

FUNCION PRINCIPAL

El compilador requiere obligatoriamente definir una función principal con el fin de hacer uso de las demás funciones que posee, esta función se define de la siguiente manera:

```
NombreFuncion comienzo{  
Instrucciones  
Instrucciones  
...  
..  
..  
fin}
```

NombreFuncion

El nombre de la función admite cualquier palabra que inicie sea con letra mayúscula o minúscula, no obstante, no se permite declarar una función con caracteres numéricos o especiales, algunos ejemplos no validos son los siguientes:

4funcion

%funcion

&2

Ejemplos validos

Funcion

FUNCION

Funcion

FuNclon

Instrucciones

Las siguientes acciones son permitidas dentro de la función principal del compilador:

Definición de variables

Es posible declarar tres tipos de variables:

TIPO	Palabra reservada	Sintaxis
Entero	entero	entero variable; o entero var1,var2;
Decimal	decimal	decimal variable; o decimal var1,var2;
Cadena	cadena	cadena variable; o cadena var1,var2;

Nota:

El nombre de los tipos de variables no debe iniciar con caracteres especiales o numéricos

Visualizar mensajes en la consola

Si desea mostrar un mensaje en pantalla, utilice la siguiente sintaxis

mostrar (#Mostrar mensaje en pantalla#);

En caso de que el mensaje se encuentre en una variable tipo cadena, puede visualizar su contenido así:

mostrar (variable);

Ingreso de datos por teclado

Si desea pedir datos a través de teclado utilice la siguiente sintaxis:

Leer(variable);

Igualmente puede incluir un mensaje junto con el dato que desea pedir usando el símbolo +

Leer(#Ingrese el dato#+variable);

Estructuras de control

Sobre la función principal es posible utilizar estructuras condicionales, ciclos for y while, la sintaxis de cada código es el siguiente.

Condicional Si

La sintaxis para esta estructura condicional es la siguiente:

si(condicion){

Instrucciones

Instrucciones

...

..

.

finsi}

Estructura Mientras

Para la formación de la estructura 'mientras' se utiliza la siguiente sintaxis:

```
mientras(condicion) {  
  Instrucciones  
  Instrucciones  
  ...  
  ..  
  .  
finmientras}
```

Condición

Tanto para la estructura 'si' como para la estructura 'mientras', se deben realizar comparaciones entre variables o números con el fin de que se cumpla o no la sentencia y pueda ejecutar las instrucciones que dispone, algunas condiciones validas son las siguientes:

1. $Var > Var2$
2. $Var < Var2$
3. $Var \geq Var2$
4. $Var \leq Var2$
5. $Var > 3$

Estructura Para

Para la formación de la estructura para se utiliza la siguiente sintaxis:

```
para var con_paso valor hasta valor2{  
  Instrucciones  
  Instrucciones
```

...

..

.

finfor}