

G3-11 课后练习 —— 列表推导式

姓名： 日期：

练习说明

姓名： 日期：

预计完成时间：30-40分钟

难度递进：填空 仿写 创作

提交方式：下次上课前交给老师

第一部分：知识回顾（5分钟）

1.1 填空：补全口诀

把下面的口诀补充完整：

(1) 核心口诀：

" 画图 过滤，一句生成新 "

(2) 分解口诀：

"方括号里写 ， for字前面是 "

"for字后面 跑， if字后面是 "

(3) 身体锚点：

双手合拢 =

双手张开 =

单手做"停" =

双手画圈 =

1.2 连一连

把左边的列表推导式和右边的结果连起来：

[i * 2 for i in range(1, 5)] [1, 2, 3, 4]

[i for i in range(1, 5)] [2, 4, 6, 8]

[i for i in range(10, 21) if i % 10 == 0] [10, 20]

```
[i ** 2 for i in range(1, 4)]    [1, 4, 9]
```

1.3 判断对错

判断下面的说法对不对，对的打"√"，错的打"×"：

- () 列表推导式要用小括号括起来。
 - () if条件要写在for循环的后面。
 - () 列表推导式只能处理数字，不能处理字符串。
 - () 嵌套列表推导式可以用两个for循环。
 - () [x for x in range(5)] 生成的列表里有5个元素。
-

第二部分：基础练习（10分钟）

练习1：补全代码（填空）

补全下面的列表推导式，让它们能正确运行。

(1) 生成5的倍数

```
# 生成 5, 10, 15, 20, 25
倍数 = [i * 5 ____ i ____ range(1, 6)]
print(倍数)
```

(2) 温度转换

```
# 把摄氏温度列表转为华氏温度
# 公式：华氏 = 摄氏 * 9/5 + 32
摄氏 = [0, 10, 20, 30, 100]
华氏 = [c * 9/5 + 32 ____ c ____ 摄氏]
print(华氏)
# 期待输出：[32.0, 50.0, 68.0, 86.0, 212.0]
```

(3) 选择偶数位置的字符

```
# 选出字符串中偶数索引位置的字符
word = "Python"
结果 = [word[i] ____ i ____ range(len(word)) ____ i % 2 == 0]
print(结果)
# 期待输出：['P', 't', 'o']
```

(4) 单词首字母大写

```
动物 = ["cat", "dog", "bird", "fish"]
大写 = [a.capitalize() for a in 动物]
print(大写)
# 期待输出: ['Cat', 'Dog', 'Bird', 'Fish']
```

练习2：把for循环改成列表推导式

把下面的for循环代码改写成一行列表推导式。

(1) 平方数

```
# for循环写法
结果 = []
for n in range(1, 11):
    结果.append(n ** 2)
print(结果)
```

改成列表推导式（写在这里）：

(2) 选奇数

```
# for循环写法
奇数 = []
for x in range(1, 31):
    if x % 2 == 1:
        奇数.append(x)
print(奇数)
```

改成列表推导式（写在这里）：

(3) 过滤长度

```
# for循环写法
城市 = ["北京", "上海", "广州", "深圳", "杭州", "南京"]
长名 = []
for c in 城市:
    if len(c) >= 2:
        长名.append(c + "市")
print(长名)
```

改成列表推导式（写在这里）：

练习3：看代码说结果

不看电脑，写出下面代码的运行结果。

```
# (1)
r1 = [i + 100 for i in [1, 2, 3]]
```

```
print("r1 =", r1) # 你的答案: _____
```

```
# (2)
r2 = [n * 3 for n in range(0, 10, 2)]
print("r2 =", r2) # 你的答案: _____
```

```
# (3)
r3 = [s for s in ["hi", "hello", "hey", "howdy"] if len(s) > 3]
print("r3 =", r3) # 你的答案: _____
```

```
# (4)
r4 = [(x, x * x) for x in range(1, 5)]
print("r4 =", r4) # 你的答案: _____
```

```
# (5)
r5 = [a * b for a in [1, 2, 3] for b in [10, 20]]
print("r5 =", r5) # 你的答案: _____
```

第三部分：turtle绘图练习（10分钟）

练习4：画出笑脸

用列表推导式生成笑脸的坐标，然后用turtle画出来。

```
import turtle
import math
```

```
t = turtle.Turtle()
t.speed(3)
t.width(3)
```

```
# 第1步：用列表推导式生成圆形上的12个点（做笑脸的轮廓参考）
# 提示：角度从0到360，步长30，半径100
```

```
# 第2步：画脸（大圆）
t.penup()
t.goto(0, -100)
t.pendown()
t.circle(100)
```

```
# 第3步：画眼睛（两个小圆）
# 左眼
t.penup()
t.goto(-35, 20)
t.pendown()
t.dot(15)
# 右眼
```

```
t.penup()
t.goto(35, 20)
t.pendown()
t.dot(15)
```

```
# 第4步：用列表推导式生成嘴巴的弧线坐标
# 嘴巴是一个半圆，角度从 0 到 180
# 提示：x = 40 * cos(角度), y = 40 * sin(角度) - 10
嘴巴坐标 = [_____]
```

```
# 画嘴巴
t.penup()
t.goto(嘴巴坐标[0])
t.pendown()
t.color("red")
for 点 in 嘴巴坐标:
    t.goto(点)
```

```
t.hideturtle()
```

挑战加分：

- 用列表推导式生成腮红（两个粉色小圆的位置）
- 用列表推导式生成睫毛

练习5：画出彩色风车

用列表推导式生成风车的4个叶片，每个叶片用不同的颜色。

```
import turtle
import math
```

```
t = turtle.Turtle()
t.speed(5)
```

```
# 提示：风车有4个叶片，每个叶片是一个三角形
# 每个叶片相隔90度
# 叶片顶点：中心(0,0)，和两个偏移点
```

```
颜色列表 = ["red", "blue", "green", "orange"]
```

```
# 用列表推导式生成4个叶片的方向角度
角度列表 = [90 * i for i in range(4)]
```

```
# 画叶片
for idx, 角度 in enumerate(角度列表):
    t.penup()
    t.goto(0, 0)
    t.color(颜色列表[idx])
```

```
t.pendown()
t.begin_fill()
```

```
# 画出三角形叶片
t.setheading(角度)
t.forward(80)
t.left(120)
t.forward(80)
t.left(120)
t.forward(80)
```

```
t.end_fill()
```

```
# 画中心点
t.penup()
t.goto(0, -10)
t.color("yellow")
t.begin_fill()
t.circle(10)
t.end_fill()
```

```
t.hideturtle()
```

挑战加分：

- 用列表推导式生成多个半径的风车叶片（叠层风车）
 - 用列表推导式给叶片添加图案
-

练习6：画出雪人

用列表推导式生成雪人身体各部分的坐标和颜色，然后画出来。

```
import turtle
```

```
t = turtle.Turtle()
t.speed(4)
```

```
# 雪人由3个圆球组成：底部、中部、头部
# 用列表推导式生成每个圆球的半径和y坐标偏移
# 提示：底部半径80，中部半径60，头部半径40
```

```
# 雪人身体数据：[(半径, y坐标), ...]
身体 = [(80, -200), (60, -80), (40, 20)]
```

```
for 半径, y in 身体:
    t.penup()
    t.goto(0, y - 半径)
    t.pendown()
    t.color("white")
```

```
t.begin_fill()
t.circle(半径)
t.end_fill()
```

```
# 画眼睛
t.penup()
t.goto(-15, 35)
t.color("black")
t.dot(8)
t.penup()
t.goto(15, 35)
t.dot(8)
```

```
# 画鼻子（橙色的三角形）
t.penup()
t.goto(0, 20)
t.color("orange")
t.pendown()
t.begin_fill()
t.goto(10, 25)
t.goto(-5, 15)
t.goto(0, 20)
t.end_fill()
```

```
# 画扣子（用列表推导式生成扣子的坐标）
# TODO：在雪人身体中间画3个扣子
# 提示：y坐标从 -50 到 -90，间隔20
扣子y = [-50, -70, -90]
扣子坐标 = [(0, y) for y in 扣子y]
```

```
for x, y in 扣子坐标:
    t.penup()
    t.goto(x, y)
    t.color("black")
    t.dot(10)
```

```
t.hideturtle()
```

挑战加分：

- 用列表推导式生成雪花背景
- 用列表推导式生成围巾的颜色条纹
- 用列表推导式生成手臂的位置

第四部分：创作挑战（10分钟）

练习7：创意画图

请用列表推导式和turtle画一幅你喜欢的图案。

选题建议：

- 花朵：用列表推导式生成花瓣的坐标和颜色
- 烟花：用列表推导式生成烟花散开的点和颜色
- 小房子：用列表推导式生成房子各部分的坐标
- 自己的创意：想画什么就画什么

要求：

- 至少使用1个列表推导式
- 如果用if条件过滤，额外加分
- 如果用嵌套列表推导式，额外加分

我的作品名称： ____

代码：

```
import turtle
import math
import random

t = turtle.Turtle()
t.speed(0)

# 我的创意在这里！
```

完成度自评：

我用了__个列表推导式

我用了if条件过滤吗？ 是 / 否

我用了嵌套列表推导式吗？ 是 / 否

我给作品起了名字吗？ 是 / 否

我的代码能运行吗？ 是 / 否

我的画面好看吗？ ☆☆☆☆☆☆

练习8：解谜题

用列表推导式解决下面的数学问题。

题目：找出 100 到 999 之间的所有"水仙花数"。

- 水仙花数：一个三位数，每个位上的数字的立方和等于它本身。
- 例如：153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153

提示：

```
# 分解三位数
百位 = n // 100    # 整除100取百位
十位 = (n // 10) % 10 # 先整除10再取余数
个位 = n % 10      # 取余数得个位
```

挑战：能否用一行列表推导式找出所有的水仙花数？

```
# 试试看！
水仙花数 = [ ]
print(水仙花数)
```

参考答案：153, 370, 371, 407

第五部分：拓展思考（5分钟）

练习9：比较分析

比较下面两种写法，回答后面的问题。

```
# 写法A（for循环）
结果A = []
for i in range(10):
    if i % 2 == 0:
        结果A.append(i * 10)
```

```
# 写法B（列表推导式）
结果B = [i * 10 for i in range(10) if i % 2 == 0]
```

问题：

- 结果A和结果B的内容一样吗？ ____ _
- 哪种写法更短？ ____ _
- 哪种写法更容易理解？为什么？ ____ _
- 如果你要生成1000个数据，你会选哪种写法？为什么？ ____ _

练习10：家庭小老师

把今天学的内容教给爸爸妈妈，然后让他们做下面的小测试。

家长测试题：

(1) 请问：列表推导式的中文名称是什么？

家长回答：_____

(2) 请问：`[x**2 for x in range(1, 6)]` 的结果是什么？

家长回答：_____

(3) 请家长用一句话说说列表推导式的好处：

家长回答：_____

学生签名：_____

提交检查清单

在提交作业前，对照检查：

- ☐ 所有填空题都写完了
- ☐ 所有选择题都圈了答案
- ☐ for循环改推导式的题目都写了
- ☐ 看代码说结果的题目都写了答案
- ☐ turtle绘图代码都写了（至少3个练习）
- ☐ 创作挑战的作品名称写好了
- ☐ 解谜题尝试了
- ☐ 家长测试部分完成了
- ☐ 代码都在电脑上运行验证过了

额外挑战（选做）

这些题比较难，如果你全部做完了上面的练习还有时间，可以试试！

挑战1：质数过滤器

用一行列表推导式找出 1 到 50 之间的所有质数。

提示：质数是只能被1和自身整除的数。

质数 = `[n for n in range(2, 51) if _____]`

挑战2：蛇形路径

用列表推导式生成"蛇形"坐标，让turtle走出S形路径。

```
import turtle
```

```
t = turtle.Turtle()
t.speed(3)
```

```
# 提示：S形可以用正弦波来实现
# 横向x从-300到300，纵向y在-100和100之间摆动
蛇形 = [(x, 100 * math.sin(math.radians(x * 2))) for x in range(-300, 301, 5)]
```

```
for 点 in 蛇形:
    t.goto(点[0], 点[1])
```

挑战3：彩色像素画

用嵌套列表推导式生成一个10x10的彩色像素画。

```
import turtle
```

```
t = turtle.Turtle()
t.speed(0)
```

```
# 用嵌套列表推导式生成所有像素的位置和颜色
像素 = [ ]
颜色 = [ ]
```

```
# 每个像素10x10
for i in range(len(像素)):
    x, y = 像素[i]
    t.penup()
    t.goto(x, y)
    t.color(颜色[i])
    t.pendown()
    t.begin_fill()
    for _ in range(4):
        t.forward(10)
        t.right(90)
    t.end_fill()
```

```
t.hideturtle()
```

自我评价： ☐ 全都会了 ☐ 大部分会了 ☐ 还要再练练