



Manual De Programación En Astral

INTRODUCCIÓN AL LENGUAJE DE PROGRAMACIÓN ASTRAL

ÍNDICE:

1.- INTRODUCCIÓN A ASTRAL

1.1 ESTRUCTURA

2.- TIPOS DE DATOS

2.1 VARIABLES Y CONSTANTES

3.- EXPRESIÓN Y OPERADORES

3.1 OPERADOR DE ASIGNACIÓN

3.2 OPERADORES ARITMÉTICOS

3.3 OPERADORES DE COMPARACIÓN

4.- CONTROL DE FLUJO DEL PROGRAMA

4.1 Propositiones y bloques

1.- INTRODUCCIÓN A ASTRAL

Astral es un lenguaje de programación estructurada de propósito académico. En sus filas de instrucciones cuenta con estructuras ya conocidas como **IF,ELSE,WHILE** y **FOR**. Tiene bastantes características similares al lenguaje C , sin embargo destaca su inclusión a la enseñanza de la programación, que toma como principio el patrón minimalista.

1.1 ESTRUCTURA

Dentro de la estructura de Astral encontraremos herramientas y funciones tales como:

Constantes, funciones de división, ciclo for, estructura de control if, if else, mayor, mayor igual, menor, menor igual, multiplicación, print, resta, suma, total igual, variable de asignación, variable declaración, variable referencia, ciclo while.

Con todo lo mencionado anteriormente seremos capaces de realizar una basta colección de programas útiles para la enseñanza básica de la programación

2.- TIPOS DE DATOS

Los tipos de datos que se incluyen en Astral son: Tipo de dato entero, Tipo de dato string y Tipo de dato booleano, con estos tipos de datos será más que suficiente para poder lograr una correcta inclusión a la programación.

El tipo entero o **int** es el clásico entero normal, que como su nombre lo dice contiene a todos los enteros de menos a más.

El tipo de dato string como su nombre en inglés lo dice, es una cadena de caracteres que establecerá el usuario.

Por último el tipo de dato booleano, quizá el más sencillo de los tre implementados sin embargo citando al Dr. Stephen Vincent Strange "it's a simple spell but quite unbreakable" aportando una cantidad inmensa de variaciones a los programas.

2.1 VARIABLES Y CONSTANTES

Un punto ya característico es tomar en cuenta la diferencia existente entre las expresiones variables y constantes

- una expresión constante es un valor que no puede variar a lo largo del programa
- una variable es un espacio de memoria identificado por un nombre, donde se van a almacenar valores de un tipo determinado y que pueden variar a lo largo del programa.

```
var a = 2;  
var a = 7;  
if (a==2):  
    print("La variable a tiene el valor de 2");  
else:  
    print("La variable a tiene el valor de 7");  
endif
```

En el fragmento de código anterior podemos observar que la variable a toma el valor de 2 y en consecutivo se le asigna el valor de 7 por lo tanto el mensaje que aparecerá en pantalla será el de "La variable a tiene el valor de 7"

Uno de los grandes apartados en los que se diferencia Astral en comparación a C es que para declarar una variable tan solo hace falta colocar la palabra reservada **var** y a continuación el tipo de variable que se tenga en mente como por ejemplo:

var x = 6; el programa sabrá que se refiere a un int

var y = true; El programa sabrá que se refiere a un booleano

var z = "Hola mundo"; El programa sabrá que se refiere a un string

3.- EXPRESIONES Y OPERADORES

En esta sección se pretende abordar y estudiar la manera de operar con los datos anteriormente vistos. Las expresiones son operaciones con uno o más valores que se relacionan mediante operadores.

3.1 OPERADOR DE ASIGNACIÓN

Se representa con el carácter de igualdad (=).

- Sintaxis:

variable = expresión;

Donde variable es el nombre de la variable donde almacenamos el valor como resultado de la expresión.

3.2 OPERADORES ARITMÉTICOS

Los operadores aritméticos básicos son los siguientes:

- + Adición
- Sustracción
- * .. Multiplicación
- / División

3.3 OPERADORES DE COMPARACIÓN

Los operadores de comparación son los siguientes:

- < Menor que
- <= Menor o igual que
- > Mayor que
- >= MAyor o igual que
- == igual que

Los operandos se usan para crear expresiones lógicas, cuyo resultado puede tener 2 valores posibles: 1 (Cierto) ó 0 (Falso).

4.- CONTROL DE FLUJO DEL PROGRAMA

En un programa no siempre se ejecutan las mismas instrucciones y en el mismo orden. Es importante resaltar la importancia de las sentencias y proposiciones por bloques ya que con estas herramientas podemos esperar que se tomen unas u otras decisiones. Por eso es que a continuación se describen las distintas instrucciones o sentencias contenidas en Astral.

4.1 Proposiciones y bloques

Una proposición es una expresión seguida de punto y coma. Una proposición compuesta o bloque es un conjunto de declaraciones y proposiciones agrupadas y finalizadas por distintas instrucciones.

Ejemplo:

```
if((a+b) == 4):  
    print("Esto es un if");  
endif
```

```
while(c <= 3):  
    print(c);  
    c = c + 1;  
endwhile
```

```
for(c = 0; c < 3; c = c + 1):  
    print(c);  
endfor
```

El endif, endwhile, endfor, vienen a sustituir lo que serían las llaves y de esta manera darle paso al minimalismo.