**الدوال في جافا :**

هي مجموعة من التعليمات البرمجية تنفذ فقط عند استدعاءها وتسمى  **function**

تستخدم الدوال لإعادة استخدام الكود حيث نقوم بكتابته مرة واحدة ونستخدمه عدة مرات

تحوي جافا مجموعة من الدوال الجاهزة التي يمكنك استعمالها مباشرة

**مثال :**

System.out.println( )

**إنشاء الدوال :**

عند انشاء أي دالة يجب معرفة بعض المصطلحات **:**

**Modifier** :تعني محدد الوصول

**returnType**: يحدد النوع الذي سترجعه الدالة يمكن ان يكون أي نوع من أنواع البيانات ((int-double-float-…

وفي حال ان الدالة لا ترجع قيمة يكون نوع الارجاع void

**methodname: يمثل** اسم الدالة

**parameter:** الوسطاء التي نمررها للدالة

**Method body** : تعني التعليمات التي تنفذها الدالة

Modifier returnType methodname (parameter ){

Method body

}

**مثال :**

public class Main {

static void myMethod() {

System.out.println("I just got executed!");

}

public static void main(String[] args) {

myMethod();

}

}

يمكن تمرير وسطاء إلى الدالة يمكن أضافة أكثر من وسيط فقط افصلهم بفاصلة

مثال :

public class Main {

static void myMethod(String fname) {

System.out.println(fname + " Refsnes");

}

**استدعاء الدالة :**

لا يتم تنفيذ التابع بمجرد التصريح عنه لتنفيذ التعليمات الموجودة ضمنه يجب علينا استدعاءه ضمن تابع اخر عن طريق ذكر اسمه مع ارسال قيم الوسطاء الخاصة به

نلاحظ أذا كان نوع الارجاع void نقوم بذكر اسم التابع مع تمرير الوسطاء بينما اذا كان نوع الارجاع

(int-float-double) يفضل اسناد ناتج الاستدعاء في متحول لاستخدامه لاحقا او يمكن طباعته مباشرة عن طريق تابع الطباعة .

يستخدم نوع الارجاع void عندما لا ترجع الدالة أي شيء إذا أردنا للدالة أن ترجع نوع بيانات معين يمكن استخدام أحد أنواع البيانات (int,char,float,..)

ويجب استخدام return بداخل الدالة

مثال :

public class Main {

static void myMethod(String fname) {

System.out.println(fname + " Refsnes");

}

public static void main(String[] args) {

myMethod("Liam");

}

}

// Liam Refsnes

يتم هنا استبدال الوسيط fname في التابع الذي تم استدعاءه بالقيمة liam ليتم بعدها تنفيذ التعليمات الموجودة ضمنه باستخدام هذه القيم .

**مثال :**

public class Main {

static **int** myMethod(int x) {

**return** 5 + x;

}

public static void main(String[] args) {

System.out.println(myMethod(3));

}

}

// Outputs 8 (5 + 3)

كما يمكننا تخزين النتيجة في متغير **.**

**مثال :**

public class Main {

static int myMethod(int x, int y) {

return x + y;

}

public static void main(String[] args) {

int z = myMethod(5, 3);

System.out.println(z);

}

}

// Outputs 8 (5 + 3)

**التحميل الزائد للدوال :**

يمكن لأكثر من دالة أن يكون لها نفس الاسم مع الاختلاف في نوع الوسطاء أو عددها أو ترتيبها

مثال:

بدلا من تعريف دالتين يقومان بنفس المهمة يمكن أن نقوم بالتحميل الزائد لهم

static int plusMethod(int x, int y) {

return x + y;

}

static double plusMethod(double x, double y) {

return x + y;

}

public static void main(String[] args) {

int myNum1 = plusMethod(8, 5);

double myNum2 = plusMethod(4.3, 6.26);

System.out.println("int: " + myNum1);

System.out.println("double: " + myNum2);

}