

الوحدة 04: القوائم في لغة بايثون

المحاضرة 07:

القوائم في لغة بايثون

في لغة بايثون، القوائم هي نوع من أنواع البيانات يستخدم لتخزين مجموعة من العناصر داخل متغير واحد. يمكن أن تكون هذه العناصر أعدادًا، نصوصًا، أو حتى قوائم أخرى.

ما يميز القوائم هو أنها مرنة أي يمكننا تعديلها أثناء تشغيل البرنامج، بإضافة عناصر جديدة، حذف بعضها، أو تغيير قيمها، وهذا يجعلها أداة أساسية للتعامل مع البيانات المنظمة أو المتغيرة بسهولة.

مثال (لتبيان أهمية القوائم في البرمجة):

لنفترض أن موظف لديه دخل شهري واحد ، نكتب : $revenu = 50000$ (قيمة واحدة تعني متغير واحد).

الآن نريد دراسة دخل نفس الموظف خلال 3 أشهر، فنكتب :

$$r1 = 50000$$

$$r2 = 52000$$

$$r3 = 48000$$

إذا طلب منا حساب المتوسط ، سنكتب: $moyenne = (r1 + r2 + r3) / 3$

لكن ماذا لو أردنا دراسة دخل 50 شهرًا؟ هل سننشئ 50 متغيرًا؟

هل سنكتب:

$$r1 = 50000$$

$$r2 = 51000$$

$$r3 = 49000$$

...

$$r50 = 52000$$

ماذا لو أضفنا زيوتاً جديداً؟

ماذا لو حذفنا زيوتاً؟

هنا نصبح ملزمين باستخدام بنية تسمح بتجميع القيم.

بدل 50 متغير، نكتب:

```
revenus = [50000, 51000, 49000 , ..., 52000]
```

الآن لدينا:

✓ متغير واحد

✓ يحتوي على عدة قيم

✓ يمكن التعامل معه كوحدة واحدة

ولحساب المجموع نكتب: `total = sum(revenus)`

عمليات على القوائم:

(1) إنشاء وتهيئة القوائم في لغة البايثون

لإنشاء قائمة ، نستخدم الأقواس المربعة [] ونفصل كل عنصر بفاصلة", ".

```
# إنشاء قائمة فارغة
empty_list = []

# إنشاء قائمة أعداد صحيحة
numbers = [1, 2, 3, 4, 5]

# إنشاء قائمة نصوص
names = ["Ali", "Ahmed", "Sara"]
```

لطباعة محتوى قائمة نستخدم الدالة `print()` كما يلي:

```
myList = ["Python", "Django", "Laravel"]

# طباعة محتوى القائمة

print(myList)

.....:الناتج

      ]'Laravel', 'Django', 'Python'["
```

(2) الوصول إلى عناصر القائمة

للوصول إلى عنصر من القائمة، نكتب اسم القائمة متبوعاً برقم (`index`) بين الأقواس المربعة:

```
# طباعة العنصر الثاني من القائمة

my_list = ["Ahmed", "Omar", "Ali"]

print(my_list[1]) # يعرض: 'Omar'
```

ملاحظة: رقم الـ `index` الأول في القائمة يبدأ من الصفر

(3) تعديل قيمة عنصر في القائمة

لتغيير قيمة عنصر محدد، نستخدم رقم الـ `index`:

```
# تغيير العنصر الثالث في القائمة

my_list = ["Ahmed", "Omar", "Ali"]

my_list[2] = "Sami"

print(my_list) # يعرض: ['Ahmed', 'Omar', 'Sami']
```

(4) تحديد طول القائمة في بايثون

لحساب عدد عناصر القائمة، نستخدم الدالة `len()`

```
# عرض عدد عناصر القائمة

my_list = ["Ahmed", "Omar", "Ali"]

print("طول القائمة هو: 3 #", len(my_list))
```

(5) المرور على عناصر القائمة

الطريقة الأولى: المرور مباشرة على العناصر

```
# طباعة جميع عناصر القائمة واحدًا تلو الآخر

my_list = ["Ahmed", "Omar", "Ali"]

for element in my_list:

    print(element) # يعرض كل عنصر على حدة
```

الطريقة الثانية: المرور باستخدام الـ `index`

```
my_list = ["Ahmed", "Omar", "Ali"]

n = len(my_list) # الحصول على طول القائمة

for i in range(0, n):

    print(my_list[i]) # يعرض كل عنصر على حدة
```

(6) إضافة أو حذف عناصر من القائمة

أ. إضافة عنصر في نهاية القائمة: باستخدام الدالة `append()`

```
# append() إضافة عنصر في نهاية القائمة باستخدام

my_list = ["Ahmed", "Omar", "Ali"]

my_list.append("Sami")

print(my_list) # يعرض: ['Ahmed', 'Omar', 'Ali', 'Sami']
```

ب. إضافة عنصر في موقع محدد: باستخدام الدالة `insert ()`

```
# إدراج عنصر في الموقع الثاني
my_list = ["Ahmed", "Omar", "Ali"]
my_list.insert(1, "Sami")
print(my_list) # يعرض: ['Ahmed', 'Sami', 'Omar', 'Ali']
```

ج. حذف عنصر من القائمة:

باستخدام الدالة `remove()` لحذف عنصر محدد

```
# حذف عنصر محدد باستخدام remove()
my_list = ["Ahmed", "Omar", "Ali"]
my_list.remove("Ahmed")
print(my_list) # يعرض: ['Omar', 'Ali']
```

باستخدام الدالة `pop()` لحذف عنصر حسب الـ `index`

```
my_list = ["Ahmed", "Omar", "Ali"]
my_list.pop(2)
print(my_list) # يعرض: ['Ahmed', 'Omar']
```

ملاحظة: يمكن أيضاً حذف القائمة بالكامل باستخدام الدالة `del`

```
my_list = ["Ahmed", "Sami", "Ali"]  
  
del my_list  
  
print(my_list) # هذا سيسبب خطأ لأن القائمة لم تعد موجودة
```

استخدام الدالة clear() لتفريغ القائمة

```
my_list = ["Ahmed", "Sami", "Ali"]  
  
my_list.clear()  
  
print(my_list) # يعرض: [] (قائمة فارغة)
```

تمرين:

لدينا مداخيل 5 زبائن في شهر معين كالتالي :

الدخل	الزبون
50000	Ahmed
62000	Fatima
58000	Omar
47000	Sara
70000	Youssef

قم بما يلي باستخدام القوائم في بايثون:

1. أنشئ قائمة تحتوي على أسماء الزبائن.
2. أنشئ قائمة تحتوي على مداخيل الزبائن.
3. اطبع دخل الزبون الثالث.
4. احسب متوسط الدخل لجميع الزبائن.
5. أضف زبوناً جديداً "Ali" بدخل 55000.
6. احذف دخل الزبون "Sara".

7. اطبع جميع الزبائن مع دخلهم بعد التعديلات.

الحل:

```
# 1. قائمة أسماء الزبائن
customers = ["Ahmed", "Fatima", "Omar", "Sara", "Youssef"]

# 2. قائمة مداخيل الزبائن
incomes = [50000, 62000, 58000, 47000, 70000]

# 3. طباعة دخل الزبون الثالث (Omar)
print("دخل الزبون الثالث:", incomes[2]) # 58000

# 4. حساب متوسط الدخل
average_income = sum(incomes) / len(incomes)
print("متوسط الدخل:", average_income) # 57400.0

# 5. بإضافة زبون جديد "Ali" 55000
customers.append("Ali")
incomes.append(55000)

# 6. حذف دخل الزبون "Sara"
index_sara = customers.index("Sara") # إيجاد قيمة index الخاص بسارة
incomes.pop(index_sara) # pop() باستخدام سارة
# حذف دخل الزبون سارة باستخدام ()

# 7. طباعة جميع الزبائن مع دخلهم بعد التعديلات
for i in range(len(customers)):
    print(customers[i], ":", incomes[i])
```

مجموعة تمارين حول استعمال التعليم الشرطية و التعليم التكرارية لمعالجة القوائم

تمرين 01: اكتب برنامجا يقوم بما يلي:

- (1) إنشاء قائمة تحتوي على مداخيل 6 زبائن كالتالي:
[45000, 72000, 51000, 83000, 39000, 60000]
- (2) طباعة المداخيل التي تفوق 60000 دج فقط.
- (3) حساب عدد الزبائن الذين دخلهم أقل من 50000 دج.
- (4) طباعة هذا العدد في النهاية.

الحل

```
incomes = [45000, 72000, 51000, 83000, 39000, 60000]
# طباعة المداخيل التي تفوق 60000
for i in incomes:
    if i > 60000:
        print("دخل مرتفع:", i)
# حساب عدد الزبائن الذين دخلهم أقل من 50000
count = 0
for i in incomes:
    if i < 50000:
        count += 1
print("عدد الزبائن بدخل أقل من 50000:", count)
```

التمرين 2:

اكتب برنامجا يقوم بإنشاء قائمة أرباح لأربعة شركات كالتالي: [120000, 80000, 45000, 200000]

ثم يقوم بحساب الضريبة لكل شركة حسب الشرط:

✓ إذا كان الربح أقل من 100000 : الضريبة = 10%

✓ إذا كان الربح أكبر أو يساوي 100000 : الضريبة = 20%

ثم يقوم بطباعة الربح والضريبة المقابلة له لكل شركة.

الحل

```
profits = [120000, 80000, 45000, 200000]

for p in profits:
    if p < 100000:
        tax = p * 0.10
    else:
        tax = p * 0.20
    print("الربح:", p, "الضريبة:", tax)
```

التمرين 03

اكتب برنامجا يقوم بما يلي:

(1) إنشاء قائمة معدلات: [9, 12, 15, 7, 14, 10, 16, 8]

(2) طباعة "ناجح" إذا كان المعدل أكبر أو يساوي 10.

(3) طباعة "راسب" إذا كان المعدل أقل من 10.

(4) حساب عدد الناجحين.

(5) طباعة عدد الناجحين في النهاية.

```
grades = [9, 12, 15, 7, 14, 10, 16, 8]
success_count = 0
for g in grades:
    if g >= 10:
        print("ناجح")
        success_count += 1
    else:
        print("راسب")
print("عدد الناجحين:", success_count)
```