Lenguaje Me&Myself

Manual de usuario

1. Estructura general de un program en MeMyself

Sintaxis

program [nombre del programa];

main() { }

Ejemplo

program Alejandro;

main {}

Descripción.Un programa en MeMyself tiene dos componentes obligatorios: la declaración del nombre del programa (método global), y el método principal (método “main”)

1. Declaración de variables

Sintaxis

var [tipo] : [lista de variables];

Ejemplo

var int : numero1, numero2, arr[10];

Uso

program MeMyself;

var int : numero1, numero2, arr[10];

var float : valor, yepei ;

int module fact (int j)

var int : pop;

{

return 1+2;

}

main (){

}

Descripción. La declaración de variables debe seguir la sintaxis descrita. Además solo es posible declarar variables después de la declaración del nombre del programa y después de la declaración de la firma de una variable. Por último, solo se permite declarar tres tipos de variables: float, char e int

1. Declaración de funciones

Sintaxis

[tipo de retorno] module [nombre del módulo] (<lista parámetros>)

<declaracion de variables>

{ <return [expresión] ; >}

Ejemplo

void module snip(int w, int z)

var int : nopusok[5],sobresitos;

{

}

Uso

program MeMyself;

var int : i, j, p;

var float : valor, yepei ;

void module snip(int w, int z)

var int : nopusok[5],sobresitos;

{

}

main (){

}

Descripción. La declaración de funciones debe seguir la sintaxis descrita. Además solo es posible declarar funciones después de la declaración del nombre del programa y antes del método principal main. Además, si el tipo de retorno de la función es diferente de void, entonces se debe incluir un estatuto “return”. Por último, los tipos de retorno validos para una función, son: void, int, float y char

1. Expresiones, operadores y operandos

Sintaxis

[operando] [operador] [operando]

([operando] [operador] [operando])

([operando])

Ejemplo

(suma(3+4,4)+4) \*(3) - 2

Uso

program MeMyself;

var int : i, j, p;

var float : valor, yepei ;

int module suma(int w, int z)

{

return w + z

}

main (){

j = (suma(3+4,4)+4) \*(3) - 2

}

Descripción. El uso de expresiones debe seguir la sintaxis descrita. Además, no es posible utilizar expresiones como estatutos aislados, sino que necesariamente deben ir dentro como parte de los estatutos que lo requieran (por ejemplo, una asignación

1. Estatutos de asignación

Sintaxis

[variable] = [expresión]

Ejemplo

j = (2+4) \* 8

arr[4] = 12

Uso

program MeMyself;

var int : i, j, p, arr[10];

var float : valor, yepei ;

main (){

j = (2+4) \* 8

arr[4] = 12

}

Descripción. El uso de estatutos de asignación debe seguir la sintaxis descrita. Además, no es posible asignar a una variable un valor que no sea del mismo tipo que la variable en sí.

1. Estatutos condicionales

Sintaxis

if ( [expresión]) then {<estatutos>}

if ( [expresión]) then {<estatutos>} else {<estatutos>}

Ejemplo

if ( 3 > 4) then {print(“well done”);} else {print(“bad done”);}

Uso

program MeMyself;

var int : i, j, p, arr[10];

var float : valor, yepei ;

main (){

if ( 3 > 4) then {

print(“well done”);

} else {

print(“bad done”);

}

}

Descripción. El uso de estatutos de condición debe seguir la sintaxis descrita. Además, la expresión condicional de un estatuto de condición debe siempre resultar en un valor lógico (true o false)

1. Ciclos while

Sintaxis

while ([expresión]) do { <estatutos>}

Ejemplo

while (j < 2) do { print(“hola hellow”); j = j + 1}

Uso

program MeMyself;

var int : i, j, p, arr[10];

var float : valor, yepei ;

main (){

j = 0

while (j < 2) do

{

print(“hola hellow”);

j = j + 1

}

}

Descripción. El uso de estatutos de repetición while debe seguir la sintaxis descrita. Además, la expresión condicional de un estatuto de repetición while debe siempre resultar en un valor lógico (true o false). Por ultimo, es necesario utilizar un mecanismo (ejemplo: contador incremental) para salir del ciclo.

1. Ciclos For

Sintaxis

for [asignacion] to [expresion] do { <estatutos> }

Ejemplo

for j = 0 to j >3 { print(“hola hellow”); j = j + 1}

Uso

program MeMyself;

var int : i, j, p, arr[10];

var float : valor, yepei ;

main (){

for j = 0 to j >3

{

print(“hola hellow”);

j = j + 1

}

}

Descripción. El uso de estatutos de repetición for debe seguir la sintaxis descrita. Además, la expresión condicional de un estatuto de repetición for debe siempre resultar en un valor lógico (true o false). Por ultimo, es necesario utilizar un mecanismo (ejemplo: contador incremental) para salir del ciclo.

1. Llamadas a métodos void

Sintaxis

/[id](<lista parámetros>);

Ejemplo

/aumentaVariable(3);

Uso

program MeMyself;

var int : i, j, p, arr[10];

var float : valor, yepei ;

void module aumentaVariable(int z)

{

j = j + z

}

main (){

/aumentaVariable(3);

print(j);

}

Descripción. El uso de estatutos de llamadas a métodos void debe seguir la sintaxis descrita. Además, no es posible realizar esta clase de estatutos dentro de expresiones, pues no tienen un valor de retorno que pueda ser utilizado como operando de ninguna operaciónn

1. Lista de funciones predefinidas o built in

Sintaxis

print([Lista de variables o constantes string]); #constante o variable que mostrar en consola

write([variable]) #variable a la cual asignarle un valor

penup();

pendown();

clear();

arc([expresion]. [expresion]); #radio, angulo

line([expresion], [expresion], [expresion], [expresion]); #x1,y1,x2,y2

circle([expresion]) #radio

point([expresion], [expresion]) #x,y

color([constante string]); #color elegido

size([expresion]); #tamaño

Descripción. El uso de las funciones predefinidas o built in debe seguir la sintaxis descrita. La funcionalidad de cada de esta se describe a continuación:

1. print: escribe en consola la lista de variables y constantes especificadas
2. write: asigna a la variable especificado un valor de consola
3. penup: levanta el lápiz de python turtle
4. pendown: baja el lápiz de python turtle
5. clear: borra los gráficos dibujados en el GUI de python turtle
6. arc: dibuja un arco en el GUI de python turtle
7. line: dibuja una línea en el GUI de python turtle
8. circle: dibuja un circulo en el GUI de python turtle
9. point: dibuja un punto en el GUI de python turtle
10. color: cambia el color de los trazos del lápiz de python turtle
11. size: cambia de tamaño los trazos del lápiz de python turtle