

أنواع البيانات في بايثون

القوائم (Lists)

القائمة هي تسلسل ديناميكي من القيم يمكن أن تحتوي على عناصر من أنواع بيانات مختلفة. أهم الميزات:

إنشاء القوائم:

```
L1 = [] # قائمة فارغة
L2 = [1, 2, 3] # قائمة بأرقام
L3 = [1, True, "ESU"] # قائمة بعناصر من أنواع مختلفة
```

الوصول إلى العناصر:

```
numbers = [1, 2, 3, 4, 5]
print(numbers[0]) # طباعة العنصر الأول
print(numbers[-1]) # طباعة العنصر الأخير
```

تقطيع القوائم:

```
L1 = [1, 2, 3, 4, 5, 6, 7, 8, 9]
print(L1[3:]) # من العنصر الرابع إلى النهاية
print(L1[:5]) # من البداية إلى العنصر الخامس
print(L1[2:5]) # من العنصر الثالث إلى الخامس
```

عمليات القوائم:

```
L1 = [1, 2, 3, 4]
L2 = L1 + [5, 6] # دمج القوائم
L2 = L1 * 2 # تكرار القائمة
```

دوال القوائم:

```

L1 = [1, 2, 3, 4, 5]
print(len(L1))      # طول القائمة
print(min(L1))      # أصغر عنصر
print(max(L1))      # أكبر عنصر
L1.append(6)         # إضافة عنصر للنهاية
L1.insert(2, 7)      # إدراج عنصر في مكان محدد
L1.pop(3)            # إزالة العنصر من مكان محدد
L1.remove(2)         # إزالة العنصر بالقيمة
L1.clear()           # إزالة كل العناصر
L1.reverse()         # عكس ترتيب العناصر
L1.sort()            # ترتيب العناصر تصاعدياً

```

المجموعات (Tuples)

المجموعة تشبه القائمة لكنها غير قابلة للتغيير. مفيدة عندما تريد تخزين بيانات ثابتة.

إنشاء المجموعات:

```

T1 = (1, 2, 3)
T2 = tuple([4, 5, 6])

```

الوصول إلى العناصر:

```

print(T1[0]) # طباعة العنصر الأول

```

دوال المجموعات:

```

print(len(T1)) # طول المجموعة
print(min(T1)) # أصغر عنصر
print(max(T1)) # أكبر عنصر

```

القواميس (Dictionaries)

القاموس ينظم البيانات بشكل أزواج من المفاتيح والقيم.

إنشاء القواميس:

```
D1 = {} # قاموس فارغ
D2 = {"a": 1, "b": 2} # قاموس بأزواج مفاتيح وقيم
```

الوصول إلى العناصر:

```
print(D2["a"]) # طباعة القيمة المرتبطة بالمفتاح "a"
```

دوال القواميس:

```
D1 = {1: 'A', 2: 'B', 3: 'C'}
print(D1.keys()) # طباعة كل المفاتيح
print(D1.values()) # طباعة كل القيم
D1.update({4: 'D'}) # إضافة زوج مفتاح وقيمة جديد
D1.pop(2) # إزالة الزوج بالمفتاح المحدد
D1.clear() # إزالة كل الأزواج
```

أمثلة إضافية:

البحث عن أكبر رقم في قائمة:

```
def find_max(numbers):
    return max(numbers)

numbers = [10, 20, 30, 40, 50]
print(find_max(numbers)) # 50
```

التحقق من وجود عناصر مشتركة بين قائمتين:

```
def has_common_member(list1, list2):
    for item in list1:
        if item in list2:
            return True
    return False

list1 = [1, 2, 3]
list2 = [3, 4, 5]
print(has_common_member(list1, list2)) # True
```

التأكد من الأعداد الأولية في قائمة:

```
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, n):
        if n % i == 0:
            return False
    return True

def all_primes(numbers):
    for num in numbers:
        if not is_prime(num):
            return False
    return True

numbers = [2, 3, 5, 7]
print(all_primes(numbers)) # True
```

إنشاء قاموس بالدورات الدراسية:

```
courses = {  
    "Math": "MTH101",  
    "Physics": "PHY101",  
    "Chemistry": "CHM101"  
}  
print(courses)
```

تمارين

1. Find the Largest Number

Ask the user to enter 8 numbers and store them in a list.
Print the largest number in the list.

2. Remove Duplicate Values

Ask the user to enter a list of numbers.
Create and print a new list that contains only unique values while preserving the original order.

Example:

Input: [3, 5, 3, 2, 5, 7]

Output: [3, 5, 2, 7]

3. Student Grades Dictionary

Create a dictionary that stores the names and grades of 5 students.
Ask the user for a student's name and print their grade.
If the student does not exist, print **"Student Not Found"**.

4. Count Even and Odd Numbers

Ask the user to enter 10 integers and store them in a list.

Print:

- Number of even values
 - Number of odd values
-

5. Find Common Elements

Ask the user to enter two lists of numbers.

Print all numbers that appear in both lists without duplicates.

6. Tuple Statistics

Create a tuple containing 10 numbers.

Print:

The largest number •

The smallest number •

The sum of all numbers •

7. Word Frequency

Ask the user to enter a sentence.

Create a dictionary where each word is a key and its value is the number of times that word appears.

Example:

Input: apple orange apple banana apple

Output:

apple : 3

orange : 1

banana : 1

8. Reverse a List

Ask the user to enter 6 numbers and store them in a list.
Print the list in reverse order without using the reverse() method.

9. Product Price Lookup

Create a dictionary containing at least 5 products and their prices.
Ask the user for a product name.
If it exists, print its price; otherwise print "**Product Not Found**".

10. Prime Numbers in a List

Ask the user to enter 10 integers and store them in a list.
Print only the numbers that are prime.