

一、今日内容

- a. 强制类型转换
- b. python逻辑语句
- c. 数据类型 --- string
- d. 数据类型 --- 列表
- e. 数据类型 --- 元组
- f. 数据类型 --- 字典

二、昨日复习

- a. 计算机体系
- b. python简介
- c. python环境安装
 - i. python解释器
 - ii. pycharm编辑器
- d. python语法
 - i. 注释
 - 1. 单行注释
 - 2. 多行注释
 - ii. 变量
 - 1. 概念： 程序运行过程中，可以发生改变的量
 - 2. 特点： 无需提前声明，即拿即用
 - iii. 标识符
 - 1. 规则： 由数字,字母,下划线组成, 数字不能开头,不能和关键字重合
 - 2. 驼峰命名法
 - 3. 见名知意
 - 4. 通过下划线链接起来
 - iv. 运算符
 - 1. 算数运算符: / // ** = python没有 ++ -- ^
 - 2. 逻辑运算符: and or not
 - v. 输入输出:
 - 1. print
 - 2. input

三、类型转换

3.1 概念

将变量原来的类型转换为想要的数据类型

3.2 python数据类型

- i. 数字类型: int
- ii. 布尔类型: bool # define true 1 True False
- iii. 字符串类型: string 顺序表: 引号之间的内容就是字符串
- iv. 列表类型: list : 链表: []
- v. 元组类型: tuple: ()
- vi. 字典类型: dict 红黑树 {key : value }

3.3 强制类型转换

格式: 需要转换的类型 (需要进行转换的变量)

- i. 强制转换为字符串: str(var)
- ii. 强制转换为数字: int(var)
- iii. 强制转换为bool: bool(var)

.....

注意: 类型无法转换的时候, 会直接报错。

```
1 number = 100
2 # target: 将数字100转换为字符串类型
3 str_number = str(number)
4
5 print(type(str_number))
6 print(type(number))
7
8 # error 案例
9 name = "1231zzz23456" # 数字类型的字符串
10
11 int_name = int(name)
12 print(type(int_name))
13 print(type(name))
```

四、逻辑语句

4.1 判断语句

```
1 # 判断语句
```

```

2 """
3 单分支if语句
4
5 格式:
6 if condition:
7
8 (tab)code
9
10
11 含义:
12     如果 condition判断为true,则执行code
13
14 demo: 情景模型:从键盘中输入年龄,如果年龄>18岁,则输出哥网吧随便进
15
16 age = int(input("please input your age >>> "))
17
18 print("----- 判断开始 -----")
19
20 if age > 18:
21     print("哥, 网吧随便进!!")
22
23 print("----- 判断结束 -----")
24
25 双分支语句结构 if / else
26 格式:
27 if condition:
28     code1
29 else:
30     code2
31
32 含义:
33     condition为真的时候,执行code1, 如果条件为 false执行 code2
34
35 demo: 从键盘上输入年月日, 判断一下 日期是否合法.
36 year = int(input("请输入年份: >>>"))
37 month = int(input("请输入月份: >>>"))
38 day = int(input("请输入天: >>>"))
39
40 days = [0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
41 cur_day = days[month]
42
43 if month == 2 and (((year % 4 == 0) and (year % 100 != 0)) or (year % 400 == 0)):
44     cur_day = 29
45 if year > 0:

```

```

46     # 年份符合
47     if 0 < month < 13:
48         # 月份符合
49         if 0 < day <= cur_day:
50             print("您输入的日期合法")
51             print("aaa")
52         else:
53             print("您输入的日期不合法")
54     else:
55         print("month erorr!")
56 else:
57
58     print("year error !")
59
60
61 多分支条件语句 if / elif / else
62 格式:
63     if condition1:
64         code1
65     elif condition2:
66         code2
67     elif condition3:
68         code3
69     ....
70     else:
71     """

```

4.2 循环语句

4.2.1 while循环

```

1  # while循环
2  """
3  格式:
4      while condition:
5          code
6
7  含义: condition条件为 true 会一直执行code
8
9  demo:
10
11  """

```

```

12 # 向女朋友道歉:喝酒回来晚了,然后说100遍,我再也不喝酒了
13
14 i = 1
15
16 while i < 201:
17
18     print("第 %d 次,对不起,我再也不喝酒了, 5201314."% i)
19     i += 1

```

4.2.2 for 循环

```

1 # for 循环语句
2 """
3 格式:
4 for 临时变量 in 字符串,元组,列表,字典等可迭代的对象:
5     code
6
7 相关概念:
8     可迭代对象: 拥有迭代器的对象.一般是一个容器的对象
9     迭代器: 可以通过迭代器找到后面的内容.
10     mystr = "hello world"
11
12 含义:
13     将可迭代对象里面的值,依次赋值给 临时变量
14     """
15 mystr = "hello"
16
17 for i in mystr:
18
19     print(i)
20

```

4.2.3 循环控制关键字

1. break: 退出循环,
2. continue: 跳出本次循环

4.2.4 课堂练习

```

1 # target: 从键盘上随机输入一个数字, 判断一下这个数字,至少经过几步,可以变成一个斐波那契数字
2 """
3 斐波那契数字: 组成斐波那契数列的数字
4
5 1   1   2   3   5   8   13   21   .....
6
7 输入: 11
8         11 ---> 13 : + 2
9         11 ---> 8  : - 3
10 """
11 f1 = 1
12 f2 = 1
13
14 number = int(input("请输入number >>> "))
15
16 while True:
17
18     if number == f2:
19
20         print("0")
21         break
22
23     elif number < f2:
24
25         if abs(number - f2) > abs(number - f1):
26
27             print(abs(number - f1))
28         else:
29             print(abs(number - f2))
30
31         break
32
33     else:
34         f1, f2 = f2, f1 + f2
35
36 """

```

五、数据类型 --- str字符串类型

5.1 字符串概念

引号之间的内容，就是所谓的字符串。

三引号之间的内容： 多行字符串。

5.2 字符串的访问

- i. 下标： python中是支持负下标
 - 1. 从左到右： 下标是从0开始的。
 - 2. 从右到左： 下标是从-1开始的。
- ii. 切片: 获取字符串中的一部分内容

```
1 mystr = "abcdefg"
2
3 # python支持切片操作
4 """
5 切片功能：获取字符串中的一部分内容
6 语法格式： str[初始位置:终止位置:步长]
7 注意：切片的范围：[ )
8
9 切片中的默认值：
10     初始位置：字符串开始的位置
11     终止位置：字符串结束位置 + 1
12     步长： 1
13
14 """
15 print(mystr[::-1])
```

5.3 字符串的相关操作

5.3.1 运算符操作 + * == !=

```
1 # 运算符家族
2 str1 = "hello "
3 str2 = "world"
4 """
5 + : 字符串拼接。
6 demo:
7 mystr = "hello world hello python hello c"
8 # 将里面的空格替换为 %20
9 result = ""
10
```

```

11 for i in mystr:
12
13     if i == ' ':
14         result += "%20"
15     else:
16         result += i
17 print(result)
18
19 * : 字符串只能和数字相乘 ,表达: 字符串的复制
20
21 比较运算符
22 ==: 判断字符串是否相等
23 !=: 判断字符串是否不相等
24
25 """
26 print(str1 > str2)

```

5.3.2 转换家族 upper lower

```

1 # 转换家族 upper lower
2 """
3 使用场景: 图片验证码
4 upper函数
5
6 函数功能:
7     将字符串中的字母,全部转换为大写字母。
8
9 使用方式:
10     result = str.upper()
11
12 返回值:
13     result: 转换的结果
14
15
16 lower函数
17
18 函数功能:
19     将字符串中的大写字母,全部转换为小写字母
20
21 使用方式:
22     result = str.lower()
23

```



```

24 函数返回值:
25     result: 转换之后的结果
26
27     """
28
29 mystr = "GEDKSAJKLDJKLASJDLKSAJEIOWQJILE"
30
31 result = mystr.lower()
32 print(result)
33

```

5.3.3 查询家族 find rfind index rindex

```

1  # 查询函数 find rfind index rindex
2  """
3  find函数
4
5  函数功能:
6      在原字符串中, 查找子字符串所在的位置
7
8  使用方式:
9      result = str.find(sub)
10
11  函数参数:
12      sub : 需要查找的子字符串
13
14  返回值:
15      result : 成功: 子字符串所在原字符串的位置
16              失败: -1
17
18  demo:
19  # target: 判断url使用的是什麼协议
20  url = "http://cplusplus.com/"
21
22  # 寻找 ://
23  pos = url.find("://123321")
24  print("pos : ", pos)
25
26  if url[:pos] == "http":
27      print("this is http 协议")
28
29  elif url[:pos] == "https":

```

```

30     print("this is https 协议")
31 else:
32     print("其他协议")
33
34
35 rfind函数
36
37 函数功能：
38     从右向左在原字符串中, 查找子字符串所在的位置
39
40 使用方式：
41     result = str.rfind(sub)
42
43 函数参数：
44     sub : 需要查找的子字符串
45
46 返回值：
47     result : 成功：子字符串所在原字符串的位置
48             失败：-1
49
50 filename = "mio.c.cpp.hpp.jar.cc.h.py"
51
52 pos = filename.rfind(".")
53 print(pos)
54
55 print("this file type is ", filename[pos+1:])
56
57 index : 使用方式与 find相同,如果index 查找不到子字符串的位置,会直接报错。
58 rindex: 使用方式与 rfind相同,如果 rindex 查找不到子字符串的位置,会直接报错。
59 """

```

5.3.4 分割家族 split splitlines

```

1 # 分割家族 split splitlines
2 """
3 split函数
4
5 函数功能：
6     将原字符串按照子字符串进行分割操作
7
8 使用方式：
9     result = str.split(sub)

```

```

10
11 函数参数：
12     sub：分割方式
13
14 返回值：
15     result：[] 列表：容器,存放好了每一个字符串
16
17 splitlines函数
18
19 函数功能：
20     将字符串按照行来进行分割
21
22 使用方式：
23     result = str.splitlines()
24
25 返回值：
26     按行分割后的值：
27
28 http = '''Request URL: https://www.bilibili.com/
29 Request Method: GET
30 Status Code: 200
31 Remote Address: 111.31.33.19:443
32 Referrer Policy: strict-origin-when-cross-origin'''
33
34 # 按行进行分割
35 result = http.splitlines()
36
37 for ret in result:
38     tmp = ret.split(": ")
39     print(tmp)
40
41
42 """
43

```

六、数据类型 -- 列表list

6.1概念

列表是一个容器，列表里可以存放各式各样的数据类型，列表使用[int, bool, str, list, tuple, dict]. 本质：双向带头的循环链表

demo: l1 = [100, True, "hello world", [1, 2, 3, 4]]

6.2 访问

列表也是支持下标和切片进行访问

i. 下标：列表支持负下标访问

1. 从左往右：下标从 0 开始的
2. 从右往左：下标从-1 开始的

ii. 切片：语法 list[初始位置：终止位置：步长]

```
1 l1 = [100, True, "hello world", [1, 2, 3, 4]]
2 print(l1[0:3:1])
```

6.3 列表的增删改查

```
1 # 三国演义 曹老板的故事
2
3 home = ["曹操"]
4 job = ["董事长"]
5
6 # append：函数
7 """
8 功能：
9     在列表的尾部插入数据。
10 使用方式：
11     list.append(element)
12 """
13 # 将荀彧加入 曹氏文化有限公司
14 home.append("荀彧")
15 job.append("人事主管")
16
17 # insert：函数
18 """
19 功能：
20     在列表的指定位置插入元素
21 使用方式：
22     list.insert(pos, element)
23 """
24 home.insert(1, "夏侯惇")
25 job.insert(1, "保安队长")
26
27 # extend函数
```

```

28 """
29 功能：
30     将一个容器中的元素，依次加入到列表中
31 使用方式：
32     list.extend(list1)
33 """
34 tang = ["武则天", "狄仁杰", "大乔", "小乔", "杨玉环", "李白"]
35 jobs = ["CEO", "男秘书", "东吴女主播", "带货主播", "减肥达人", "洗浴中心-刮痧师傅"]
36
37 home.extend(tang)
38 job.extend(jobs)
39
40 #列表的修改 语法格式：list[pos] = value
41 home[7] = "黄月英"
42 job[7] = "科学家"
43
44 # 列表的删除
45 """
46 del ：根据下标进行删除。如果没有给出下标，会删除掉整个变量。
47 remove：根据值对列表的内容进行删除操作，如果给出的内容不存在，则会报错处理
48 pop：根据下标，对列表进行删除操作，如果没有给出下标，则会删除最后一个元素
49 """
50 # del home[7]
51 # del job[7]
52 # home.remove("黄月英111")
53 # job.remove("科学家11")
54 # home.pop()
55 # job.pop()
56
57 print("""*20, "曹氏文化有限公司", "*" * 20)
58 i = 0
59
60 while i < len(home): # len:返回列表的长度
61
62     print(" *15, "(%d)" % i, "职位：", job[i], "姓名：", home[i])
63
64     i += 1
65
66 print("""*20, "曹氏文化有限公司", "*" * 20)
67
68

```

