

TÍTULO

PROGRAMACION ORIENTADA A OBJETOS

Semana 02

**TIPOS DE MÉTODOS Y ÁMBITO DE LAS
VARIABLES**

AGENDA

- Tipos de métodos y ámbito de las variables.
- Manejo del IDE.



OBJETIVOS

- ◆ Entender como se define y se llama a los diferentes tipos de métodos.
- ◆ Aplicar la programación haciendo uso métodos con parámetros, sin parámetros, con retorno y sin retorno.
- ◆ Aplicar la programación haciendo uso del método onCreate().
- ◆ Entender el ámbito de las variable.

MÉTODOS

- ◆ Un método es un conjunto de una o más instrucciones del programa, que puede ser ejecutado por referencia al nombre del método.
- ◆ Para utilizar los métodos, es necesario:
 - ◆ Definir los métodos.
 - ◆ Llamar a los métodos.

ESTRUCTURA PARA DEFINIR UN MÉTODOS

- ◆ Los elementos de la declaración de un método incluyen el tipo de retorno, el nombre del método, la lista de parámetros y el cuerpo del método.
- ◆ A continuación se muestra la sintaxis para la definición de un método:

```
<Return Type> <Method Name> (Parameter List)  
{  
    Method body  
}
```

DEFINICIÓN DE MÉTODOS

Especificador de Acceso:

Determina el grado en que una variable o método se puede acceder desde otra clase.

```
<Access specifier> <Return Type> <Method Name> (Parameter  
    List)  
  
{  
    Method body  
}
```

Tipo de Retorno :

Un método puede devolver un valor de cualquier tipo de dato. Si el método no devuelve ningún valor, se utiliza la palabra reservada **void** .

```
<Access specifier> <Return Type> <Method Name>
    (Parameter List)

{

    Method body

}
```

DEFINICIÓN DE MÉTODOS

Nombre del Método :

Este es un identificador único y diferencia entre mayúsculas y minúsculas. El nombre del método no puede ser el mismo que el nombre de una variable declarada en la misma clase.

```
<Access specifier> <Return Type> <Method Name> (Parameter  
    List)  
  
{  
  
    Method body  
  
}
```


Lista de Parámetros :

Se utiliza para pasar y recibir los datos de un método. Está encerrada entre paréntesis. Los paréntesis se incluyen, aunque no existan parámetros

```
<Access specifier> <Return Type> <Method Name> (Parameter  
List)
```

```
{
```

```
Method body
```

```
}
```



DEFINICIÓN DE MÉTODOS

Cuerpo del Método :

Aquí se ingresa una o mas instrucciones necesarias para realizar una determinada operación o actividad requerida.

```
<Access specifier> <Return Type> <Method Name> (Parameter List)
{
    .
    Method body
}
```

LLAMAR AL MÉTODO

- ◆ Después de definir el método, se puede ejecutar llamándolo.
- ◆ Usted puede llamar a un método con el nombre del método.
- ◆ El nombre del método es seguido de paréntesis, incluso si la llamada al método no tiene parámetros, como se muestra en el siguiente

ejemplos:

- ✓ Método sin retorno y sin parámetros:

```
leerNotas ();
```

- ✓ Método sin retorno y con parámetros:

```
nombreDelMetodo (n1, n2) ;
```

- ✓ Método con retorno y sin parámetros:

```
double prom=retornarN1 ();
```

- ✓ Método con retorno y con parámetros:

```
double prom=calcularPromedio (n1, n2, n3) ;
```

LLAMAR AL MÉTODO

- ◆ El siguiente es un ejemplo de cómo llamar a un método:

```
public class Calculadora extends Activity {  
  
    protected void onCreate(Bundle savedInstanceState) {  
        int sum=calcularSuma(3,5);  
    }  
  
    public static int calcularSuma(int num1, int num2){  
        int s;  
        s = num1 + num2;  
        return s;  
    }  
}
```

- El *ámbito de una variable* define su *alcance de uso*, o lo que es lo mismo, en que secciones de código una variable estará disponible. Fuera de este ámbito, una variable no podrá ser accedida (no existe).
 - Tipos de ámbito de las variables, en Android tenemos tres tipos de ámbito que pueden aplicar a una variable:
 - Local
 - Global
 - Estático
-
1. **Variables Local** : Las variables de ámbito local, o de bloque, son aquellas que sólo pueden ser accedidas desde el bloque de código en el que han sido declaradas
 2. **Variables Global** : Las variables de ámbito global o de instancia, son aquellas que pertenecen a cada instancia concreta de la clase donde han sido declaradas y dependiendo del modificador de visibilidad usado podrían ser sólo accedidas desde la propia instancia a la que pertenecen

3. **Variables Estáticas** : Conocidas como variables de clase, son aquellas que pertenecen a la propia clase donde han sido declaradas, y dependiendo del modificador de visibilidad usado podrían ser sólo accedidas desde la propia clase en la que han sido declaradas.

```
public class ClaseApp {  
    static int variableEstatica;  
  
    int variableGlobal; (o de instancia)  
  
    void metodoApp(int parametro) {  
        int variableLocal;  
  
        // El parámetro también es local dentro del método  
    }  
}
```

- ◆ Un método es un conjunto de una o más sentencias de programa que puede ser ejecutado por referencia al nombre del método.
- ◆ La definición de un método supone la declaración de los elementos de su estructura.
- ◆ Los parámetros permiten que la información que se tiene dentro de un método se pueda pasar a otro método. Cuando se define un método, que incluye una lista de parámetros entre paréntesis.
- ◆ La sentencia **return** se utiliza para devolver un solo dato de cualquier tipo.
- ◆ Las variables se declaran y se les asigna un valor.
- ◆ Según su alcance, las variables pueden ser globales, locales o de clase.