

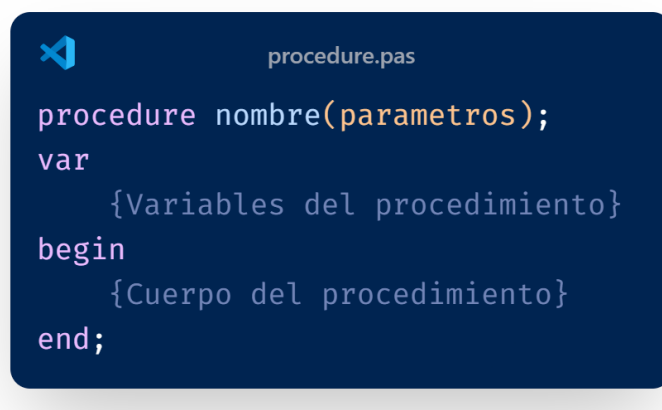
## # Modularización

Modularizar significa dividir un problema en partes funcionalmente independientes que encapsulen operaciones y datos.

Los módulos son tareas específicas bien definidas que se comunican entre sí y cooperan para conseguir un objetivo común.

**Procedimientos:** son un conjunto de instrucciones que realizan una tarea específica y retornan más de 1 valor.

Sintaxis:

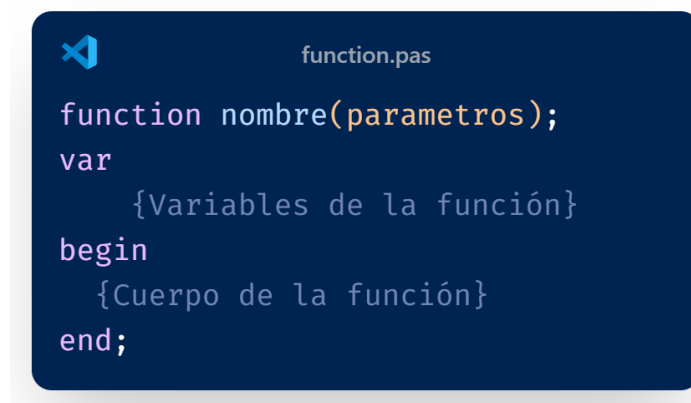
A screenshot of a code editor showing the syntax for a Pascal procedure. The file name is 'procedure.pas'. The code is as follows:

```
procedure nombre(parametros);  
var  
    {Variables del procedimiento}  
begin  
    {Cuerpo del procedimiento}  
end;
```

Estos NO pueden ser invocados en estructuras de control

**Funciones:** son un conjunto de instrucciones que realizan una tarea y retornan solo un valor SIMPLE (integer, boolean, real..)

Sintaxis:

A screenshot of a code editor showing the syntax for a Pascal function. The file name is 'function.pas'. The code is as follows:

```
function nombre(parametros);  
var  
    {Variables de la función}  
begin  
    {Cuerpo de la función}  
end;
```

Pueden ser invocados mediante una variable (mismo tipo de dato), estructuras de control, programa principal y procedimientos

## # Parámetros

Llamado Data Hiding (ocultamiento de datos), significando que los datos exclusivos de un módulo no deben ser visibles o utilizables por los demás módulos. Existen 2 tipos de parámetros; por valor y por referencia.

**Por valor:** el módulo recibe (sobre una variable local) un valor proveniente. Se pueden realizar operaciones y/o cálculos, pero no producirá ningún cambio fuera del módulo.

Sintaxis:

```
paramPorValor.pas

procedure nombre(nombre: dato)
var
    {Variables del procedimiento}
begin
    {Cuerpo principal del procedimiento}
end;
```

**Por referencia:** el módulo recibe (sobre una variable local) un valor proveniente. Se pueden realizar operaciones y/o cálculos y su valor original dentro del módulo hará que se modifique en los demás módulos que usen esa variable.

Sintaxis:

```
paramPorRef.pas

procedure nombre(var nombre: dato);
var
    {Variables del procedimiento}
begin
    {Cuerpo del procedimiento}
end;
```

¡¡Estos solo se usan en procedimientos!!