الخوارزميات والبرمجة
الدرس السادس /الدوال البرمجية

أُقيِّمُ تعلُّمي

المعرفة :أُوظِّف في هذا الدرس ما تعلَّمْتُه من معارف في الإجابة عن الأ سئلة الآتية:

السؤال الأوَّل :أُوضِّح المقصود بكلٍّ ممّا يأتي:

1. مُعامِلات الدالَّة.

هي مجموعة من البيانات التي يجب توافرها عند استخدام بعض الدوالِّ حيث تعمل الدوالُّ على تحليلها ومعالجتها فمثلاً

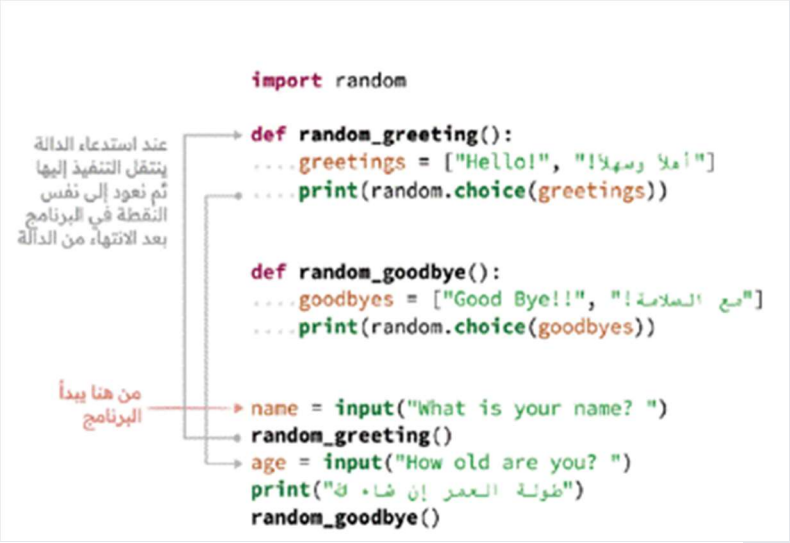
لا يُمكِن استخدام دالَّة الجذر التربيعي قبل استقبال رقم يُمكِّن من حساب جذره التربيعي

ولهذا يجب تحديد عدد مدخلات(أو مُعامِلات)الدالَّة parameters عند تعريفها وإعطاء كل مدخل(مُعامِل)اسمًا

2. استدعاء الدالَّة.

تنفيذ الدالة (function) بعد تعريفها. حيث تعريف دالة باستخدام الكلمة المفتاحية def، فتصبح جاهزة للتنفيذ، ولكنها لن تعمل حتى يتم استدعاؤها.

مثال :



3 .مدى المُتغيِّر.

هي المنطقة أو النطاق التي يكون فيها المتغير معرّفاً ومتاحاً للوصول والاستخدام ويعتمد المدى على مكان التعريف داخل الدالة أم خارجها وهو

من حيث	المدى المحلي	المدى العام
المفهوم	يتم إنشاء المتغيرات المحلية داخل الدوال أو الكتل البرمجية. هذه المتغيرات تكون مرئية فقط داخل الدالة التي تم تعريفها فيها ولا يمكن الوصول إليها خارجها. بمجرد انتهاء تنفيذ الدالة، يتم حذف المتغيرات المحلية.	المتغيرات التي يتم تعريفها خارج الدوال أو الكتل البرمجية تُعتبر متغيرات عامة (Global). يمكن الوصول إلى هذه المتغيرات من أي مكان في البرنامج، سواء داخل أو خارج الدوال

نوعان مدى محلي ومدى عام

4 .استيراد الوحدة.

مثال	def my_function() : x = 10 هنا متغير محلي print(x(	x = 5 هنا متغير عام def my_function: () print(x) يمكن الوصول إلى المتغير العام داخل الدالة

الوحدة هي ملفًّا يحوي دوالَّ برمجية خاصة بها(إضافةً إلى تعريفات أخرى لذا فإنَّ استخدام هذه الدوالِّ يتطلَّب أوَّلًا استيراد الوحدة باستخدام كلمة im port

السؤال الثاني :ما الفرق بين المدى العام والمدى المحلي للمُتغيِّرات

السؤال الثالث :ما الفرق بين الدالَّتين sort وreverse والدالَّتين sorted و reversed ؟

الفرق الرئيسي بينهم يكمن في كيفية التعامل مع القوائم وما إذا كانت تعدّل القائمة الأصلية أم تعيد نسخة جديدة معدّلة منها

الدالة	sort()	sorted()
آلية عملها	ترتيب القائمة نفسها ولا تُرجِع أيَّ نتيجة	تعمل على إرجاع إحدى القوائم المُرتَّبة حيث تقوم بترتيب العناصر في القائمة بترتيب تصاعدي أو تنازلي
تعديل القائمة الأصلية	تقوم بتعديل القائمة الأصلية بشكل مباشر	لا تعدل القائمة الأصلية
الإرجاع	لا تُرجع قيمة مرتبة فهي تعدل القائمة في مكانها	تُرجع نسخة مرتبة
مثال	Numbers=[1 ,9 ,2 ,4] numbers.sort() print(numbers) الناتج يكون [1, 2, 4, 9]	numbers = [4, 2, 9, 1] sorted_numbers = sorted(numbers) print(sorted_numbers) الناتج يكون[1, 2, 4, 9] القائمة الأصلية لم تتغير[4, 2, 9, 1]

الدالة	()reverse	()reversed
آلية عملها	عكس ترتيب عناصر في القائمة	عكس ترتيب العناصر لكنها لا تعدل القائمة الأصلية
تعديل القائمة الأصلية	تعدل القائمة الأصلية	لا تعدل القائمة الأصلية
الإرجاع	لا ترجع قيمة تعدل القائمة في مكانها	ترجع نسخة جديدة معكوسة من القائمة
مثال	numbers = [1, 2, 3, 4] numbers.reverse() print(numbers) الناتج [4, 3, 2, 1] تعدّل القائمة الأصلية وتعكسها	Numbers= [1, 2, 3, 4] reversed_numbers =reversed(numbers) print(list(reversed_numbers)) سيكون الناتج [4, 3, 2, 1] print(numbers(القائمة [1, 2, 3, 4] الأصلية تبقى كما هي

السؤال الأوَّل : أكتب دالَّة تستقبل (3) أرقام، وتُرجِع الرقم الوسيط، ثمَّ أكت ب برنامجًا بسيطًا يستدعي هذه الدالَّة.

```
import statistics
user_input = input("Enter numbers separated
by  spaces: ")
numbers = list(map(float, user_input.split()))
median_value = statistics.median(numbers)
print(f"The median is: {median_value}")
```

السؤال الثاني : أكتب دالَّة تستقبل رقمين أحدهما يُمثِّل الطول والآخر يُمثِّل ا لعرض، ثمَّ أطبع مستطيلًا مرسومًا من علامة '#' ثم أكتب برنامجًا بسيطًا يستدعي هذه ا لدالَّة.

```
def square(width, height):
  for i in range(height):
    print('#' * width)
```

السؤال الثالث : أكتب دالَّة تستقبل قائمتين، وتتأكَّد أنَّ كل عنصر في القائمة ا لأولى يساوي العنصر المُقابِل له في القائمة الثانية(يجب أنْ تُرجِع الدالَّة (True) أو (False))، ثمَّ أكتب برنامجًا بسيطًا يختبر هذه الدالَّة.

```
def compare_lists(list1, list2):
    if len(list1) != len(list2):
        return False
  for i in range(len(list1)):
```

```
        if list1[i] != list2[i]:
            return False
    return True

if compare_lists(list1, list2):
    print ("equal list")
else:
    print (" not equal list")
```

السؤال الرابع: أكتب برنامجًا يحتوي على دالَّة من تصميمي، ويطبع جدول ا لضرب لجميع الأرقام من 0 إلى 9 بَدْءًا بطباعة جدول الصفر كاملًا، ثمَّ طباعة جدول الرقم 1 كاملًا، وهكذا.

```
def print_multiplication_tables():
    for i in range(10):
        print(f" multiple table to {i} :")
        for j in range(11):
            print(f"{i} × {j} = {i * j}")
        print("." * 20)
print_multiplication_tables()
```

السؤال الخامس: أكتب برنامجًا يحتوي على دالَّة من تصميمي، ويطبع شكل المَعين كما في الأمثلة الآتية :

```
def print_upper_diamond_part(n):
لطباعة الجزء العلوي  من المعين حسب
    for i in range(n):
        print(' ' * (n - i - 1), end='')
```

```
        print('*' * (2 * i + 1))

def print_lower_diamond_part(n):
لطباعة الجزء السفلي من المعين
    for i in range(n - 2, -1, -1):
        print(' ' * (n - i - 1), end='')
        print('*' * (2 * i + 1))

def print_diamond(n):
    print_upper_diamond_part(n)
    print_lower_diamond_part(n)
```