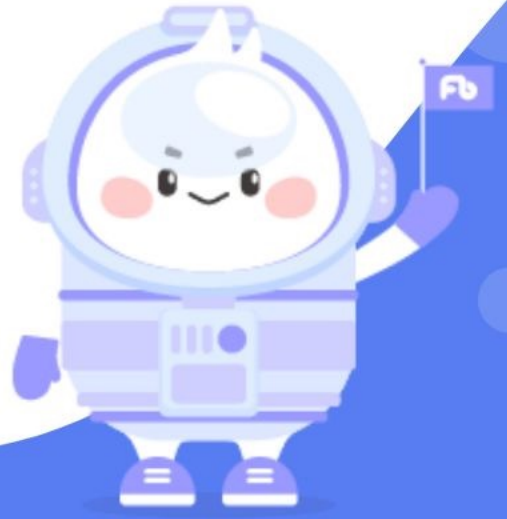
粉笔

2023上粉笔教资

《信息技术》

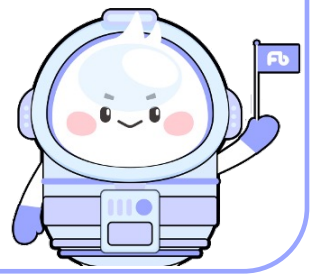
Python程序设计 3/4

讲师：孙珍珍



更多干货关注 粉笔教师教育 粉笔教师

❀ 复习一下

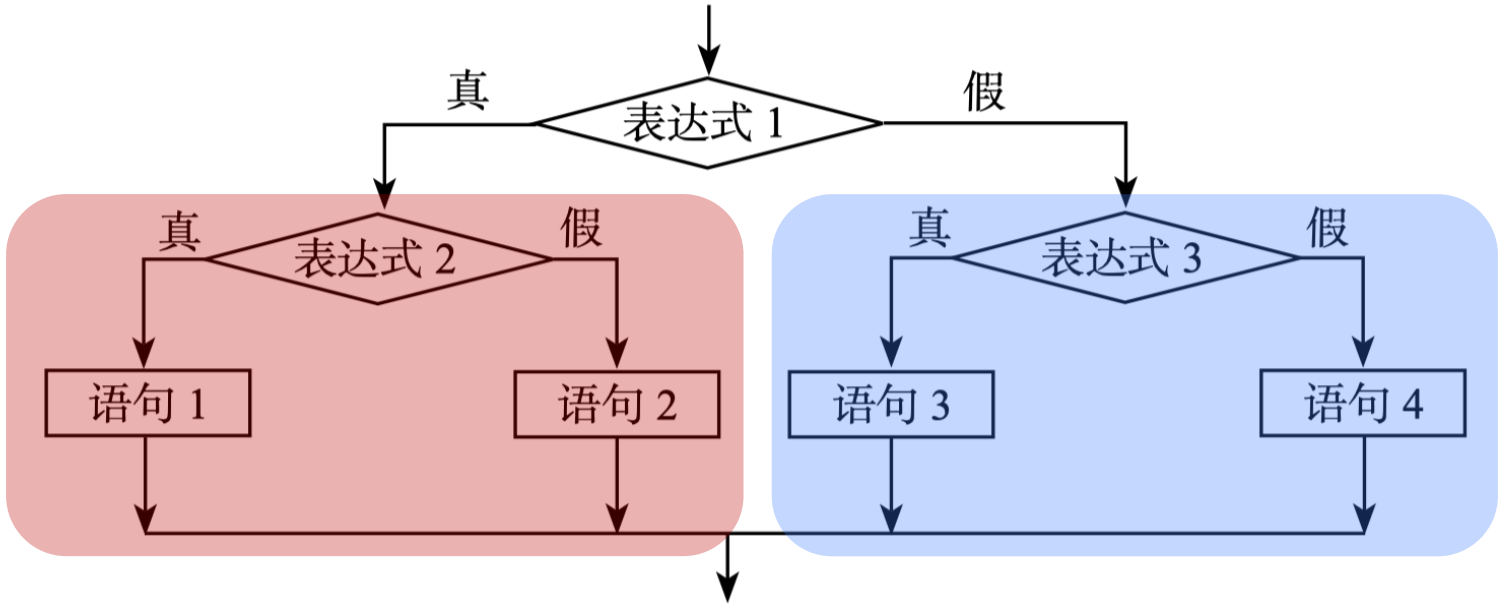


四、嵌套的if函数

P360

粉笔

基本格式	流程图

if 判断条件1:
　if 判断条件2:
　　语句块1
　else:
　　语句块2
else:
　if 判断条件3:
　　语句块3
　else:
　　语句块4
	

说明：配对原则，看缩进

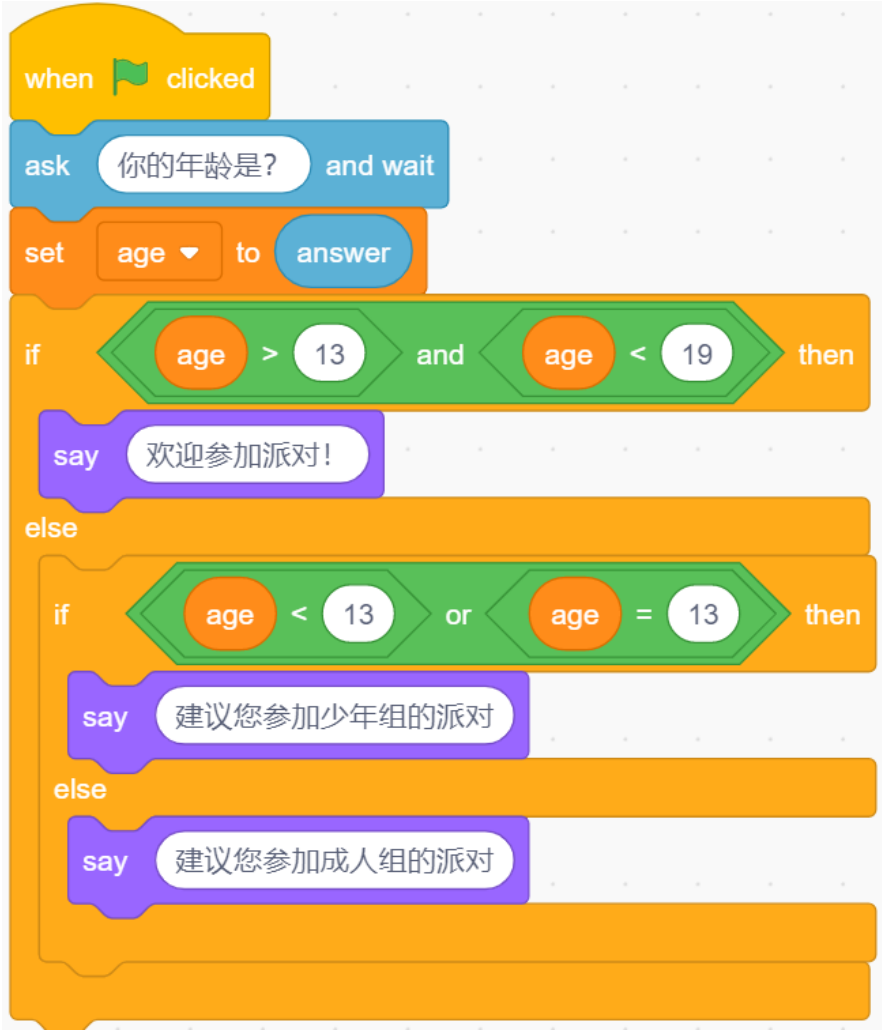
（2022下·初中）如下图所示是根据年龄对来宾是否准入派对的程序段，能够让系统提示“欢迎参加派对”的年龄是（　　）。

A.11

B.13

C.15

D.19



四、嵌套的if函数

P360

粉笔

基本格式	实例

if 判断条件1:
if 判断条件2:
语句块1
else:
语句块2
else:
if 判断条件3:
语句块3
else:
语句块4
	【例4】编写程序，实现输入三个整数，输出最大值。
	分析：

四、嵌套的if函数

P360

粉笔

实例

【例4】编写程序，实现输入三个整数，输出最大值。

```
a = eval(input("请输入a的值:"))
b = eval(input("请输入b的值:"))
c = eval(input("请输入c的值:"))
if a > b:
    if a > c:
        max1 = a
    else:
        max1 = c
else:
    if b > c:
        max1 = b
    else:
        max1 = c
print("最大值为: ", max1)
```

```
请输入a的值:5
请输入b的值:9
请输入c的值:3
最大值为:  9
```

```
1  a = eval(input("请输入a的值:"))
2  b = eval(input("请输入b的值:"))
3  c = eval(input("请输入c的值:"))
4  if a > b:
5      max1 = a
6  else:
7      max1 = b
8  if max1 < c:
9      max1 = c
10 print("最大值为: ", max1)
```

粉笔教师

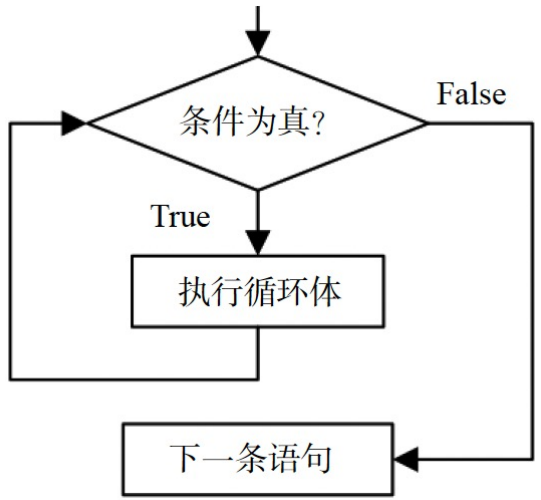
第三节循环结构程序设计

一、while 循环

P361

粉笔

基本格式	实例
while 判断条件: 语句块 后续语句	说明：



一、while 循环

P361

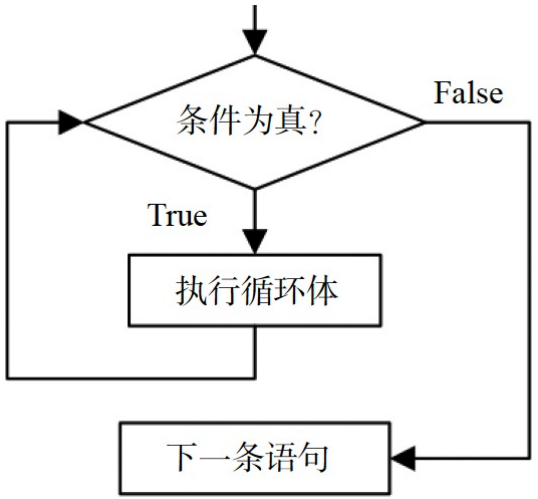
粉笔

基本格式

while 判断条件:

语句块

后续语句



说明：

①使用场景：条件为真则重复执行

②相关语句：while前有初始状态语句

while缩进中有循环状态变化语句

③作用范围：冒号和缩进

- 真，执行缩进的语句块，然后返回继续判断条件
- 假，执行后续语句

实例

【例1】编写程序，求s=1+2+3+...+100的值。

分析：

一、while 循环

P362

粉笔

基本格式

while 判断条件:

　　语句块

后续语句

说明：

①使用场景：条件为真则重复执行

②相关语句：while前有初始状态语句

　　while缩进中有循环状态变化语句

③作用范围：冒号和缩进

- 真，执行缩进的语句块，然后返回继续判断条件
- 假，执行后续语句

实例

【例1】编写程序，求s=1+2+3+...+100的值。

```
1    i = 1
2    s = 0
3    while i <= 100:
4        s += i
5        i += 1
6    print("1+2+3+…+100=", s)
```

```
1+2+3+…+100= 5050
```

二、for循环

P362

粉笔

1.序列循环，基本格式	2.搭配range()函数，基本格式
for 变量 in 序列: 语句块	for 变量 in range([start,] stop [,step]): 语句块
	说明： ①start：开始值，默认从0开始，0可省 ②stop：结束值，但不包含stop ③step：步长，默认为1，1可省 ④执行过程：在[start,stop-1]范围内，则循环

二、for循环

P363

粉笔

2.搭配range()函数，基本格式	实例
for 变量 in range([start,] stop [,step]): 语句块 后续语句	【例2】用for语句求s=1+2+3+…+100的值。 分析：
说明： ①start：开始值，默认从0开始，0可省 ②stop：结束值，但不包含stop ③step：步长，默认为1，1可省 ④执行过程：在[start,stop-1]范围内，则循环	

二、for循环

2.搭配range()函数，基本格式

for 变量 in range([start,] stop [,step]):

语句块

后续语句

说明：

①start：开始值，默认从0开始，0可省

②stop：结束值，但不包含stop

③step：步长，默认为1，1可省

④执行过程：在[start,stop-1]范围内，则循环

实例

【例2】用for语句求s=1+2+3+…+100的值。

```
1    s = 0
2    for i in range(1, 101):
3        s += i
4    print("1+2+3+…+100=", s)
```

```
1+2+3+…+100= 5050
```

（2021下·高中）运行如图的Python程序段，可以统计1，2，3，4四个数字能够组成多少个互不相同且无重复数字的三位数。横线处应填语句是（　　）。

A.in range(1,4)

B.in range(1,5)

C.in random.uniform(1,4)

D.in random.uniform(1,5)

```
total = 0
for i _____:
    for j _____:
        for k _____:
            if (i != j) and (j != k) and (k != i):
                print(i, j, k)
                total += 1
print(total)
```

三、循环嵌套

P364

粉笔

【例3】编写一个程序，输出九九乘法表。

```
1 * 1 = 1
1 * 2 = 2   2 * 2 = 4
1 * 3 = 3   2 * 3 = 6   3 * 3 = 9
1 * 4 = 4   2 * 4 = 8   3 * 4 = 12   4 * 4 = 16
1 * 5 = 5   2 * 5 = 10   3 * 5 = 15   4 * 5 = 20   5 * 5 = 25
1 * 6 = 6   2 * 6 = 12   3 * 6 = 18   4 * 6 = 24   5 * 6 = 30   6 * 6 = 36
1 * 7 = 7   2 * 7 = 14   3 * 7 = 21   4 * 7 = 28   5 * 7 = 35   6 * 7 = 42   7 * 7 = 49
1 * 8 = 8   2 * 8 = 16   3 * 8 = 24   4 * 8 = 32   5 * 8 = 40   6 * 8 = 48   7 * 8 = 56   8 * 8 = 64
1 * 9 = 9   2 * 9 = 18   3 * 9 = 27   4 * 9 = 36   5 * 9 = 45   6 * 9 = 54   7 * 9 = 63   8 * 9 = 72   9 * 9 = 81
```

分析：

三、循环嵌套


```
print("%d * %d = %2d" % (y, x, x * y), end='    ')
```

```
print("{} * {} = {:2}".format(y, x, x * y), end='    ')
```

```
1 * 1 =  1
1 * 2 =  2   2 * 2 =  4
1 * 3 =  3   2 * 3 =  6   3 * 3 =  9
1 * 4 =  4   2 * 4 =  8   3 * 4 = 12   4 * 4 = 16
1 * 5 =  5   2 * 5 = 10   3 * 5 = 15   4 * 5 = 20   5 * 5 = 25
1 * 6 =  6   2 * 6 = 12   3 * 6 = 18   4 * 6 = 24   5 * 6 = 30   6 * 6 = 36
1 * 7 =  7   2 * 7 = 14   3 * 7 = 21   4 * 7 = 28   5 * 7 = 35   6 * 7 = 42   7 * 7 = 49
1 * 8 =  8   2 * 8 = 16   3 * 8 = 24   4 * 8 = 32   5 * 8 = 40   6 * 8 = 48   7 * 8 = 56   8 * 8 = 64
1 * 9 =  9   2 * 9 = 18   3 * 9 = 27   4 * 9 = 36   5 * 9 = 45   6 * 9 = 54   7 * 9 = 63   8 * 9 = 72   9 * 9 = 81
```

三、循环嵌套

P364

粉笔

【例3】编写一个程序，输出九九乘法表。

```
for x in range(1, 10, 1):
    for y in range(1, x + 1, 1):
        print("%d * %d = %2d" % (y, x, x * y), end='    ')
        # print("{} * {} = {:2}".format(y, x, x * y), end='    ')
    print()
```

```
1 * 1 =  1
1 * 2 =  2   2 * 2 =  4
1 * 3 =  3   2 * 3 =  6   3 * 3 =  9
1 * 4 =  4   2 * 4 =  8   3 * 4 = 12   4 * 4 = 16
1 * 5 =  5   2 * 5 = 10   3 * 5 = 15   4 * 5 = 20   5 * 5 = 25
1 * 6 =  6   2 * 6 = 12   3 * 6 = 18   4 * 6 = 24   5 * 6 = 30   6 * 6 = 36
1 * 7 =  7   2 * 7 = 14   3 * 7 = 21   4 * 7 = 28   5 * 7 = 35   6 * 7 = 42   7 * 7 = 49
1 * 8 =  8   2 * 8 = 16   3 * 8 = 24   4 * 8 = 32   5 * 8 = 40   6 * 8 = 48   7 * 8 = 56   8 * 8 = 64
1 * 9 =  9   2 * 9 = 18   3 * 9 = 27   4 * 9 = 36   5 * 9 = 45   6 * 9 = 54   7 * 9 = 63   8 * 9 = 72   9 * 9 = 81
```

（2021下·高中）我国古代数学家张丘建在《算经》中提出了一个著名的数学问题：鸡翁一值钱五，鸡母一值钱三，鸡雏三值钱一。百钱买百鸡，问鸡翁、鸡母、鸡雏各几何？请使用C语言或者Python语言编写程序，输出全部可能的解。

【参考答案】

```
1    for jw in range(1, 20):
2        for jm in range(1, 33):
3            jc = 100 - jw - jm
4            if 5 * jw + 3 * jm + 1 / 3 * jc == 100:
5                print("鸡翁可为：%d，鸡母可为：%d，鸡雏可为：%d。" % (jw, jm, jc))
```

```
鸡翁可为：4，鸡母可为：18，鸡雏可为：78。
鸡翁可为：8，鸡母可为：11，鸡雏可为：81。
鸡翁可为：12，鸡母可为：4，鸡雏可为：84。
```

四、转移语句

1.break 语句

- 提前结束整个循环。通常与if语句一起使用。

2.continue语句

- 提前结束本次循环。通常与if语句一起使用。

四、转移语句

P364

（1）break 语句	（2）continue语句
【例-书上无】输出 1 ~ 10 以内的第一个奇数。	【例-书上无】输出 1 ~ 10 以内的所有奇数。

四、转移语句

P364

粉笔

（1）break 语句

【例-书上无】输出 1 ~ 10 以内的第一个奇数。

```
1    for i in range(1, 11):
2        if i % 2 != 0:
3            print('第一个奇数是: ', i)
4            break
```

```
第一个奇数是:  1
```

（2）continue语句

【例-书上无】输出 1 ~ 10 以内的所有奇数。

```
1    print("奇数有: ", end='')
2    for i in range(1, 11):
3        if i % 2 == 0:
4            continue
5        print(i, end='   ')
```

```
奇数有: 1   3   5   7   9
```

（2019下·高中）《孙子算经》是中国古代重要的数学著作，该著作卷下第26题：“今有物不知其数，三三数之剩二，五五数之剩三，七七数之剩二，问物几何？”《孙子算经》不但提供了答案，而且给出了解法。请编程求卷下第26题的最小正整数解。

【参考答案】

```
x = 1
while True:
    if x % 3 == 2 and x % 5 == 3 and x % 7 == 2:
        print("最小正整数解为: ",x)
        break
    x = x + 1
```

```
最小正整数解为:  23
```

第四节 序列

一、认识序列

P366

粉笔

```
>>> a = "hi~123"
>>> a
'hi~123'
>>> a[4]
'2'
```

```
>>> b = ['hi','~',1,2.3]
>>> b
['hi', '~', 1, 2.3]
>>> b[3]
2.3
```

```
>>> c = 'hi','~',1,2.3
>>> c
('hi', '~', 1, 2.3)
>>> c[3]
2.3
```

字符串（str）

①引号括起来

②元素之间无分隔

③元素类型均为字符

④元素内容不可变化

列表（list）

①中括号括起来

②元素之间逗号分隔

③元素类型可以不同

④元素内容可修改

元组（tuple）

①圆括号括起来

②元素之间逗号分隔

③元素类型可以不同

④元素内容不可变化

二、通用操作

	含义	实现方法	实例

1.索引	查找某一个元素	序列名[下标]
		左下标从0开始
		右下标从-1开始
			`>>> s = 'hi 123'`
			`>>> s[3]` → `'1'`
			`>>> s[-2]` → `'2'`

3.相加	实现序列的拼接	“+”号	`>>> 'hi,'+'Python'` `'hi,Python'`
4.相乘	实现序列的复制	“*”号	`>>> 'hi,'*3` `'hi,hi,hi,'`

二、通用操作

操作	含义	实现方法
2.分片	查找某一段元素	序列名[start:end:step] start：开始下标，0可省 end：结束下标【不含】，尾可省 step：步长，1可省

```
s1 = ['a', 'b', 'c', 'd', 1, 2, 3]
print("1.", s1[0:7:1])
print("2.", s1[1:3])
print("3.", s1[1:])
print("4.", s1[:3])
print("5.", s1[::2])
print("6.", s1[::-1])
print("7.", s1[3::-1])
```

```
1. ['a', 'b', 'c', 'd', 1, 2, 3]
2. ['b', 'c']
3. ['b', 'c', 'd', 1, 2, 3]
4. ['a', 'b', 'c']
5. ['a', 'c', 1, 3]
6. [3, 2, 1, 'd', 'c', 'b', 'a']
7. ['d', 'c', 'b', 'a']
```

粉笔教师

-- 讲义纠错 P368 --

【例】创建序列：s1= 'abcdefg'，使用分片获取情况如下表所示。

分片方式	描述	结果
s1[0:7:1]	获取序列 s1 中所有元素	'abcdefg'
s1[1:3]	获取 s1 中从索引 1 到索引 3 之间的所有元素	'bc'
s1[1:]	获取 s1 中从索引 1 开始到最后一个索引之间的所有元素	'bcdefg'
s1[:3]	获取 s1 中从索引 0 到索引 3 之间的所有元素	'abc'
s1[0:7:2] 或 s1[::2]	将步长设置为 2，获取 s1 从开始到结束的元素	'aceg'
s1[7:0:-1]	获取 s1 中从索引 7 到索引 0 之间的所有元素	'gfedcb'

二、通用操作

5.	含义	实例

len()	返回序列中元素的长度（个数）	`>>> s2=(1,2,'a','b')`
		`>>> len(s2)`
		`4`
		
		`>>> s3='12ab'`
		`>>> len(s3)`
		`4`

min()	查找序列中最小的元素
	（序列中类型要相同）
		`>>> s4 = '123jyALG'`
		`>>> min(s4)`
		`'1'`
		`>>> max(s4)`
		`'y'`
		
		`>>> s5 = [4, 2.3, 1, 3]`
		`>>> min(s5)`
		`1`
		`>>> max(s5)`
		`4`

max()	查找序列中最大的元素 （序列中类型要相同）	

6.查找

含义	在序列中查找子序列；存在，返回位置；不存在，抛出异常	
格式	序列名.index(sub,start,end)	①sub：要查找的子序列 ②start：查找的开始位置，默认从下标0开始，可省 ③end：查找的结束位置【不含】，默认到序列的结尾，可省
实例		

```
>>> s4 = '123jyALG'
>>> s4.index('j')
3
```

```
>>> s4='123jyALG'
>>> s4.index('jy',3,6)
3
```

```
>>> s4='123jyALG'
>>> s4.index('JY',3,6)
Traceback (most recent ca
  File "<input>", line 1,
ValueError: substring not
```

7.计数

含义	统计子序列出现的次数	
格式	序列名.count(sub,start,end)	①sub：要查找的子序列 ②start：查找的开始位置，默认从下标0开始，可省 ③end：查找的结束位置【不含】，默认到序列的结尾，可省
实例		

```
>>> s4 = '123jyALG'
>>> s4.count('jy')
1
```

```
>>> s4 = '123jyALG'
>>> s4.count('jy',2,5)
1
```

```
>>> s4 = '123jyALG'
>>> s4.count('jy',2,4)
0
```

8.遍历

（1）while循环

借助索引和序列总长度依次遍历

```
1    lb = [123, 12.3, 'jy', 'alg']
2    i = 0
3    while i < len(lb):
4        print(lb[i])
5        i = i + 1
```

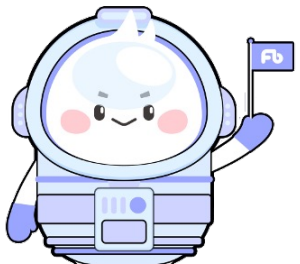
```
123
12.3
jy
alg
```

（2）for循环

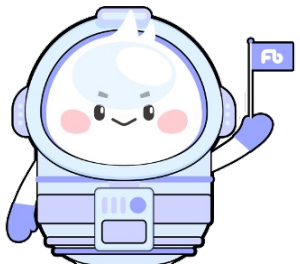
借助for循环的格式

```
1    lb = [123, 12.3, 'jy', 'alg']
2    for i in lb:
3        print(i)
```

- 3种
  - 字符串str —— ____括起来 —— 元素间____分隔 —— 元素类型为____ —— 元素内容不可变
  - 列表list —— ____括起来 —— 元素间____分隔 —— 元素类型可不同 —— 元素内容可____
  - 元组tuple —— ____括起来 —— 元素间____分隔 —— 元素类型可不同 —— 元素内容不可变
- 通用
  - 拼接：_____ —— 复制：_____
  - 长度：_____ —— 最大值：_____ —— 最小值：_____
  - 索引：名_下标__ —— 分片：名[_____]
  - 查找：名._____(sub) —— 计数：名._____(sub,start,end)
  - 遍历：while和for均可



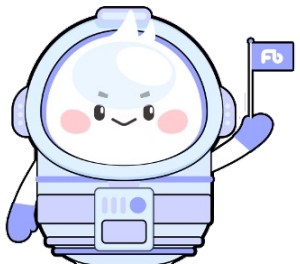
- 3种
  - 字符串str — 引号括起来 — 元素间无分隔 — 元素类型为字符 — 元素内容不可变
  - 列表list — 中括号括起来 — 元素间逗号分隔 — 元素类型可不同 — 元素内容可修改
  - 元组tuple — 圆引号括起来 — 元素间逗号分隔 — 元素类型可不同 — 元素内容不可变
- 通用
  - 拼接：+ — 复制：*
  - 长度：len — 最大值：max — 最小值：min
  - 索引：名[下标] — 分片：名[start:end:step]
  - 查找：名.index(sub) — 计数：名.count(sub,start,end)
  - 遍历：while和for均可



有疑问没？等你吖

下节内容

P370 ~ P382



粉笔教师

岸上等你

THE TEST

粉笔
教师

光芒万丈
不负理想