

PROGRAMING LENGUAJE

Este lenguaje de programación fue hecho a base de las ideas de aprender a programar. Únicamente se crearon las instrucciones o comandos básicos de cada lenguaje de programación. Acá se define cada instrucción a base de ejemplos. La filosofía detrás de esto no es mas que simplemente enseñar a programar a las personas que no han entrado al mundo de la programación aún.

EL EJECUTABLE DEL ANALISADOR SINTACTICO APARTE DE ANALISAR, MUESTRA EN CONSOLA QUE INSTRUCCIÓN ES LA QUE LEYÓ.

CONCEPTOS

➤ WHATIS:

- QUE ES WHATIS
Whatis es una palabra reservada propia de este código que me muestra que tipo de dato es lo que yo estoy poniendo;
- COMO SE ESCRIBE
Se escribe el comando whatis seguido de algún dato.
- EJEMPLOS
 - whatis 4;
 - whatis 3.4;
 - whatis a;
 - whatis hola;
 - whatis 3 + 3;

➤ VARIABLE:

- QUE ES UNA VARIABLE
Una variable es un espacio de memoria donde podemos almacenar valores que pueden variar conforme el código avance.
- TIPOS DE VARIABLES
Las variables poseen un tipo, los cuales pueden ser:
 - INTEGER: Este tipo de variable almacena datos de tipo entero.
 - FLOAT: Este tipo de variable almacena datos de tipo flotante.
 - CHARACTER: Este tipo de variable almacena datos de tipo carácter.
 - CHAIN: Este tipo de variable almacena datos de tipo cadena de caracteres.
- DECLARACION DE VARIABLES
 - EJEMPLO DE DECLARACION TIPO INTEGER:
 - Integer variable1; (declaración de variable entero sin valor)
 - Integer variable2 = 0; (se almacena un valor entero)
 - integer variable3 = x1; (donde x1 almacena valor entero)
 - integer variable4 = 9 + 4; (expresión algebraica con resultado entero)
 - integer variable5 = x2 + 4; (donde x2 almacena valor entero)
 - EJEMPLO DE DECLARACION TIPO FLOAT:
 - float variable1; (declaración de variable flotante sin valor)
 - float variable2 = 3.6; (se almacena un valor flotante)
 - float variable3 = x1; (donde x1 almacena valor flotante)
 - float variable4 = 9.1 + 4.2; (expresión algebraica de resultado flotante)
 - float variable5 = x2 + 4.3; (donde x2 almacena valor flotante)

- float variable6 = x3 + 4; (donde x3 almacena valor flotante)
- EJEMPLO DE DECLARACION TIPO CHARACTER:
 - character variable1; (declaración de variable caracter sin valor)
 - character variable2 = 'a'; (se almacena un valor caracter)
 - character variable3 = y1; (donde y1 posee valor carácter)
- EJEMPLO DE DECLARACION TIPO CHAIN:
 - chain variable1; (Declaración de cadena sin valor)
 - chain variable2 = 'hola como estas'; (se almacena un valor cadena)
 - chain variable3 = 'palabra'; (se almacena un valor cadena)
- ASIGNACION DE VALORES
 - EJEMPLO DE ASIGNACION TIPO INTEGER:
 - Variable1 = 0; (donde variable1 fue declarada como integer)
 - Variable2 = x1; (donde variable2 es variable integer y x1 almacena valor entero)
 - Variable3 = 9 + 4; (donde variable3 fue declarada como integer)
 - variable4 = x2 + 4; (donde variable4 es variable integer y x2 almacena valor entero)
 - EJEMPLO DE ASIGNACION TIPO FLOAT:
 - Variable1 = 0.5; (donde variable1 fue declarada como float)
 - Variable2 = x1; (donde variable2 es variable float y x1 almacena valor flotante)
 - Variable3 = 9.4 + 4.2; (donde variable3 fue declarada como integer)
 - variable4 = x2 + 4; (donde variable4 es variable float y x2 almacena valor flotante)
 - EJEMPLO DE ASIGNACION TIPO CHARACTER:
 - variable2 = 'a'; (se almacena un valor caracter)
 - variable3 = y1; (donde y1 posee valor caracter)
 - EJEMPLO DE ASIGNACION TIPO CHAIN:
 - variable2 = 'hola como estas'; (se almacena un valor cadena)
 - variable3 = 'palabra'; (se almacena un valor cadena)

➤ **IMPRIMIR:**

- QUE ES IMPRIMIR
Imprimir en pantalla se puede lograr haciendo uso de la palabra reservada print.
- ESCRITURA DE PRINT
Print es un comando que nos ayuda a imprimir valores en la pantalla. Debe escribirse seguido de paréntesis de apertura y entre comillas simples el texto deseado a imprimir, luego paréntesis de cerradura. Si se desea imprimir el valor de una variable solamente va escrito el comando seguido del nombre de la variable.
- EJEMPLOS DE PRINT
 - print('hola mundo'); (Impresión de texto solamente)
 - print x1; (x1 es una variable)

➤ **INPUT:**

- QUE ES INPUT
Es un comando para solicitar datos del usuario.
- ESCRITURA DE INPUT

Solamente se debe escribir el comando input seguido de paréntesis y esto pedirá el dato al usuario

- EJEMPLOS DE INPUT
 - Input();
 - Variable10 = input();

➤ **CICLOS:**

- WHILE
 - QUE ES EL CICLO WHILE
El ciclo while es un bloque de código que se ejecutara repetidas veces hasta que una condicional sea cierta.
 - ESCRITURA DEL CICLO WHILE
EL ciclo while se debe escribir seguido de paréntesis en el cual estará metida la condicional a cumplir, Luego se abre llaves para meter dentro las líneas de código que se desea ejecutar repetidas veces.
 - EJEMPLOS DE WHILE
 - while(x1 != x2){ bloque de código };
 - while(x3 != 7){ bloque de código };
- DOWHILE
 - QUE ES EL CICLO DOWHILE
El ciclo dowhile es un bloque de código que se ejecutara repetidas veces hasta que una condicional sea cierta, la diferencia con while es que este código se ejecutara como mínimo una vez siempre.
 - ESCRITURA DEL CICLO DOWHILE
EL ciclo dowhile se debe escribir seguido de llaves para meter dentro las líneas de código que se desea ejecutar repetidas veces luego de paréntesis en el cual estará metida la condicional a cumplir.
 - EJEMPLOS DE WHILE
 - do{ bloque de código }while(x1 != x2);
 - do{ bloque de código }while(x3 != 7);
- FOR
 - QUE ES EL CICLO FOR
El ciclo for está programado para poder repetir un bloque de código el número de veces que nosotros asignemos, ni mas ni menos. Funciona con una variable que servirá de contador y será el numero de veces que este bloque se repetirá.
 - ESCRITURA DEL CICLO FOR
EL ciclo dowhile se debe escribir seguido de paréntesis de apertura, una declaración o asignación de variable, una condicional de esta misma variable y un sumador para esta variable, se cierra el paréntesis y se abre llaves donde ingresaremos las líneas de código que deseamos.
 - EJEMPLOS DE FOR
 - for(integer i = 0;i<5;i++){Bloque de código};
 - for(i = 10;i>5;i--){Bloque de código};

➤ **CONDICIONALES:**

- IF
 - QUE ES LA CONDICIONAL IF

La condicional if nos sirve para las ocasiones en las que deseamos que un bloque de código se ejecute solo cuando se cumpla una condición. Se pueden anidar ciclos tanto como esta condicional.

- ESCRITURA DE LA CONDICIONAL IF

La condicional if, se puede escribir de varias maneras, pero principalmente se debe escribir la palabra reservada if, seguido de la condición que deseamos verificar si es verdadera. Luego se puede agregar un else, que es un bloque de código que se ejecutara si la condición es falsa, o podemos seguir agregando if, como se ve en el ejemplo numero 3

- EJEMPLOS DE IF

- `if(x1 == 2){ bloque de código};` (if simple)
- `if(x1 == 2){ bloque de código}else{Bloque de código};` (if completo)
- `if(x1 == 2){ bloque de código}else if(condicional){Bloque de código}else{ Bloque de código};` (if compuesto)

- SWITCH

- QUE ES EL CONDICIONAL SWITCH

La condicional switch es especial para cuando deseamos verificar el valor de una variable cuando tenemos muchas opciones, esto sustituye de mejor manera al condicional if compuesto.

- ESCRITURA DEL CONDICIONAL SWITCH

EL condicional switch se puede escribir de varias maneras, principalmente como el comando switch seguido de la variable entre paréntesis, para así empezar a abrir las llaves donde irán las opciones y el código a ejecutar. Dentro de estas llaves podemos escribir la palabra reservada case seguido de un valor y dos puntos, que significa que en caso de que la variable sea igual a ese valor se ejecute el código que esta bajo el case, podemos agregar case cuantas veces queramos, pero al final debe escribirse el comando break. Existe un ultimo comando que es el default, el cual sirve en las ocasiones en las que la variable no concuerde con ningún case, el default sustituye al último case que quiere decir que bajo este se encontrara el código que se debe ejecutar en estos casos, se puede agregar break a la ultima condicional del switch o puede omitirse.

- EJEMPLOS DE SWITCH

- `switch(variable) {case 1: Código break; default: Código break;};` (switch simple con default y una opcion)
- `switch(variable) {case 1: Código break; case 2: Código break;};` (switch con dos opciones sin default)

➤ **FUNCIONES**

Las funciones son bloques de código que pueden contener cualquier sentencia dentro que solo se ejecutaran cuando estas sean llamadas, se les puede enviar parámetros y estas pueden retornar un parámetro de vuelta.

- DECLARACION DE FUNCIONES

- QUE ES UNA DECLARACION DE FUNCION

Estas pueden declararse con los valores de retorno siguientes: integer, float, character, char y void, estos valores definen de que tipo será la variable que esta función retornará. Estas funciones estarán disponibles para su uso luego de ser creadas siempre que las llamen.

- ESCRITURA DE DECLARACION DE FUNCIONES

Se debe escribir la palabra reservada para la declaración de una función la cual es "function" seguido del tipo de valor que esta retornará, en caso de no querer retornar nada se debe poner el tipo void, luego el nombre de la función y entre paréntesis los parámetros que recibirá, estos deben estar escritos con tipo de variable y su nombre y separados por comas en caso de ser más de uno, también puede no aceptar ningún parámetro. En todos los casos donde la función no sea declarada con void, pero si con otro tipo, debe contener al final del código el comando return seguido de una variable para así retornar un valor.

- EJEMPLOS DE FUNCIONES

- function void f () {Código sin return};
- function void g (integer a, character x) {Código sin return};
- function integer in () {Código return variable};
- function chain ch (character x) {Código return variable2};

- LLAMADO DE FUNCIONES

- QUE ES UN LLAMADO A FUNCION

Esto nos sirve para cuando deseamos que una función se ejecute o cuando necesitamos que una función nos retorne algún valor para almacenarlo

- ESCRITURA DE LLAMADO A FUNCIONES

Se debe escribir el nombre de la función seguido de paréntesis en los cuales puede o no haber parámetros de envío, los cuales están definidos por la función ya creada.

- EJEMPLOS DE LLAMADOS A FUNCIONES

- f ();
- g (a, x);
- variable = fun(hola,variable2);