

G3-04 课后练习 —— 元组tuple

姓名： ____ 日期： ____

核心记忆钩子：

- "元组 = 只读列表，不能增删改"
 - "解包 = 元组拆开赋给多个变量"
 - "单元素元组——数字后面加逗号！"
-

第一部分：温故知新（4 分钟 · 每题 2 分，共 10 分）

第1题 元组用什么符号创建？

- A. 「[]」
- B. 「()」
- C. 「{}」
- D. 「<>」

第2题 哪个选项是正确的单元素元组？

- A. 「(8)」
- B. 「(8,)」
- C. 「[8]」
- D. 「8」

第3题 「my_tuple[0] = 10」 会怎样？

- A. 成功修改
- B. 报 TypeError
- C. 报 IndexError
- D. 删除元素

第4题 解包 「a, b = (50, 100)」 后，a 和 b 的值是？

- A. a=100, b=50

- B. a=50, b=100
- C. a=(50,100), b=空
- D. 报错

第5题 元组和列表最大的区别是？

- A. 元组用圆括号，列表用方括号
 - B. 元组不可变，列表可变
 - C. 元组不能索引，列表可以
 - D. 元组不能切片，列表可以
-

第二部分：代码阅读（5 分钟 · 每题 3 分，共 12 分）

第6题 写出以下代码的输出：

```
data = (10, 20, 30, 40, 50)
print(data[0])    # 输出：_____
print(data[-1])   # 输出：_____
print(data[2:5])  # 输出：_____
print(len(data))  # 输出：_____
```

第7题 写出以下代码的输出：

```
point = (120, 80)
x, y = point
print("x =", x)
print("y =", y)
# 输出：
# _____
# _____
```

第8题 写出以下代码的输出：

```
a = 7
b = 12
a, b = b, a
print("a =", a)
print("b =", b)
# 输出：
# _____
# _____
```

第9题 以下代码哪里错了？请写出修正后的代码：

```
# 有错误的代码：
my_tuple = (1, 2, 3)
my_tuple.append(4)
print(my_tuple)
```

错误原因： _____

修正后的代码： _____

第三部分：代码补全（6 分钟 · 每题 4 分，共 16 分）

第10题：用元组画正方形

补全代码，用 4 个元组存正方形的 4 个顶点。

```
import turtle
t = turtle.Turtle()
t.speed(6)
t.color("green")
```

```
# 正方形的 4 个顶点
p1 = (__, __) # 左上 (-50, 50)
p2 = (__, __) # 右上 (50, 50)
p3 = (__, __) # 右下 (50, -50)
p4 = (__, __) # 左下 (-50, -50)
```

```
t.penup()
t.goto(p1)
t.pendown()
t.goto(p2)
t.goto(p3)
t.goto(p4)
t.goto(p1)
```

```
turtle.done()
```

第11题：用解包画菱形

补全代码，用解包读取每个顶点画菱形。

```
import turtle
t = turtle.Turtle()
t.speed(6)
t.color("blue")
```

```
# 菱形的 4 个顶点
diamond = [
```

```
(0, 100),    # 上
(80, 0),     # 右
(0, -100),   # 下
(-80, 0)    # 左
]
```

```
# 解包第一个顶点
____, ____ = diamond[____] # ① 取第 0 个点 (上)
t.penup()
t.goto(x0, y0)
t.pendown()
```

```
# 遍历其余顶点
for i in range(1, 4):
    ____, ____ = diamond[____] # ② 解包第 i 个点
    ____(____, ____)          # ③ goto 到该点
```

```
# 回到起点
x0, y0 = diamond[____]      # ④ 取第 0 个点
t.goto(x0, y0)
```

```
turtle.done()
```

第12题：元组 vs 列表对比

补全下面代码，展示元组和列表的区别：

```
# 列表——可以修改
my_list = [1, 2, 3]
print("修改前：", my_list)
my_list[____] = 100    # ① 把第一个元素改成 100
print("修改后：", my_list) # 预期：_____
```

```
# 元组——不能修改
my_tuple = (1, 2, 3)
print("元组：", my_tuple)
print("尝试修改元组...")
# my_tuple[0] = 100    # ② 取消注释运行会报_____
```

第13题：函数返回元组+解包

补全代码，让函数返回坐标元组，用解包接收。

```
import turtle
t = turtle.Turtle()
t.speed(6)
```

```
def get_mouse_click_coord():
    """假装获取鼠标点击坐标（实际上返回固定坐标）"""
    return (____, ____ ) # ① 返回坐标 (200, 150)
```

```
# 接收返回值并解包
x, y = _____() # ② 调用函数并解包
t.penup()
t.goto(0, 0)
t.pendown()
t.goto(____, ____ ) # ③ 用解包后的变量移动

turtle.done()
```

第四部分：小测（5 分钟 · 每题 1 分，共 6 分）

第14题 判断对错（打 或 ）：

题号	说法	对错
①	`(10, 20)` 是一个元组	()
②	`(10)` 是一个元组	()
③	元组可以添加元素：`tup.append(4)`	()
④	元组可以切片：`tup[1:3]`	()
⑤	元组解包：`a, b = (1, 2)` 让 a=1, b=2	()
⑥	元组比列表更适合存固定不变的坐标数据	()

第五部分：编程挑战（10 分钟 · 6 分）

第15题：用元组创作一个"小房子"图案

用至少 5 个坐标元组，画一个简单的房子（正方形+三角形屋顶）。

```
import turtle
t = turtle.Turtle()
t.speed(7)
t.color("brown")
```

```
# 小房子的坐标——用元组存每个顶点
# 提示：房子由矩形（墙壁）+ 三角形（屋顶）组成
# 至少需要 5 个点：左右下右上屋顶顶点左上左下
```

```
# 请在这里补全代码：
# _____
# _____
```

```
# _____
# _____
# _____
# _____
# _____

turtle.done()
```

评分标准：

要求	分值	检查项
至少 5 个元组坐标	2 分	每个点是 `(x, y)` 形式
画出房子形状	2 分	矩形 + 三角形屋顶
使用了元组特性	2 分	索引/解包/直接传元组至少用一种

第六部分：拓展思考（附加题 · 3 分）

第16题

为什么

Python

要同时有列表和元组两种"看起来差不多"的数据类型？在什么场景下应该优先用元组？（提示：从"数据安全"和"代码意图"两个角度思考）

你的思考： _____

自我检查清单

完成练习后，逐项打：

- ☐ 我能说出元组和列表的最大区别
- ☐ 我能正确创建单元素元组（带逗号）
- ☐ 我能用索引和切片访问元组
- ☐ 我能用解包把元组赋给多个变量
- ☐ 我知道元组不能修改（报 `TypeError`）
- ☐ 我能在 `turtle` 中用元组控制坐标

- [] 我能用「(x, y)」作为坐标传入「t.goto()」

自我评价: ☐ 全都会了 ☐ 大部分会了 ☐ 还要再练练