



贺江阳 按时通关

学号：24103219

分班：25春季

截止前完成关卡：6/6

最新完成关卡：6/6

完成效率：--

课堂最高完成效率：--

通关时间	计时规则 ?	实训总耗时 ?	评测次数	查重扣分	补交扣分	最终成绩	总评
2025-04-22 13:46	页面停留时长	8分 28秒	6	--	--	100.0/100.0	优秀

阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	列表的创建	2025-04-22 13:37	5	1	2025-04-22 13:38	1分 24秒	否	100/100	16.66/16.66	16.66
2	列表元素的访问	2025-04-22 13:38	7	1	2025-04-22 13:40	1分 35秒	否	100/100	16.66/16.66	16.66
3	列表元素的添加	2025-04-22 13:40	10	1	2025-04-22 13:42	1分 33秒	否	100/100	16.66/16.66	16.66
4	列表元素的排序	2025-04-22 13:42	4	1	2025-04-22 13:43	1分 27秒	否	100/100	16.66/16.66	16.66
5	列表元素的删除	2025-04-22 13:43	9	1	2025-04-22 13:44	1分 15秒	否	100/100	16.66/16.66	16.66
6	列表推导式	2025-04-22 13:45	2	1	2025-04-22 13:46	1分 14秒	否	100/100	16.70/16.7	16.70

图形统计

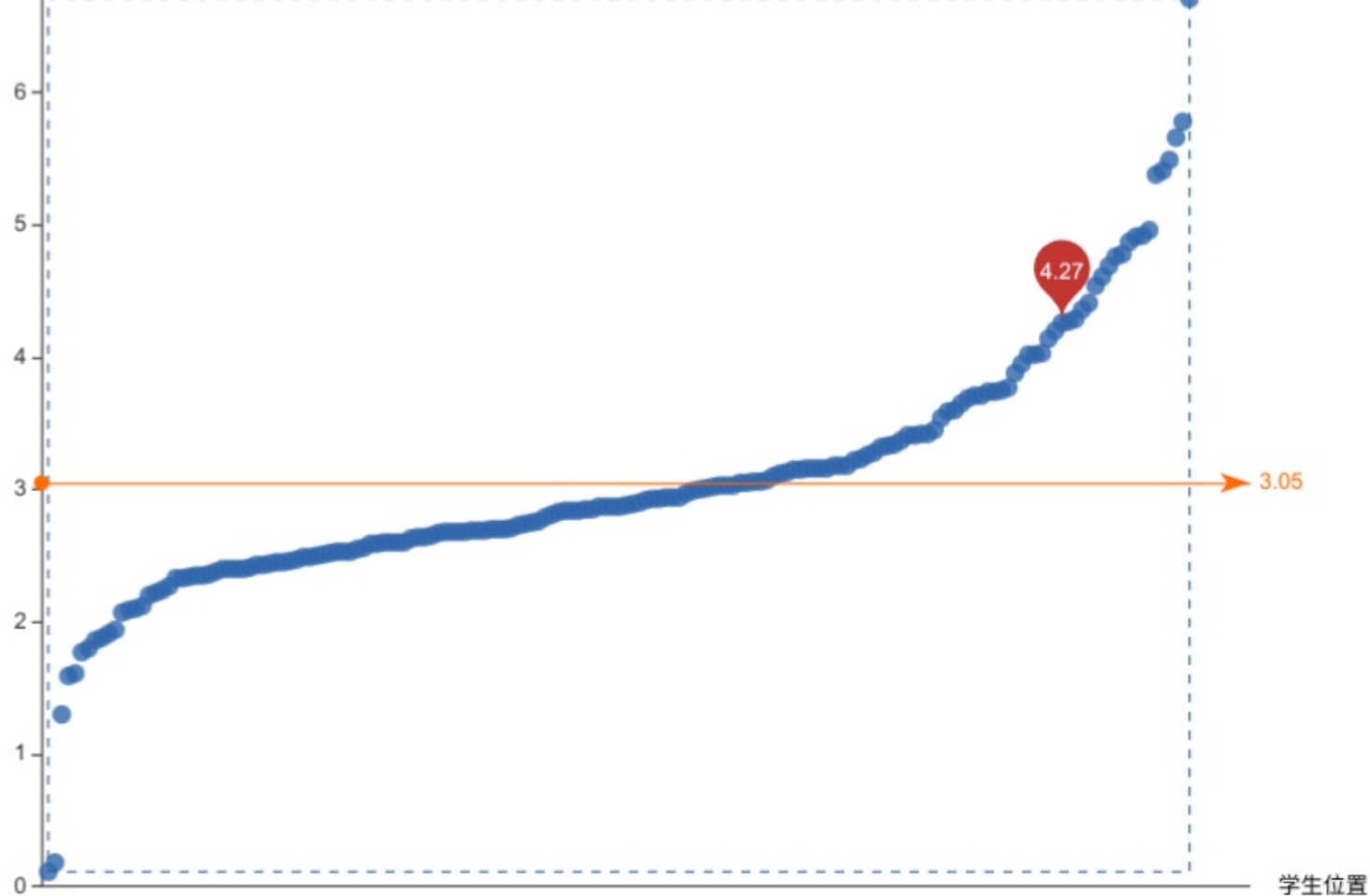
学习效率

学习效率=log(实训总得分/实训总耗时)

7

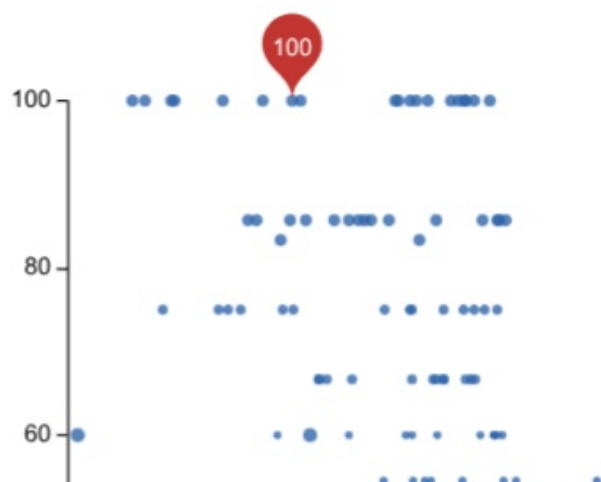
学习效率分布区间

姓名 贺江阳
学号 24103219
学习效率 4.27
当前排名 10



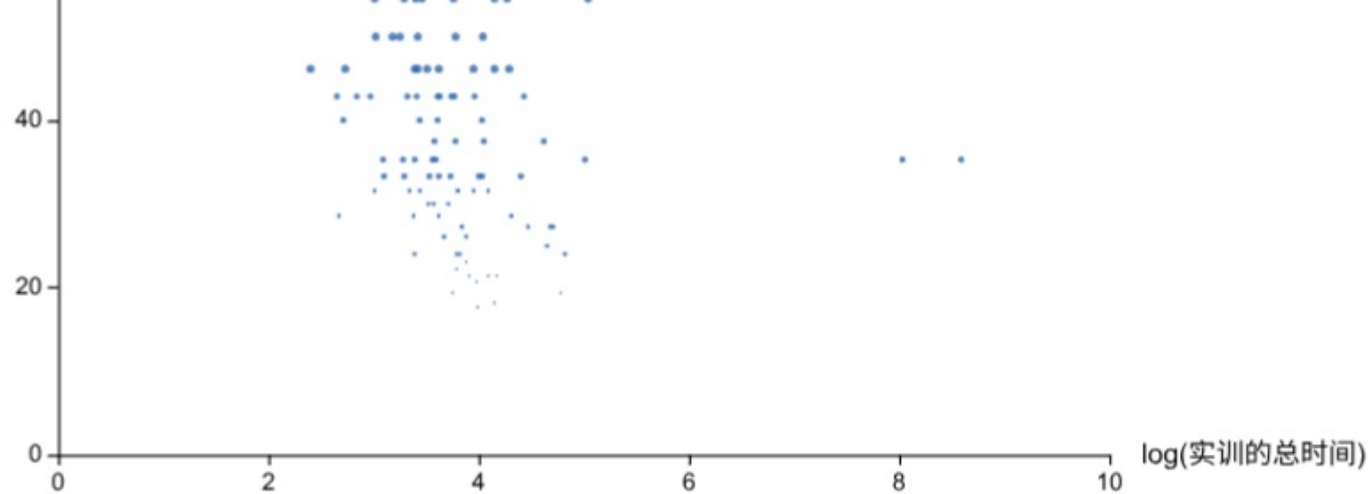
能力值

能力值（实训获得经验值/实训评测次数）



姓名	贺江阳
学号	24103219
能力	100

1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情



第1关 列表的创建

学员代码 [?](#)

代码修改行数 5

step1/step1.py



[复制代码](#)

```
1 my_str = "hello"
2 #####Begin#####
3 # 创建一个列表，并赋值："蒸羊羔儿"，"蒸熊掌"，"蒸鹿尾儿"，"烧花鸭"
4 food= ["蒸羊羔儿"，"蒸熊掌"，"蒸鹿尾儿"，"烧花鸭"]
5
6 # 输出创建的列表
7 print(food)
8
9 # 将 my_str 转换成列表，并输出
10 a = list(my_str)
11 print(a)
12 #####End#####
13
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.06	76.19	2025-04-22 13:38	查看代码

代码评价

暂无评价~



第2关 列表元素的访问

学员代码

代码修改行数 7

step2/step2.py



复制代码

```
1 #####Begin#####
2 # 创建一个列表，并赋值： "蒸羊羔儿", "蒸熊掌", "蒸鹿尾儿", "烧花鸭"
3 foods = ["蒸羊羔儿", "蒸熊掌", "蒸鹿尾儿", "烧花鸭"]
4
5 # 输出列表
6 print(foods)
7
8 # 输出列表的第2个元素（注意：索引从0开始）
9 print(foods[1])
10
11 # 输出列表的前两个元素
12 print(foods[:2])
13
14 # 输出列表的最后一个元素
15 print(foods[-1])
16
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.05	137.9	2025-04-22 13:40	查看代码

代码评价

暂无评价~



第3关 列表元素的添加

学员代码 [?](#)

代码修改行数 10

step3/step3.py



[复制代码](#)

```
1 my_list1 = ["Python", "C++", "Java"]
2 my_list2 = [1991, 1998, 1995]
3 #####Begin#####
4 # 在 my_list1 最后位置添加一个数组: ['C#', 'Go'] , 并输出
5 my_list1.append(["C#", "Go"])
6 print(my_list1)
7
8 # 在 my_list1 的二、三个元素之间插入一个元素: 'SQL' , 并输出
9 my_list1.insert(2, "SQL")
10 print(my_list1)
11
12 # 将 my_list2 中的元素逐个添加到 my_list1 当中, 并输出
13 for item in my_list2:
14     my_list1.append(item)
15 print(my_list1)
16 #####End#####
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.05	137.9	2025-04-22 13:42	查看代码

代码评价

暂无评价~



第4关 列表元素的排序

学员代码 [?](#)

代码修改行数 4

step4/step4.py



[复制代码](#)

```
1 my_list = [1,6,2,4,3,5,9,8,7]
2
3 #####Begin#####
4 # 对 my_list 正序排序，对原列表进行修改
5 my_list.sort()
6 print(my_list)
7
8 # 在不改变 my_list 的情况下得到倒序排列的新列表 new_list
9 new_list = sorted(my_list, reverse=True)
10 print(my_list)
11 print(new_list)
12 #####End#####
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.05	76.19	2025-04-22 13:43	查看代码

代码评价

暂无评价~



第5关 列表元素的删除

学员代码 [?](#)

代码修改行数 9

step5/step5.py



[复制代码](#)

```
1 my_list = ['蒸羊羔儿', '蒸熊掌', '烧雏鸡', '蒸鹿尾儿', '烧花鸭', '烧子鹅', '炉猪', '炉鸭', '酱鸡', '腊肉']
2
3 #####Begin#####
4 # 删除 my_list 中的第6个到第8个元素（包括第8个），并将删除元素后的 my_list 输出。
5 del my_list[5:8]
6 print(my_list)
7
8 # 将 my_list 的最后一个元素删除，并输出被删除的元素
9 removed = my_list.pop()
10 print(removed)
11
12 # 将 my_list 中的 烧雏鸡 删除，并将删除元素后的 my_list 输出。
13 my_list.remove('烧雏鸡')
14 print(my_list)
15
16 # 将 mv list 列表清空，并输出 mv list
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.05	76.19	2025-04-22 13:44	查看代码

代码评价

暂无评价~



第6关 列表推导式

学员代码

代码修改行数 2

step6/step6.py



复制代码

```
1 """
2 任务：给定一个列表，使用列表推导式来复制它，并筛选掉其中的负数，打印复制后的列表。
3 """
4
5 list1 = [2,3,-43,4,5,-5,4,-535,34,5,345,-3,45,3,6,345,-546,546,34,-56,34,5,86,7,0,12341,979,7,67,-856,454,8,-64,342,-63,56]
6
7 # 请在下面的Begin-End之间按照注释中给出的提示编写正确的代码
8 ##### Begin #####
9 # 使用列表推导式来复制给定的列表，并筛选掉其中的负数，打印复制后的列表
10 print([x for x in list1 if x >= 0])
11 ##### End #####
12
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.05	76.19	2025-04-22 13:46	查看代码

代码评价

暂无评价~