认到末尾。  
print(message[2:])# 齐天大圣。  
# 正向获取  
print(message[:])# 我叫齐天大圣。  
# 反向获取  
print(message[::-1])# 。圣大天齐叫我  
print(message[-3:-6:-1])# 大天齐  
# 可以同时使用反向与正向索引  
print(message[-3:1:-1])# 大天齐  
print(message[3:1])# 空  
print(message[3:1:-1])# 天齐  
print(message[1:1])# 空  
print(message[-2:1])# 空  
print(message[1:500])# 叫齐天大圣。
```

内建函数

1. len(x) 返回序列的长度
2. max(x) 返回序列的最大值元素
3. min(x) 返回序列的最小值元素
4. sum(x) 返回序列中所有元素的和(元素必须是数值类型)

字符串 str

定义

由一系列字符组成的不可变序列容器，存储的是字符的编码值。

编码

1. 字节 byte：计算机最小存储单位，等于 8 位 bit.
2. 字符：单个的数字，文字与符号。
3. 字符集(码表)：存储字符与二进制序列的对应关系。
4. 编码：将字符转换为对应的二进制序列的过程。
5. 解码：将二进制序列转换为对应的字符的过程。
6. 编码方式：
 - ASCII 编码：包含英文、数字等字符，每个字符 1 个字节。
 - GBK 编码：兼容 ASCII 编码，包含 21003 个中文；英文 1 个字节，汉字 2 个字节。

- Unicode 字符集：国际统一编码，旧字符集每个字符 2 字节，新字符集 4 字节。
- UTF-8 编码：Unicode 的存储与传输方式，英文 1 字节，中文 3 字节。

相关函数

1. ord(字符串):返回该字符串的 Unicode 码。
2. chr(整数):返回该整数对应的字符串。

"""

```
str 编码
练习:exercise08.py
"""
name = "悟空"
name = "孙悟空"
# 数 --> 字
str01 = chr(97)
print(str01)
# 字 --> 数
number = ord("b")
print(number)
```

- 3.

字面值

单引和双引号的区别

1. 单引号内的双引号不算结束符
2. 双引号内的单引号不算结束符

三引号作用

1. 换行会自动转换为换行符\n
2. 三引号内可以包含单引号和双引号
3. 作为文档字符串

"""

```
str 字面值
"""
name = "悟空"
name = '悟空'
# 可见即所得
name = """悟空"""
```

```

name = '''
    悟
    空
    '''

message = """我叫"齐'天'大圣'."""
print(message)
# 转义符:改变原有字符含义的特殊字符
# \" \' \\n 换行 \\t tab 键 \\
message = "我\\n叫\\"齐天\\t大圣\\"."
print(message)
url = "C:\\\\appData\\\\boaming\\\\ciniconfig"
# 原始字符串:没有转义符的字符串
url = r"C:\appData\boaming\ciniconfig"
print(url)

```

转义字符

1. 改变字符的原始含义。
`\' \' \'\'\'\' \\n \\ \\t \\0 空字符`
2. 原始字符串:取消转义。
`a = r"C:\newfile\test.py"`

字符串格式化

1. 定义:
生成一定格式的字符串。
2. 语法:
字符串%(变量)
"我的名字是%s,年龄是%s" % (name, age)
3. 类型码:
`%s 字符串 %d 整数 %f 浮点数`

"""

```

字符串格式化
"..%s..%s.."%(变量, 变量)
练习:exercise10
"""

name = "孙悟空"
age = 800
score = 85.62
# 我叫:xx, 今年:xx岁.
str01 = "我叫:" + name + ", 今年:" + str(age) + "岁."
str02 = "我叫:%s, 今年:%d岁, 成绩是:%.1f。"%(name, age, score)
print(str02)

```