

# 第10课 工作量计时器

## 一、课程结构导图

风变科技

风变科技

风变科技

风变科技

风变科技

风变科技

风变科技

风变科技

风变科技

风变科技



注：▲为重点知识点。

## 二、项目讲解

### 2.1 明确项目需求

目的：

- 1. 已知项目的总工时（项目大小×80），需要完成项目的人工数量，求完成项目所需的时间（工时计算）。
- 2. 已知项目的总工时（项目大小×80），需要完成项目的时间，求完成项目所需的人数（人力计算）。

图示：

总工时 = 人数 × 时间



size × 80 = number × time

## 2.2 项目代码及解析

**目的1：** 使用函数，实现人力计算和工时计算的功能。

**关键代码1：**

```
1 # 工时计算
2 def estimated_time(size,number):
3     time = size * 80 / number
4     print('项目大小为%.1f个标准项目，使用%d个人力完成，则需要工时数量为：%.1f个' %(size,number,time))
5
6 #人力计算
7 def estimated_number(size,time):
8     number = size * 80 / time
9     print('项目大小为%.1f个标准项目，如果需要在%.1f个工时完成，则需要人力数量为：%d人' %(size,time,number))
10
11 # 调用工时计算
12 estimated_time(1.5,2)
13
14 # 调用人力计算
15 estimated_number(0.5,20)
```

**代码解析：**

1. 无论是调用工时计算还是人力计算函数，都需要给对应的函数参数进行参数值传递，例如代码一中，参数赋值结果为size=1.5,number=2；代码二中，参数赋值结果为size=0.5，time=20。
2. 第4行和第9行代码中，%.1f意思是格式化字符串为浮点型，并保留一位小数。

**目的2：**人力计算中，计算结果中人数出现小数，与实际情况不符，需对人数实现向上取整操作。

**关键代码2：**

```
1 # 人力计算
2 def estimated_number(size,time):
3     if (size * 80 % time) != 0:
4         number = int(size * 80 / time) + 1 # number向上取整
5     else:
6         number = size * 80 / time
7     print('项目大小为%.1f个标准项目，如果需要在%.1f个工时完成，则需要人力数量为：%d人' %(size,time,number))
8
9 # 调用人力计算函数
10 estimated_number(1,60)
```

**代码解析：**代码先将(1,60)这两个参数值传递给size和time。通过if...else语句进行判断，当计算结果是小数时，执行if语句，将number的数据类型，从浮点型转换为整数型，实现了取整功能；后续执行加1操作，这两个步骤共同实现了向上取整功能。若计算结果是整数时，则执行else语句。

**目的3：**实现交互功能，让用户自行选择计算方式并打印计算结果。

**关键代码3：**

```
1 import math
2
3 def estimated(size=1,number=None,time=None):
4     if (number == None) and (time != None):
5         number = math.ceil(size * 80 / time)
6         print('项目大小为%.1f个标准项目，如果需要在%.1f个工时完成，则需要人力数量为：%d人' %(size,time,number))
7     elif (number != None) and (time == None):
8         time = size * 80 / number
9         print('项目大小为%.1f个标准项目，使用%d个人力完成，则需要工时数量为：%.1f个' %(size,number,time))
10
11 choice = input('请选择计算类型：（1-人力计算，2-工时计算）')
12 if choice == '1':
13     size = float(input('请输入项目大小：（1代表标准大小，可以输入小数）'))
14     number = None
15     time = float(input('请输入工时数量：（可以输入小数）'))
16     estimated(size,number,time)
17
18 elif choice == '2':
19     size = float(input('请输入项目大小：（1代表标准大小，可以输入小数）'))
20     number = int(input('请输入人力数量：（请输入整数）'))
21     time = None
22     estimated(size,number,time)
```

→ 执行计算方式

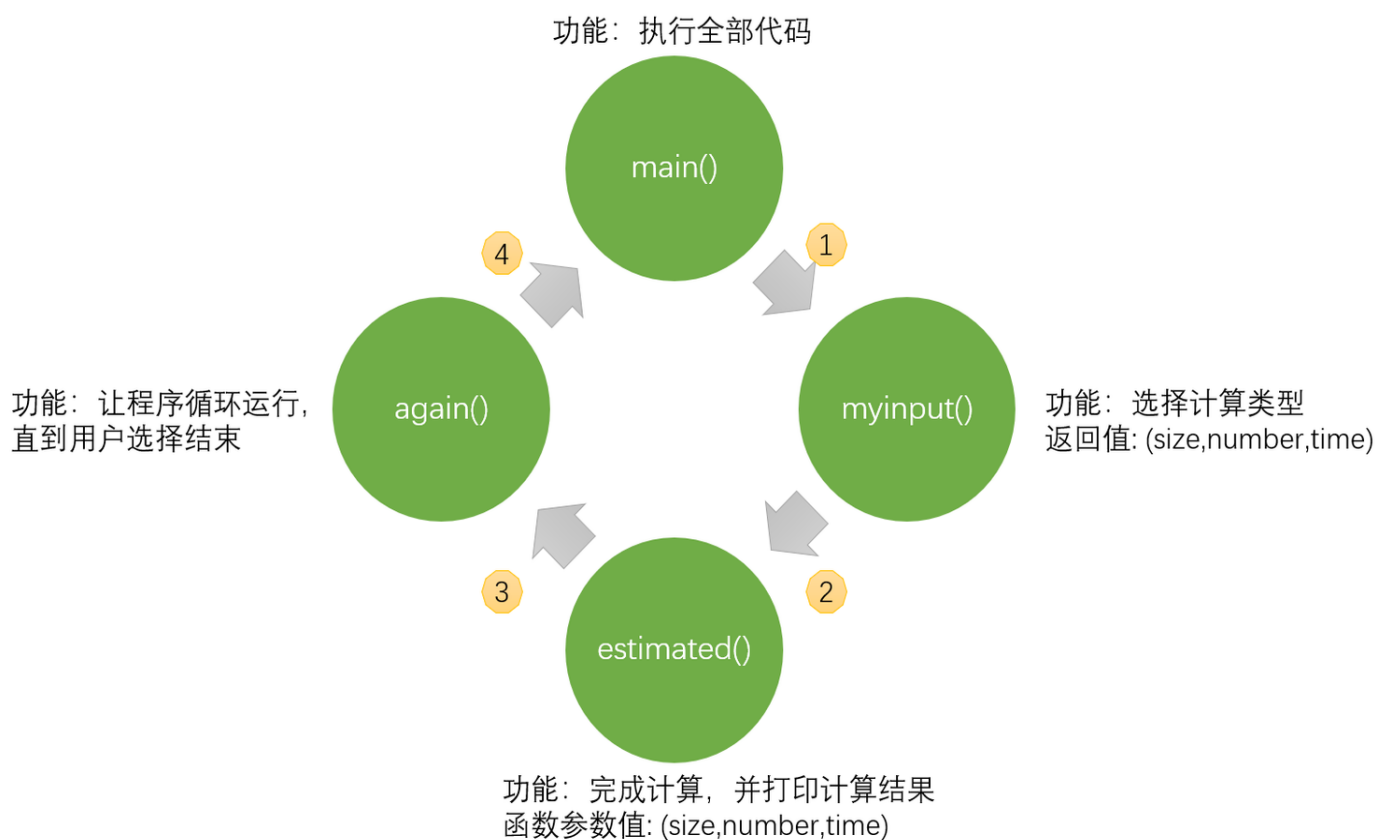
用户选择计算方式，调用对应的计算函数

**代码解析：**

1. 由于第11行代码中的choice变量是**字符串类型**，所以第12行和第18行代码在执行条件判断时，1和2需要添加引号（'1'和'2'），if语句和elif语句才会判断成立，否则if和elif语句下的代码不会执行。
2. 第15行和第20行代码中，由于time变量和number变量都赋予了值，所以在调用estimated()函数时，会将第3行代码中原来的number参数值或time参数值进行替换。
3. 在四则运算中，**整数和浮点数可以进行四则计算，但字符串不可以**。所以size, number, time这三个变量需要进行数据类型转换。

**目的4：** 使用函数进行封装，并添加again函数，代码循环运行，直到用户选择停止操作。

**图示：**



**关键代码1：**

```
1 import math
2 def myinput():
```

```
3     choice = input('请选择计算类型: (1-人力计算, 2-工时计算) ')
4     if choice == '1':
5         size = float(input('请输入项目大小: (1代表标准大小, 可以输入小数) '))
6         number=None
7         time = float(input('请输入工时数量: (可以输入小数) '))
8         return size,number,time
9
10    elif choice=='2':
11        size = float(input('请输入项目大小: (1代表标准大小, 可以输入小数) '))
12        number = int(input('请输入人力数量: (请输入整数) '))
13        time=None
14        return size,number,time
15
16    def estimated_number(my_input):
17        size = my_input[0]
18        number = my_input[1]
19        time = my_input[2]
20
21        if time==None and number!=None:
22            time = size * 80 / number
23            print('项目大小为%.1f个标准项目, 使用%d个人力完成, 则需要工时数量为: %.1f个' %(size,number,t
24
25        elif number==None and time!=None:
26            number = math.ceil(size * 80 / time)
27            print('项目大小为%.1f个标准项目, 如果需要在%.1f个工时完成, 则需要人力数量为: %d人' %(size,ti
28
29    def main():
30        my_input=myinput()
31        estimated_number(my_input)
32    main()
```

### 代码解析:

1. 第8行, 第14行代码, myinput()函数的返回值是一个元组, 第30行代码, 将myinput()函数返回值存放到my\_input变量中。所以my\_input=(size,number,time), 并作为estimated(my\_input)函数的参数值。
2. 第17, 18, 19行代码是元组的取值操作, 分别将my\_input变量中的值按索引值逐一取出。例如my\_input=(2,5,10), 那么my\_input[0]=2, my\_input[1]=5, my\_input[2]=10。
3. 第30行代码中, myinput()是一个函数, 而my\_input是一个变量, 两者要区别开来。

### 关键代码2:

```
1 key = 1 # 定义全局变量key
2
3 # 代码一
4 def again():
5     global key # 修改全局变量key
6     a = input('是否继续计算? 继续请输入y, 输入其他键将结束程序。')
7     if a != 'y':
8         key = 0 # 如果用户不输入'y', 则把key赋值为0
9
10 # 代码二
11 def main():
12     while key == 1: # 使用key的值来控制while循环是否执行
13         again()
14     print('感谢使用工作量计算小程序! ')
15
16 main()
```

#### 代码解析:

1. 代码一: 该代码块的功能是判断用户是否继续操作, 第5行代码中, global语句的作用是修改全局变量, 用户在输入提示下, 输入非y的关键字, key的值变为0。
2. 代码二: 该代码块的功能是运行所有其它函数代码, 第12行代码, 通过key的值来控制while循环是否执行, 当key的值为1时, 条件判断为while 1==1, 相当于是while True, 循环无限执行; 当key的值为0时, 条件判断为while 0==1, 相当于是while False, 循环条件不成立, 循环结束。
3. 代码一和代码二之间的联系: 变量key的值是控制while循环的‘开关’, 代码一通过global关键字对全局变量key的值进行修改, 进而影响代码二中while循环的判断条件。

## 三、知识点讲解

### 3.1 math模块 (ceil函数)

**概念:** ceil函数是用于整数型或浮点型数值进行向上取整操作。

**用法:** 当x为整数时, 返回的是等于x的整数值, 当x为小数时, 返回的是大于x的整数值。

### 示例：

```
1 # 代码一
2 import math
3 a = math.ceil(3) # 整数直接取整
4 print(a)
5
6 # 代码二
7 import math
8 a = math.ceil(3.2) # 小数先取整，再加1
9 print(a)
10
11 # 打印结果对比：
12 # 代码一打印结果为：3
13 # 代码二打印结果为：4
```

### 注意：

1. ceil函数与int函数区别在于，ceil函数是向上取整（先取整，再加一），例如math.ceil(3.20)，结果是4；而int函数是向下取整（只取整），例如int(3.20)，结果是3。
2. ceil函数的返回值是一个整数型的数值。

## 四、巩固练习

1. （单项选择题）以下哪个是代表保留两位小数（）？  
A. %f B. %.1f C. %.2f D. %.4f
2. （多项选择）关于int函数和ceil函数的说法，正确的是（）？  
A、int函数能对浮点数进行向下取整操作  
B、ceil函数能对浮点数进行向上取整操作  
C、import math ; print(math.ceil(2))，打印结果为3  
D、import math; print(math.ceil(2.1))，打印结果为3
3. （实操题）小明因为逃课，被老师逮到并罚抄课程，已知道罚抄的课程总共3856字，一张原稿纸能写400字。



题目要求：用python帮小明算一下罚抄完需要多少张原稿纸张（用两种方法解答）。

方法一：（用if...else语句条件判断）

方法二：（用ceil函数）