المحاضرة السابعة : الدوال والبرمجة المعيارية

من خلال هذه المحاضرة نسعى الى فهم اعمق للبرمجيات من خلال التطرق الى الدالة واهم المفاهيم المرتبطة بها ولهذا نسعى لتحقيق الاهداف التالية :

- **ان يتعرف الطالب على ماهية الدالة وبنيتها**
- **ان يتعرف الطالب على كيفية كتابة الدالة في برنامج بايثون**
- **ان يطبق الطالب كتابة دالة وفق برنامج بايثون**

1. . **الدالة(Function)**مجموعة من العمليات التي يقوم بها جهاز كمبيوتر يوصف بأنها وحدة متماسكة. المتغير المحدد ليستقبل قيمة داخل دالة يعرف باسم (.**argument**) استخدام(**return**) يجعل من الممكن تمرير القيمة الناتجة عن العملية داخل الدالة إلى المتصل ك (**قيمة إرجاع return value**) بالإضافة إلى ذلك ، عند استخدام دالة، من الضروري إجراء عملية استدعاء للدالة

بنية الدالة:

return_type function_name(parameters) {

الكود المحلي //

return result;

{

أنواع الدوال

النوع	الوصف	مثال(Python)
Void Functions (no return)	تنفيذ مهمة فقط	def print_welcome(): print("Hello")
Value-Returning Functions	ترجع قيمة	def add(a, b): return a + b
Functions with Parameters	تأخذ مدخلات	def greet(name): print(f"Hi {name}")
Functions without Parameters	لا مدخلات	def calculate(): return 42

.البرمجة المعيارية(Modular Programming)

تقسيم البرنامج إلى **وحدات صغيرة مستقلة** (modules/functions) لتسهيل التطوير والصيانة

مبادئ:Top-Down Design

1. تقسيم المشكلة الكبيرة إلى أجزاء صغيرة
2. كل وحدة تقوم بمهمة واحدة
3. الوحدات الرئيسية تستدعي الفرعية

فوائد البرمجة المعيارية

إعادة استخدام الكود :**Reusability**

(صيانة سهلة (تعديل دالة واحدة :**Maintainability**

اختبار كل وحدة منفصلة :**Testability**

كود منظم وقابل للقراءة :**Readability**

: إضافة ميزات جديدة بسهولة**Scalability**

دورة حياة الدالة

text

:1. Declaration/Prototypeإعلان الدالة

:2. Definitionكتابة جسم الدالة

:3. Callاستدعاء الدالة

:4. Executionتنفيذ الكود

:5. Returnإرجاع النتيجة للمستدعي

أهمية الدوال:

1. **إعادة الاستخدام :(Reusability)** بدلا من كتابة نفس الأوامر عدة مرات، يمكن استدعاء الدالة متى لزم الأمر. وهذا يقلل حجم الشفرة ويجعلها أكثر إيجازاً.(Lynch, 2020)
2. **التجريد :(Abstraction)** تسمح للمبرمج باستخدام وظيفة معينة (مثل حساب المتوسط) دون الحاجة للقلق حول كيفية تنفيذها داخليا.

3. **تنظيم الكود :**تجعل الشفرة أسهل في القراءة والفهم والاختبار، حيث يتم تقسيم البرنامج إلى وحدات منطقية صغيرة.

ب. مكونات الدالة:

- **الاسم :(Name)** يُستخدم لاستدعاء الدالة وتنفيذها.
- **المدخلات/المُعاملات :(Parameters/Arguments)** هي القيم التي يمكن تمريرها إلى الدالة لتستخدمها في عمليتها.
- **القيمة المُعادة :(Return Value)** هي النتيجة التي تُرجعها الدالة إلى الجزء الذي استدعاها من البرنامج.

.2البرمجة المعيارية(Modular Programming)

البرمجة المعيارية هي منهجية تصميم ترتكز على فكرة تقسيم البرنامج الكبير إلى أجزاء أصغر وأكثر قابلية للإدارة تُسمى **الوحدات (Modules)** أو **المكتبات .(Libraries)** وتُعد الدوال أساس هذه المنهجية (Dovgan, 2021).

أ. دورالوحدات:(Modules)

الوحدة هي ملف يحتوي على مجموعة من الدوال والفئات (Classes) المرتبطة ببعضها البعض منطقياً (مثال: وحدة خاصة بالعمليات الرياضية.(

ب. مز ايا البرمجة المعيارية:

1. **سهولة الصيانة :**إذا حدث خطأ ما، يمكن تحديد مكانه وإصلاحه داخل الوحدة المعنية دون التأثير على الأجزاء الأخرى من البرنامج.
2. **تحسين إمكانية القراءة :**البرنامج يصبح منظماً في ملفات منفصلة ذات مهام واضحة، مما يسهل على المبرمجين الآخرين فهم الشفرة.
3. **التعاون :**تُمكن هذه المنهجية عدة مبرمجين من العمل على وحدات مختلفة من نفس المشروع في وقت واحد دون تداخل.

باختصار، تسمح الدوال بإعادة استخدام الشفرة داخل الوحدة الواحدة، بينما تتيح البرمجة المعيارية إعادة استخدام الوحدات نفسها عبر مشاريع مختلفة.(Lynch, 2020)