بالطبع، سأقوم بشرح النص باللغة العربية وأيضًا سأضيف الأكواد بلغة بايثون ذات الصلة.

### النسخة المعدلة من كود بايثون

```python

# القاموس هو مجموعة غير مرتبة، قابلة للتغيير، ومرتبة. لا تحتوي على عناصر مكررة.

# تعريف القاموس 'person' مع قيم البداية

person = {

"first\_name": "ahmed",

"last\_name": "samy",

"age": 31

}

# الوصول إلى القيم في القاموس

print(person["first\_name"]) # يطبع القيمة المرتبطة بـ "first\_name"

print(person.get("last\_name")) # يحصل على القيمة المرتبطة بـ "last\_name" باستخدام get()

# المُنشئ: إنشاء قاموس آخر باستخدام الدالة dict()

person2 = dict(f\_name="sara", l\_name="wileam")

# إضافة زوج مفتاح-قيمة جديد إلى القاموس 'person'

person["phone"] = "12345678"

# عرض مفاتيح القاموس

print(person.keys())

# عرض عناصر القاموس (أزواج المفتاح-القيمة)

print(person.items())

# إنشاء نسخة من قاموس 'person' وإضافة زوج مفتاح-قيمة جديد إلى 'person3'

person3 = person.copy()

person3["city"] = "cairo"

# إزالة المفتاح "city" من 'person3'

del person3["city"]

# إزالة المفتاح "phone" من 'person' باستخدام طريقة pop()

person.pop("phone")

# مسح جميع العناصر من قاموس 'person'

person.clear()

# الحصول على طول (عدد العناصر) في 'person3'

print(len(person3))

# قائمة من القواميس، حيث يمثل كل قاموس شخصًا

people = [

{'name': 'Martha', 'age': 30},

{'name': 'Kevin', 'age': 25}

]

# الوصول إلى اسم القاموس الثاني في القائمة

print(people[1]['name']) # يطبع 'Kevin'

```

---

### الشرح سطرًا بسطر

1. \*\*السطر 1\*\*: `# القاموس هو مجموعة...`

- \*\*الشرح\*\*: هذه التعليق يوفر سياقًا حول ما هو القاموس. التعليقات لا تؤثر على تنفيذ الكود.

2. \*\*السطر 4-8\*\*: `person = {...}`

- \*\*الشرح\*\*: تم إنشاء قاموس يسمى `person` مع مفاتيح "first\_name"، "last\_name"، و "age" والقيم الخاصة بها.

3. \*\*السطر 11\*\*: `print(person["first\_name"])`

- \*\*الشرح\*\*: يطبع قيمة `first\_name` من القاموس `person`، وهي ستكون `"ahmed"`.

4. \*\*السطر 12\*\*: `print(person.get("last\_name"))`

- \*\*الشرح\*\*: يستخدم `get()` لاسترداد `last\_name`. طريقة `get()` أكثر أمانًا من الوصول المباشر لأنها لن ترفع خطأ إذا لم يكن المفتاح موجودًا.

5. \*\*السطر 15\*\*: `person2 = dict(f\_name="sara", l\_name="wileam")`

- \*\*الشرح\*\*: يستخدم مُنشئ `dict()` لإنشاء قاموس `person2` بمفاتيح "f\_name" و "l\_name".

6. \*\*السطر 18\*\*: `person["phone"] = "12345678"`

- \*\*الشرح\*\*: يضيف إدخالًا جديدًا إلى القاموس `person` مع المفتاح `"phone"` والقيمة `"12345678"`.

7. \*\*السطر 21\*\*: `print(person.keys())`

- \*\*الشرح\*\*: يطبع قائمة بجميع المفاتيح في القاموس `person`.

8. \*\*السطر 24\*\*: `print(person.items())`

- \*\*الشرح\*\*: يطبع جميع أزواج المفتاح-القيمة في القاموس كقائمة من التوابل.

9. \*\*السطر 27\*\*: `person3 = person.copy()`

- \*\*الشرح\*\*: ينشئ نسخة من `person` تُسمى `person3`. التغييرات في `person3` لن تؤثر على القاموس الأصلي `person`.

10. \*\*السطر 28\*\*: `person3["city"] = "cairo"`

- \*\*الشرح\*\*: يضيف مفتاح `"city"` إلى `person3` مع القيمة `"cairo"`.

11. \*\*السطر 31\*\*: `del person3["city"]`

- \*\*الشرح\*\*: يحذف المفتاح `"city"` من `person3` دون التأثير على `person`.

12. \*\*السطر 34\*\*: `person.pop("phone")`

- \*\*الشرح\*\*: يزيل المفتاح `"phone"` من `person`. طريقة `pop()` تزيل وتعيد القيمة المرتبطة بالمفتاح المحدد.

13. \*\*السطر 37\*\*: `person.clear()`

- \*\*الشرح\*\*: يزيل جميع العناصر من `person`، مما يجعله قاموسًا فارغًا.

14. \*\*السطر 39\*\*: `print(len(person3))`

- \*\*الشرح\*\*: يطبع عدد العناصر في `person3`، وهو طول القاموس.

15. \*\*السطر 43-47\*\*: `people = [...]`

- \*\*الشرح\*\*: ينشئ قائمة من القواميس، كل منها يمثل شخصًا مع `"name"` و `"age"`.

16. \*\*السطر 49\*\*: `print(people[1]['name'])`

- \*\*الشرح\*\*: يصل إلى القاموس الثاني في `people` (الفهرس 1) ويطبع القيمة المرتبطة بـ `"name"`، والتي ستكون `"Kevin"`.

### ملاحظات إضافية

لتأكيد المحتويات بعد إضافة أي مفتاح جديد أو إجراء تعديلات، يمكنك استخدام عبارات `print()` بعد كل عملية تغيير، كما هو موضح في النص السابق.

إذا كان لديك أي أسئلة أو تحتاج لمزيد من التوضيحات، فلا تتردد في طرحها!