Python语法基础

本节重点：

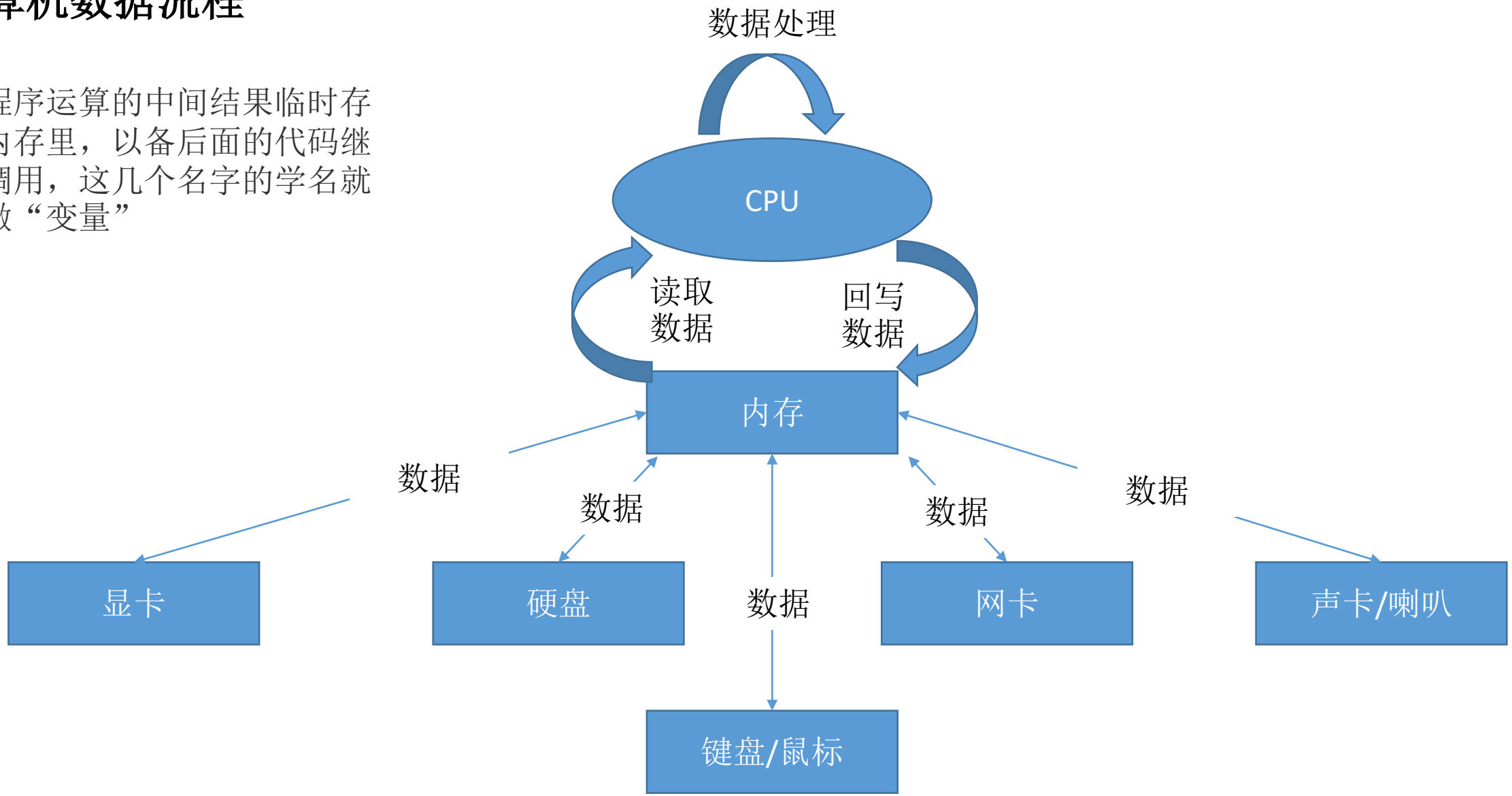
- 基本用法
- 变量定义
- 注释使用
- 运算符
- 流程控制
- 输入&输出
- 文件IO

基本用法

代码	
第一个程序	■ print(“Hello World!”)
三种执行方式	■ 启动python后，运行。。。 ■ 代码保存在py文件中，python 调用 ■ 在IDE中修改、调试、执行。。。
注释	■ 单行注释以 # 开头（pycharm快捷方式） ■ 多行注释：“”” ””””
行与缩进	■ python最具特色的就是使用缩进来表示代码块，不需要使用大括号 {} 。 ■ 缩进的空格数是可变的，但是同一个代码块的语句必须包含相同的缩进空格数。
空行	■ 空行并不是Python语法。书写时不插入空行，Python解释器运行也不会出错。 ■ 空行的作用在于分隔两段不同功能或含义的代码，便于日后代码的维护或重构。 ■ **记住：**空行也是程序代码的一部分。

计算机数据流程

把程序运算的中间结果临时存到内存里，以备后面的代码继续调用，这几个名字的学名就叫做“变量”



变量：物理对象数字化建模

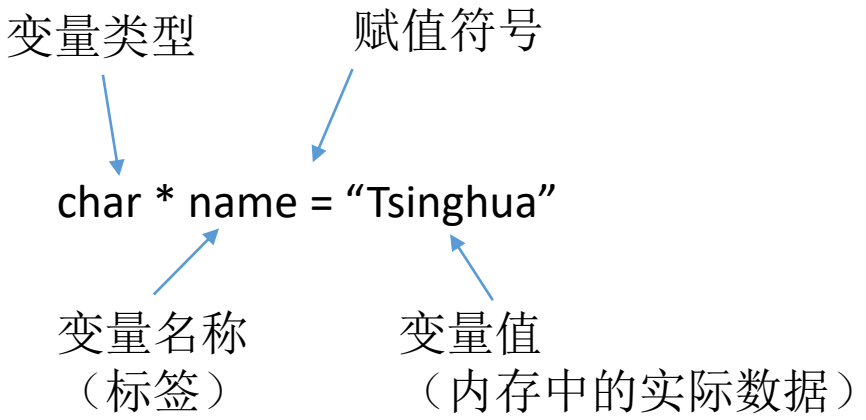
物理对象	参数	变量	类型
人	姓名	name	字符串
	年龄	age	整型数
	身高	height	浮点数
	收入	salary	浮点数
	性别	sex	整型数 布尔型

变量：基本数据类型

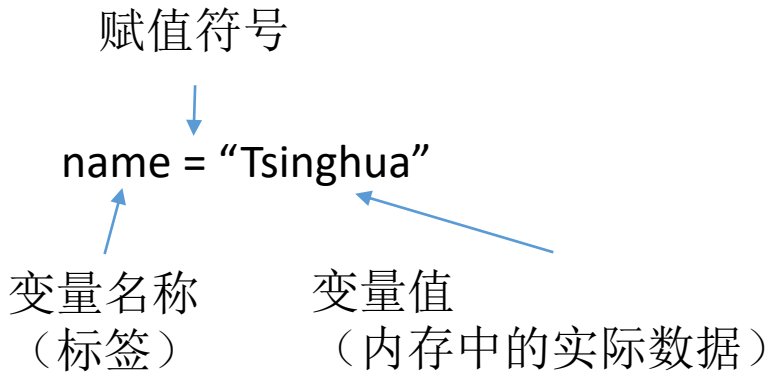
类型	c/c++	python
布尔型	bool	bool
整型	char/unsigned char short/unsigned short int/unsigned int long/unsigned long	int
浮点型	float double	float
字符串	char *	str
复数	-	complex
未赋值	-	None

变量定义

c/c++



python



变量定义规则

- 字符串中：“”与”不区分。
- 变量名只能是 字母、数字或下划线的任意组合
- 变量名的第一个字符不能是数字
- 以下关键字不能声明为变量名['and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'exec', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'not', 'or', 'pass', 'print', 'raise', 'return', 'try', 'while', 'with', 'yield']

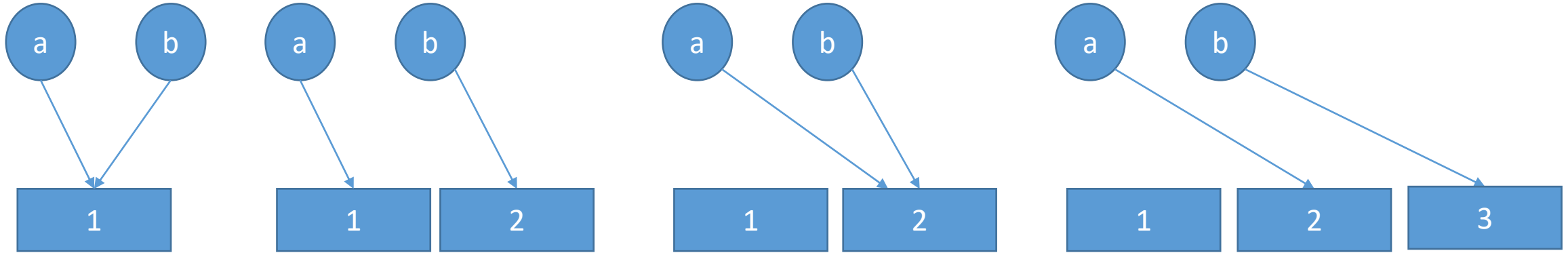
思考：

1、a=1，执行过程？

2、以下执行结果？

```
a, b=1,1
print(a, id(a))
print(b, id(b))
b=2
print(b, id(b))
a=b
print(a, id(a))
b=3
print(a, id(a))
```

变量的创建过程



a, b=1,1

b=2

a=b

b=3

命名规则

驼峰体 (java/js推荐用)	下划线 (python推荐)	匈牙利命名法 (c/c++推荐)
AgeOfTsinghua = 32	age_of_tsinghua = 32	int nValue = 0 char cValue = ‘a’ float fValue = 1.12 char * szValue = “Tsinghua” oValue… pValue… m_ g_ c_

- 常用变量：全部大写： PI=3.1415926
- 不合理的命名方式：
  - 变量名为中文拼音
  - 变量名过长
  - 变量名词不达意

代码注释

代码注释分单行和多行注释， 单行注释用#，多行注释可以用三对双引号""" """“

```
def subclass_exception(name, parents, module, attached_to=None):
    """
    Create exception subclass. Used by ModelBase below.

    If 'attached_to' is supplied, the exception will be created in a way that
    allows it to be pickled, assuming the returned exception class will be added
    as an attribute to the 'attached_to' class.
    """
    class_dict = {'__module__': module}
    if attached_to is not None:
        def __reduce__(self):
            # Exceptions are special - they've got state that isn't
            # in self.__dict__. We assume it is all in self.args.
            return (unpickle_inner_exception, (attached_to, name), self.args)
```

1.不用全部加注释，只需要在自己觉得重要或不好理解的部分加注释即可

2.注释可以用中文或英文，但绝对不要拼音噢

运算符

https://www.runoob.com/python/python-operators.html

算术运算符	比较运算符	赋值运算符	位运算符	逻辑运算符	成员运算符	身份运算符
+	==	=	&	and	in	is
-	>	+=	|	or	not in	is not
*	>=	-=	~	not		
/	<	*=	^			
%	<=	/=	>>			
**	!=	**=	<<			
//		//=				

思考：is 与 == 的区别？

流程控制

if单分支

```
if a == 1:
    pass
```

if双分支

```
if a == 1:
    pass
else:
    pass
```

三元运算赋值

```
a=10
b=20
c = a if a < b else b

c = a > b ? a : b
```

if多分支

```
if a == 0:
    pass
elif a >100:
    pass
elif a > 0:
    pass
else:
    pass
```

while循环

```
while True:
    pass
    if b == 0:
        pass
        continue
    if a == 0:
        pass
        break
    pass
else:
    pass
```

for遍历

```
for a in “python” :
    print(a)
else:
    pass
```

字典、列表、字符串等可迭代对象

range的4种用法（顾头不顾尾）

```
# range 的四种用法：

# range(10)
# 等价于：
# for(int i = 0;i < 10;i += 1)
# {
#
# }

# range(2, 10)
# 等价于：
# for(int i = 2;i < 10;i += 1)
# {
#
# }
```

```
# range 的四种用法：

# range(2, 10, 2)
# 等价于：
# for(int i = 2;i < 10;i += 2)
# {
#
# }

# range(10, 2, -1)
# 等价于：
# for(int i = 10;i > 2;i -= 1)
# {
#
# }
```

课堂练习（1）

1. 利用for循环和range输出：
   - 从大到小输出[1,00]
   - 从小到到输出[100,1]
   - 从大到小输出(1,100)中所有的偶数
   - 从大到小输出(100,1)中所有的偶数
2. 利用while循环输出：
   - 从大到小输出1 - 100
   - 从小到到输出100 – 1
   - 输出 1,2,3,4,5, 7,8,9, 11,12
   - 输出100-50，从大到小，如100，99，98…，到50时再从0循环输出到50，然后结束
   - 输出 1-100 内的所有奇数
   - 输出 1-100 内的所有偶数
   - 输出2-3+4-5+6…+100 的和

3种字符串格式化操作方法

- “旧式”字符串解析（%s操作）：
  - 'Hey %s!’ % s1
  - 'Hey %s, there is a %s error!’ % (s1, n1)
  - 'Hey %(name)s, there is a %(num)s error!' % {"name": s1, “num":n1}

- “新式”字符串格式化（str.format）
  - 'Hello, {}'.format(name)
  - 'Hey {name}, there is a {num} error!'.format(name=s1, num=n1)

- Python 3.6的新技能（f语法）：

```
name='abcd'
num = 10
c=f'Hi {name}, there is a {num} error!'
print(c)
```

```
name='abcd'
a = 5
b = 10
c=f'Hi {name} Five plus ten is {a + b} and not {2 * (a + b)}.'
print(c)
```

格式化输入与输出

示例：通过终端输入姓名、年龄、性别，并输出

涉及到的两个函数：

- input
- print

注意点：input接收的所有输入默认都是字符串格式！

```
name = input("请输入姓名")
sex = input("请输入性别")
age = input("请输入年龄")
str = """
            姓名：%s
            年龄：%s
            性别：%s
            """ % (name, age, sex)
print("姓名：%s 年龄：%d 性别：%s\n" %
name, age, sex)
```

进一步：格式化输出：

---------- info of XX ----------
姓名 : XX
年龄 : YY
性别: Teacher
---------- end ----------

```
str = """
------------ info of XX -----------
姓名：%s
年龄：%s
性别：%s
-------------- end -----------------
""" % (name, age, sex)
print(str)
```

课堂练习（2）

1. 编程计算：1+2+…+n，n为一个从键盘输入的数字。
2. 假设一年期定期利率为3.25%，计算一下需要过多少年，一万元的一年定期存款连本带息能翻番？
3. 一球从100米高度自由落下，每次落地后反跳回原高度的一半；求它在第10次落地时，共经过多少米？第10次反弹多高？
4. 编程实现：从键盘接收一百分制成绩（0~100），要求输出其对应的成绩等级A~E。
   - 其中，90分以上为'A'，80~89分为'B'，70~79分为'C'，60~69分为'D'，60分以下为'E'，若不是0-100之间，则报错并要求再次输入。
5. 输入一年份，判断该年份是否是闰年并输出结果。
   - 注：凡符合下面两个条件之一的年份是闰年。（1） 能被4整除但不能被100整除。（2） 能被400整除。
6. 制作趣味模板程序
   - 需求：根据用户输入的名字、地点、爱好，进行任意显示，如：敬爱可爱的xxx，最喜欢在xxx地方YYY
7. 使用while,完成以下图形的输出

```
*
**
***
****
*****
****
***
**
*
```

课堂练习（3）

1. 猜年龄游戏。
   - 首先预设1个年龄。
   - 用户输入猜测的年龄，如果大了，则提示“大”，如果小了，则提示“小”，如果相同，则提示“猜对了”。
   - 允许用户最多猜测3次，3次均猜不中，则提示：“3次猜错，游戏失败！”
2. 猜4位数字游戏。

   1、输入一个4位数字a（不能重复）

   2、猜一个数b，程序自动答复：mAnB

   m:数字和位次全对的数量。

   n:只有数字对但位不对个数的数量。

   m+n <= 4

   3、4A，代表完全猜出来了。
3. 实现用户输入用户名和密码，并登录验证测试：
   - 当用户名为 seven 且 密码为 123 时,显示登陆成功,否则显示登陆失败!
   - 失败时，允许重复输入三次。
   - 失败3次后，则提醒出错，程序运行退出。

文件IO

	文本文件	二进制文件	with用法
文件读	f = open(“a.txt”, “r”) line = f.readline() line = f.readlines() f.close()	f = open(“a.txt”, “rb”) line = f.read(n=256) f.close()	with open(“a.txt”, “r”) as f: line = f.read()
文件写	f = open(“a.txt”, “w”) f.wirte(line) f.wirtelines(line) f.close()	f = open(“a.txt”, “wb”) f.wirte(line) f.close()	with open(“a.txt”, “w”) as f: f.write(line)
文件追加	f = open(“a.txt”, “a”) f.write(line) f.close()	f = open(“a.txt”, “ab”) f.write(line) f.close()	with open(“a.txt”, “w”) as f: f.write(line)

文件IO

示例1：读取多行

```
with open("test.txt", "r", encoding='utf-8') as f:
    data = f.read()
    print(data)


with open("test.txt", "r", encoding='utf-8') as f:
    data = f.readlines()
    print(data)
    print(len(data))
    for str in data:
        print(str)

with open("test.txt", "r", encoding='utf-8') as f:
    data = f.readline()
    print(data)
    print(len(data))
```

示例2：文件copy

```
with open("1538312121_IMG_8699.JPG", "rb") as f_read:
    with open("new.JPG", "wb") as f_write:
        while True:
            data = f_read.read(102400)
            if not data:
                break
            f_write.write(data)
```

课堂练习（1）

写代码实现文件复制功能：

需求描述：

- 用户输入拟复制的文件名称
- 若文件不存在，则报错（例子：os.path.exists("hello.txt")）。
- 用户输入新文件的名称
- 若新文件已经存在，则提示是否覆盖。
  - 如果是，则覆盖。
  - 如果否，则退出。

课堂练习（1）

```
# #1、用户输入拟复制的文件名称
# old_file_name = input("输入拟复制的文件名称:>>").strip()
# #2、若文件不存在，则报错（例子：os.path.exists("hello.txt")）。
# import os
# if not os.path.exists(old_file_name):
#     print("文件不存在。。。")
#     exit()
#
# #3、用户输入新文件的名称
# new_file_name = input("输入新文件的名称:>>").strip()
#
# #4、若新文件已经存在，则提示是否覆盖。
# if os.path.exists(new_file_name):
#     flag = input("是否覆盖(Y/N):>>").strip()
#     if flag != 'Y':
#         # 如果否，则退出。
#         exit()
#
# #5、如果是，则覆盖。
# file_size = 0
# with open(old_file_name, "rb") as f_old:
#     with open(new_file_name, "wb") as f_new:
#         while True:
#             b = f_old.read(1024)
#             if not b:
#                 break
#             file_size += len(b)
#             f_new.write(b)
#
# #6、打印输出结果
# print(f"文件复制成功！{old_file_name} -》 {new_file_name} size = {file_size}")
```

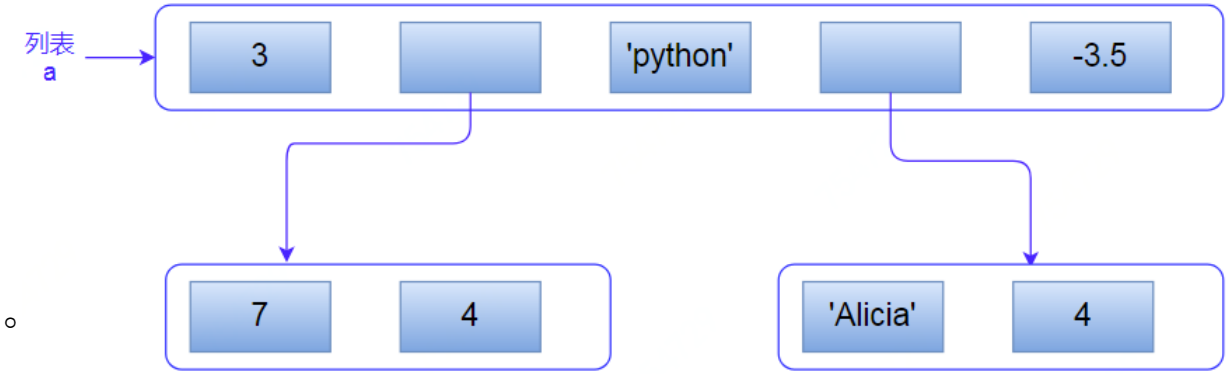
Python语法基础

本节重点：

- 复杂变量
- 列表生成式

复杂变量-列表（list)

- Python中最基本的数据结构，类比于c++的数组。
- 序列中的每个元素都分配一个数字-它的位置，或索引，第一个索引是0，第二个索引是1，依此类推。
- 列表创建方法：用逗号分隔的不同的数据项使用方括号括起来。
- li=[3,[7,4],'python',['Alicia',4],-3.5]
- 特色功能：
  - 数据成员类型不需要一致相同。
  - 内嵌操作。
  - 切片操作（顾头不顾尾、倒序、默认值）。
  - 加，乘等操作。
  - 许多内置函数。。。
    - 增：append/insert/extend
    - 删：del a[i:] / pop / remove / clear
    - 改：copy / sort / reverse
    - 查：index / in / not in / count / len / max / min / for




```
# 浅copy 与 深copy
# 浅copy：  不开辟新的内存，新变量指向已有内存块。
# 深copy:   需要开辟新的内存，原内存数据镜像到新内存，并新变量指向新的内存块。
```

示例：用列表实现栈

课堂练习（1）

写代码，有如下列表，按照要求实现每一个功能，li＝['wangbin', 'jessie', 'eric']

- 计算列表长度并输出 len(li)
- 列表中追加元素“seven”，并输出添加后的列表 li.append(“seven”)
- 请在列表的第1个位置插入元素“Tony”，并输出添加后的列表 li.insert(1, “Tony”)
- 请修改列表第2个位置的元素为“Kelly”，并输出修改后的列表 li[2] = ‘Kelly’
- 请删除列表中的元素“eric”，并输出修改后的列表 li.remove(“eric”)
- 请删除列表中的第2个元素，并输出删除的元素的值和删除元素后的列表 ele = li.pop(2)
- 请删除列表中的第3个元素，并输出删除元素后的列表 ele = li.pop(3)
- 请删除列表中的第2至4个元素，并输出删除元素后的列表 del li[2:5:]
- 请将列表所有的元素反转，并输出反转后的列表 li.reserve()
- 请使用for、len、range输出列表的索引 for idx in range(len(li))
- 请使用enumrate输出列表元素和序号（序号从100开始） for idx, val in enumerate(li) print(idx+100, val)
- 请使用for循环输出列表的所有元素 for val in li: ….

课堂练习（1）

1、变量命名的随意性问题。

2、写法。很多默认参数。

range(10)

range(1,10)

range(1,10,2)

切片： li[::]、 li[0:10:]、 li[1::2]、 li[:11:2]、 li[::2]、li[1:10:2]、li[-1:-5:-1]

2、函数用法不会用，怎么办？

(1) baidu

(2) pycharm

（3）求助

复杂变量-元组（tuple)

- 一种特殊的list,成员不可变。
- 语法： tp=(1,2,3) or tp=1,2,3
- 所有增/删/改操作均不可用。
- 与list的相关转换：
  - tp=tuple(li)
  - li=list(tp)
- 与list相关的优势：
  - 元素不可变（保护）
  - 操作速度更快（效率）。
- 一些特殊注意事项：
  - 元组里面包含可变数据类型，可以间接修改元组的内容
  - 元组如果只有一个元素的时候，后面一定要加逗号，否则数据类型不确定。
  - 支持切片操作， 切完片之后，依然是元组。

思考：如何理解成员不可变？

- 许多内置函数。。。
  - ~~增：append/insert/extend~~
  - ~~删：del a[i:] / pop / remove / clear~~
  - ~~改：copy / sort / reverse~~
  - 查：index / in / not in / count / len / max / min / for

```
# 浅copy 与 深copy
# 浅copy ：  不开辟
# 深copy：  需要开
```

复杂变量-字符串（str)

- 字符串，一种特殊的元组，成员不可变。
- 语法： s1=‘123abc’
- 通用方法：参考tuple
- 一些特殊的使用语法：
  - 判断：startswith/ endswidth / isalpha / isdigit / isalnum / isspace
  - 查找：find / rfind
  - 修改：replace / lower / upper / capitalize / title
  - 格式化：ljust / rjust / center / strip / rstrip / lstrip
  - 拆分：split / splitlines / partition / rpartition
  - 组织：join （可迭代对象） -> 固定间隔的字符串
  - 赋值：“”” ”””

课堂练习（2）

写代码，有如下元组，请按照功能要求实现每一个功能

tu＝('alex', 'eric', 'rain')

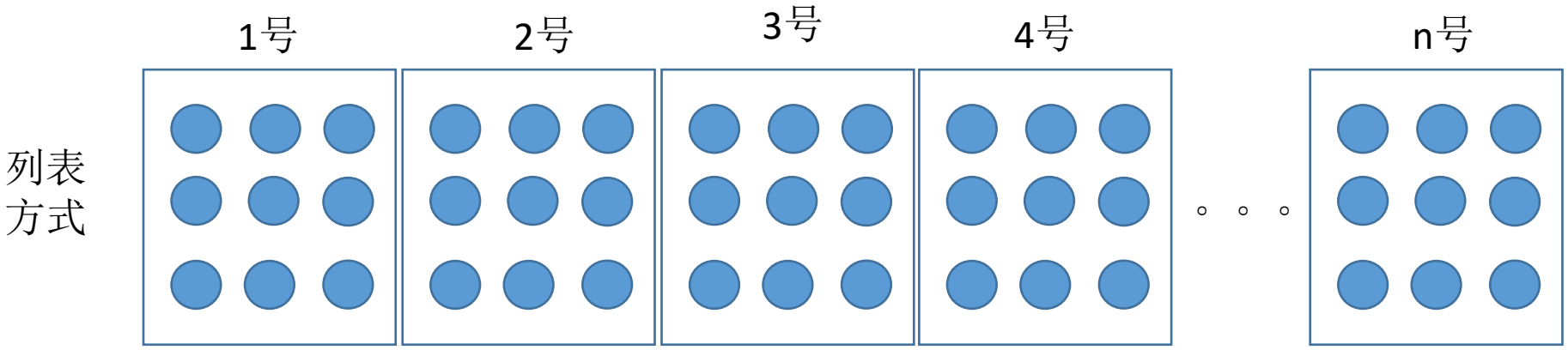
- 计算元组长度并输出： len()
- 获取元组的第2个元素，并输出，tu[2]
- 获取元组的第1-2个元素，并输出, tu[1:3]
- 请使用for输出元组的元素，for ele in tu: …
- 请使用for、len、range输出元组的索引 for idx in range(len(tu)):
- 请使用enumrate输出元组元素和序号（序号从10开始）for idx, val in enumerate(li) print(idx+10, val)

课堂练习（2）

1. 用代码将字符串“Tsinghua University”按空格切割。
2. 将列表['清', '芬', '园'] 中的每一个元素使用 ‘_’ 连接为一个字符串
3. 利用下划线将列表的每一个元素拼接成字符串 li＝['p1', 'p2, 'p3’]
4. 获取字符串子序列，仅不包含最后一个字符，如：woaini则获取woain root则获取roo
5. 将字符串变量对应的值变成小写，并输出结果
6. 将字符串变量对应的值中的"o"，替换为"p"，并输出结果
7. 判断字符串变量对应的值是否以"go"开头，并输出结果
8. 移除字符串变量对应值的两边的空格，并输出移除后的内容
9. 请输出name变量中的值“Q” 的索引的位置

复杂变量-字典（dict)

- 如何存放人员信息？



list的问题：

- 检索：O (n)复杂度
- 增删：需要做比较多的内存操作。
  - 在某位置添加元素时， 该位置后面的元素需要向后顺移。
  - 从某位置删除元素时， 该位置后面的元素需要向前顺移。

医院就医：
list: 在医生门口人肉排队
dict: 等待叫号。

- 为了保存具有映射关系的数据，Python 提供了字典，字典相当于保存了两组数据，其中一组数据是关键数据，被称为 key；另一组数据可通过 key 来访问，被称为 value。(key： 号码， value: 病人）

复杂变量-字典（dict)

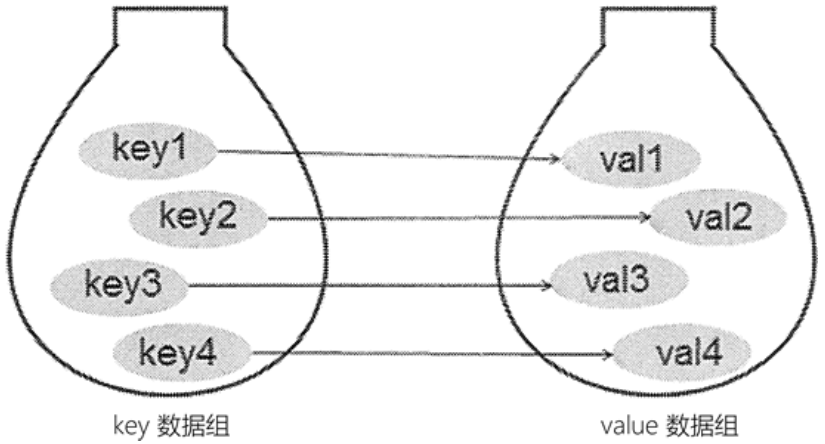
- 主要特性：
  - 用{}括起来，以冒号(:)分割键-值对，各键值对用逗号(,)分隔开
  - 字典值可以没有限制地取任何python对象，既可以是标准的对象，也可以是用户定义的
  - 不允许同一个键出现两次。创建时如果同一个键被赋值两次，后一个值会被记住：
  - 键必须不可变，所以可以用数，字符串或元组充当，所以用列表就不行，如下实例：

- 特殊用法：
  - 增加：[]
  - 删除：del dict[key] / dict.pop(key) / dict.clear()
  - 修改：[]
  - 查找：[] / dict.get(key, default) / in / not in
  - 遍历：for key in dict: / for idx, key in enumerate(dict): / for key, values in dict.items():
  - 复制：dict.copy()
  - 合并：dict.update(d2)

复杂变量-字典（dict)

创建

```
scores = {'语文': 89, '数学': 92, '英语': 93}
print(scores)
# 空的花括号代表空的dict
empty_dict = {}
print(empty_dict)
# 使用元组作为dict的key
dict2 = {(20, 30):'good', 30:'bad'}
print(dict2)
```



添加

```
dict = {'Name':  'Eric', 'Age': 7}

dict['Age'] = 8  # update existing entry
dict['School'] = "Tsinghua"  # Add new entry

print("dict['Score']: ", dict['Score'])
print("dict['Age']: ", dict['Age'])
print("dict['School']: ", dict['School'])
```

删除

```
dict = {'Name':  'Eric', 'Age': 7}

dict['Age'] = 8  # update existing entry
dict['School'] = "Tsinghua"  # Add new entry

del dict['Age']
```

复杂变量-集合（set)

■ 一类特殊的dict，只有key，没有 value

集合运算	代码示意	结果
& 交集	s1={1,2,3} s2={2,3,4} s3 = s1 & s2	s3={2,3}
| 并集	s1={1,2,3} s2={2,3,4} s3 = s1 | s2	s3={1，2, 3，4}
- 补集	s3 = s1 - s2	s3={1}
^ 对称补集	s3 = s1 ^ s2	s3={1，4}
< 子集	s1={1,2,3} s2={2,3} s2 < s1	#True 判断子集
> 超集	s1={1,2,3} s2={2,3} s1 > s2	#True 判断超集
== 相同 != 不同	s1={1,2,3} s2={2,3，1} s2 == s1 # True	s1 != s2 # False

课堂练习（3）

1. dic = {'k1': "v1", "k2": "v2", "k3": [11,22,33]}

- 请循环输出所有的key for key in dic:
- 请循环输出所有的value for key in dic: dic[key]
- 请循环输出所有的key和value
- 请在字典中添加一个键值对，"k4": "v4"，输出添加后的字典 dic[‘k4’]= ‘v4’
- 请在修改字典中“k1”对应的值为“alex”，输出修改后的字典 dic[‘k1’] = ‘alex’
- 请在k3对应的值中追加一个元素44，输出修改后的字典 dic[‘k3’].append(44)
- 请在k3对应的值的第1个位置插入个元素18，输出修改后的字典 dic[‘k3’].insert(1,18)

2. 元素分类

有如下值列表[11,22,33,44,55,66,77,88,99,90]，将所有大于66的值保存至字典的第一个key中，将小于66的值保存至第二个key的值中。

即：{‘k1’:大于66的所有值列表, ‘k2’:小于66的所有值列表}

课堂练习（3）

有两个列表

- l1 = [11,22,33]
- l2 = [22,33,44]

功能要求：

- 获取内容相同的元素列表
- 获取l1中有，l2中没有的元素列表
- 获取l2中有，l1中没有的元素列表
- 获取l1和l2中内容都不同的元素

复杂变量（容器）

	特点	定位查找的算法复杂度
列表(list) [1,2,3]	■ 可存放多个值 ■ 按照从左到右的顺序定义列表元素，下标从0开始顺序访问，有序 ■ 可修改指定索引位置对应的值，可变	O(1) O(n)

元组(tuple)
(1,2,3)
	■ 类似于列表
	■ 元素不可变
	思考题：
	`a=([1,2],1,2)`
	`# a[0]=[3,2]`
	`# a[0][1]=3`
	`print(a)`
	`b=[1,2]`
	`a=(b,1,2)`
	`b[0]=3`
	`print(a)`	

字典(dict) {'a':1, 'b':2}	■ key-value结构 ■ key必须为不可变数据类型、必须唯一 ■ 可存放任意多个value、可修改、可以不唯一 ■ 无序 ■ 查询速度快，且不受dict的大小影响。	不支持下标定位 O(log(n))
集合(set) {'a', 'b'}	■ 里面的元素不可变 ■ 天生去重，在集合里没办法存重复的元素 ■ 无序，不像列表一样通过索引来标记在列表中的位置，元素是无序的，集合中的元素没有先后之分，如集合{3,4,5}和{3,5,4}算作同一个集合	不支持下标定位 O(log(n))

复杂变量

关注：复杂变量的深浅copy问题

	增	删	改	查	遍历	切片

列表(list)
[1,2,3]
	append:追加
	insert:插入
	extend:合并
		del list[i]
		pop
		clear
			list[i] =‘ttt’	in
				not in
				index("eva")
				count("eva")
					for val in li:
					print(val)
					
					for idx, val in enumerate(li):
					print(idx, val)
						a[1:]
						a[:2]
						a[1:5:2]
						
						顾头
						不顾尾

元组(tuple) (1,2,3)	不可增	不可删	元组本身不可变，如果元组中还包含其他可变元素，这些可变元素可以改变			

字典(dict)
{‘a’:1, ‘b’:2}
	dict[‘a’]=2	del dict[i]
		clear
			duct[‘a’]=2	in
				not in
				::get
					for k in d:
					print(k, d [k])
					for k,v in d.items():
					print(k,v)
						不可
						切片

集合(set) {‘a’, ‘b’}	add & | - ^	remove clear	本身不可变，如果元组中还包含其他可变元素，这些可变元素可以改变	in isdisjoint issubset issuperset	同上	不可 切片

课堂练习（6）

1. 在不改变列表数据结构的情况下找最大值li = [1,3,2,7,6,23,41,243,33,85,56]
2. 在不改变列表中数据排列结构的前提下，找出以下列表中最接近最大值和最小值的平均值的数
   - li = [-100,1,3,2,7,6,120,121,140,23,411,99,243,33,85,56]
3. 列表test = ['alex','egon','yuan','wusir','666']中，编程实现：
   - 把666替换成999
   - 获取"yuan"的索引
   - 假设不知道前面有几个元素，分片得到最后的三个元素
4. 查找列表中元素，移除每个元素的空格，并查找以a或A开头并且以c结尾的所有元素。
   - li = ["alec", " aric", "Alex", "Tony", "rain"]
   - tu = ("alec", " aric", "Alex", "Tony", "rain")
   - dic = {'k1': "alex", 'k2': ' aric', "k3": "Alex", "k4": "Tony"}
5. 求100以内不能被3整除的所有数，并把这些数字放在列表里，并求出这些数字的总和和平均数。

课堂练习（7）

1. 计算求100以内的质数和。

2. 利用for循环和range输出9 * 9乘法表

3. 有四个数字：1、2、3、4，能组成多少个互不相同且无重复数字的三位数？各是多少？

列表生成式

https://blog.csdn.net/weixin_43936969/article/details/103721875

- 列表生成式：python内置非常简单却强大的可以用来创建list的生成式，列表生成式也可以叫做列表解析。
- 列表生成式的格式：[expression for ele in iterator if eval(ele)] == 表达式+循环+条件运用列表生成式，可以写出非常简洁的代码。
- 一般情况下循环太繁琐，而列表生成式则可以用一行语句代替多行循环生成列表。

1、成一个列表,列表元素分别为[1 ** 1,2 ** 2,...,9 ** 9]

```
import math
li = []
for i in range(1, 10):
    li.append(i ** i)
print(li)
print([i ** i for i in range(1, 10)])
```

列表生成式

```
print([i ** i for i in range(1, 10)])
print([i_** i for i in range(1, 10) if i % 2 == 0])
```

2、找出1～10之间的所有偶数

```
print([i for i in range(1, 11) if i % 2 == 0])
```

6.列表的字符串的大写改成小写，不是字符串的去掉

```
li = ['hello','World',24,24,41,7,8,False,'Apple']
print([s.lower() for s in li if isinstance(s,str)])
```

列表生成式-用例

li = [f(val) for val in iterator if cond(val)]

问题： 如何生成1-10之间所有偶数形成的list

普通做法

```
li = []
for i in range(10):
    if i >=1 and i % 2 == 0:
        li.append(i)
print(li)
```

Python做法

```
li = [ val for val in range(10) if val >=1 and val % 2 == 0]
print(li)
```

问题：如何生成1-10之间所有偶数形成的list，再取平方值

普通做法

```
li = []
for i in range(10):
    if i >= 1 and i % 2 == 0:
        li.append(i*i)
print(li)
```

Python做法

```
li = [i*i for i in range(10) if i >= 1 and i % 2 == 0]
print(li)
```

列表生成式-用例

问题： 判断某用户是否存在于某用户列表中

```
users = [
    {"name": "张三", "password": 'zhangsan', ‘age’ : 11},
    { “name” : “李四” , “password” : ‘lisi’ , ‘age': 22},
    {"name": "王五", "password": 'wangwu', ‘age': 33},
]
```

普通做法

```
name = 'zhangsan'
user = None
for u1 in users:
    if name == u1['name']:
        user = u1
        break
if user is None:
    print("您不是已有用户，请注册！")
    exit()
```

Python做法

```
name = 'zhangsan'
find_user = [u1 for u1 in users if u1['name'] == name]
user = find_user[0] if len(find_user) > 0 else None
if user is None:
    print("找不到用户", name)
    exit()
```

语法糖-列表生成式

需求：列表a=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9],要求你把列表里的每个值加1

普通青年版

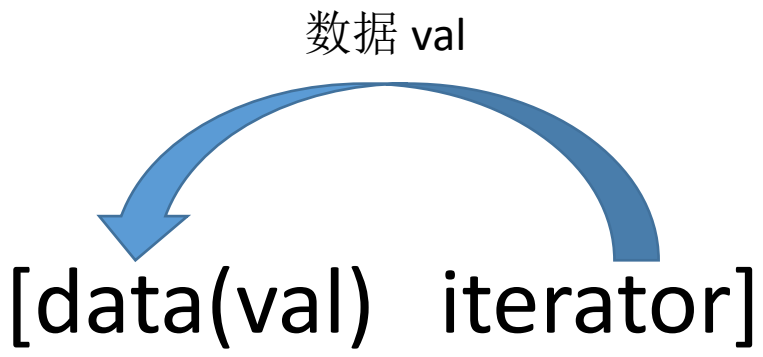
```
a = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
for i in range(0, len(a)):
    a[i] += 1
print(a)
```

文艺青年版

```
a = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
a = list(map(lambda x: x+1, a))
print(a)
```

python青年版

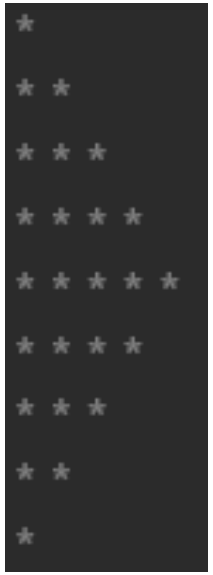
```
a = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
a = [i+1 for i in a]
print(a)
```



思考：
能否生成tuple/set?

课堂练习（8）-列表生成式

1. 列表a=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]，要求把列表里的每个值加1
2. 成一个列表,列表元素分别为[1 ** 1,2 ** 2,…,9 ** 9]
3. 找出1～10之间的所有偶数
4. 如何生成1-10之间所有偶数形成的list，再取平方值
5. 列表的字符串的大写改成小写，不是字符串的元素去掉
6. 输出99乘法表
7. 找出1～100之内的所有素数
8. 输出 *号（[*,**,***,****,*****,****,***,**,*]）
9. 判断某用户是否存在于某用户列表中



```
users = [
   {"name": "张三", "password": 'zhangsan', ‘age’: 11},
   {“name”: “李四” , “password”: ‘lisi’, ‘age': 22},
   {"name": "王五", "password": 'wangwu', ‘age': 33},
]
```

课堂练习（9）- 输出*号

```
# [22]  使用while,完成以下图形的输出
# *
# * *
# * * *
# * * * *
# * * * * *
# * * * *
# * * *
# * *
# *
```

```
# 可用
# li = list(range(1,6))+list(range(1,5))[::-1]
# print(li)

# [print('*'*i) for i in li]
# for i in li:
#     print('*' * i)

# [print('*' * i) for i in li]

# a = [print('*' * i) for i in list(range(1,6))+list(range(1,5))[::-1]]
# print(a)

# def local_print(k):
#     print('*' * k)
#     return k * k

# a = [local_print(i) for i in list(range(1,6))+list(range(1,5))[::-1]]
# print(a)
# [func logic]
```

课堂练习（10）- 输出99乘法表

```
# 练习2：生成 9/9 乘法表

# 方式1：
# for i in range(1, 10):
#     for j in range(i, 10):
#         print("%s * %s = %s" %(i, j, i * j))

# for i in range(1, 10):
#     for j in range(i, 10):
#         print("%s * %s = %s" %(i, j, i * j), end= ' ' if j < 9 else '\n')
#
# for i in range(1, 10):
#     for j in range(i, 10):
#         pass


# 方式3：
# [print("%s * %s = %s" % (i, j, i * j), end=' ' if j < 9 else '\n') for i in range(1, 10) for j in range(i, 10)]
# [print("%s * %s = %s" %(i, j, i * j), end= ' ' if j < 9 else '\n') for i in range(1, 10) for j in range(i, 10)]
```

课堂练习（11）- 打印100以内质数

```
# 打印100以内的素数之和。

# i = 17
# a = [i % j == 0 for j in range(2, int(i / 2) + 1)]
# print(a)
# print(i, sum(a))

# 方法1：带函数
# 如果判断一个数是质数：
# def is_prime(i):
#     return sum([i % j == 0 for j in range(2, int(i / 2) + 1)]) == 0
# 从列表中取出质数，形成列表生成式
# a = [i for i in range(2, 101) if is_prime(i)]
# 打印输出
# print(a, sum(a), sep='\n')

# 方法2：把函数展开形成嵌套列表生成式
# a = [i for i in range(2, 101) if sum([i % j == 0 for j in range(2, int(i / 2) + 1)]) == 0]
# print(a, sum(a), sep='\n')
```