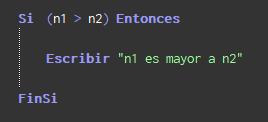
Estructuras condicionales y bucles

Diseñar un ejemplo de código con :

1. If---> Se utiliza para evaluar una expresión condicional:



si se cumple la **condición** (es **verdadera**), **ejecutará** el **bloque** de código.

si: entonces:

*Como n1(3) > n2(1) se ejecuta el bloque y se escribe por pantalla “n1 es mayor a n2”*

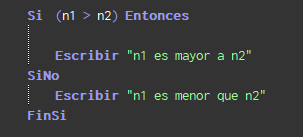
si no se cumple la **condición** (es **falsa** ), **no** se **ejecutará** el **bloque** de código.

si: no entra en el bloque por lo que sale de la expresión condicional.

*Como n1(1) < n2(3) no se ejecuta el bloque por los que no escribe por pantalla*.

------------------------------------------------------------------------------------------

2. If…else---> También se utiliza para evaluar una expresión condicional:



si se cumple la **condición** (es **verdadera**), **ejecutará** el **bloque** de código del **“*Si*”**

y se **omite** la instrucción **”*SiNo*”.**

si: entonces:

*Como n1(3) > n2(1) se ejecuta el primer bloque “Si” y escribe por pantalla*

*“n1 es mayor a n2” y sale de la expresión condicional*

Si no se cumple la **condición** “**Si**” (es **falsa** ) , pasará a **ejecutar** el **bloque** del

**“*SiNo*”.**

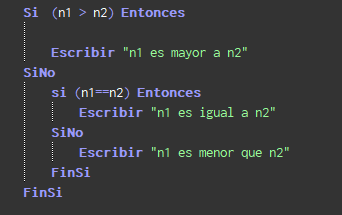
si: entonces:

*Como n1(1) < n2(3) no se ejecuta el primer bloque así que pasa al siguiente bloque el cual se ejecuta y escribe por pantalla “n1 es menor a n2”.*

*(También pasaría lo mismo si n1==n2)*

------------------------------------------------------------------------------------------

3. If…else if--->Se utiliza para evaluar una expresión condicional:



si se cumple la **condición** (es **verdadera**), **ejecutará** el **bloque** de código del **“*Si*”**

y se **omite** la instrucción **”*SiNo*”.**

si: entonces:

*Como n1(3) > n2(1) se ejecuta el bloque “Si” y escribe por pantalla*

*“n1 es mayor a n2” y sale de la expresión condicional*

Si no se cumple la **condición** “**Si**” (es **falsa** ) , pasará a **ejecutar** el **bloque** del

**“*SiNo*”, donde** volverá a **evaluar otra condición**(n1==n2).

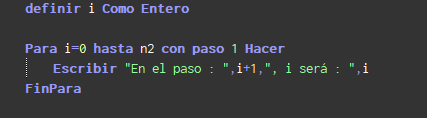
si: entonces: 

*Como n1(3) no es mayor a n2(3) no se ejecuta el primer bloque así que pasa al siguiente donde evalúa otro condicional si en este condicional el bloque”Si” es verdadero mostrará por pantalla “n1 es igual a n2” y si es falso mostrará “n1 es menor a n2”*

**

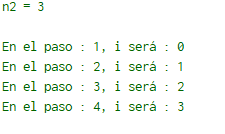
------------------------------------------------------------------------------------------

4. For--->Un bucle que se usa cuando queremos **repetir** un conjunto de **instrucciones** un **número finito de veces**.

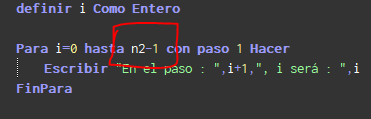


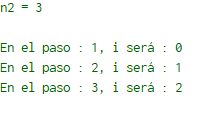
*Es una estructura con una* ***variable de control (el iterador i)*** *, una* ***condición de control(n2)*** *que es el valor final que puede tomar la variable de control y un* ***incremento (“con paso 1”)****,por último tenemos el cuerpo que es la instrucción que se hará en cada una de las iteraciones.*





*hay que estar atento porque como se inicializa i=0 hasta n2(3) no hay 3 repeticiones/pasos sino que hay 4, por lo que la estructura correcta sería :*

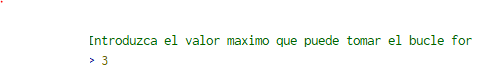
**



*Ahora si ya realiza las 3 repeticiones y en cada paso ‘i’ tiene su respectivo valor.*

*por último, se puede cambiar el incremento para ajustarlo a nuestras necesidades*

*ejemplo: en vez de ”con paso 1” se podría poner “con paso 2” ,entonces:*

**

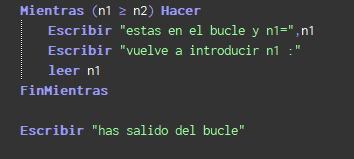
*-la primera vez que entra : i=0*

*-la segunda vez que entra : i=2*

*-y ya no entraría otra vez porque seria i=4 y el valor máximo que puede tomar es 3.*

------------------------------------------------------------------------------------------

5. While--->Unbucle que se repite mientras se cumpla una determinada condición.



**Primero** siempre **mira si se cumple la condición** y luego **si se cumple** ya **ejecuta las instrucciones** del bloque,**y si no pues omite el bloque** (es decir si n1 < n2).

para que se cumpla la condición n1 debe ser mayor o igual a n2:

Entra al bucle --->

*Como n1(3) es mayor a n2(1) se ejecuta el bloque entonces muestra por pantalla el mensaje “estás en el bucle y n1=3 ”y se vuelve a pedir que introduzca n1, introducimos ‘1’(n1=1).*

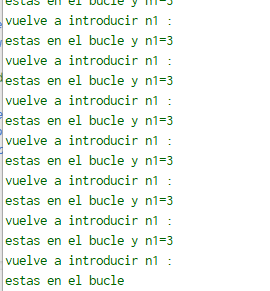
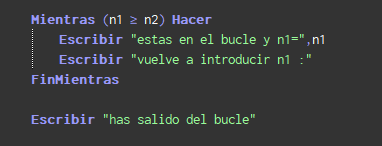


*como n1(1) es igual a n2(1) vuelve a entrar al bloque y realiza las instrucciones.*

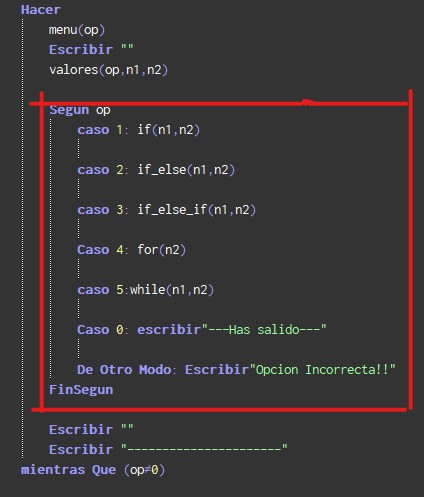
*ahora introducimos ‘0’ (n1=0) para no cumplir la condición y poder salir del bucle.*

**

*Hay que tener cuidado ya que es muy fácil entrar en un bucle infinito,en este ejemplo si al inicio de todo cuando se definió n1(3) y n2(1) no hubiese puesto la instrucción de volver a leer n1 , nunca podría salir del bucle puesto que siempre sería n1(3) >= n2(1).*

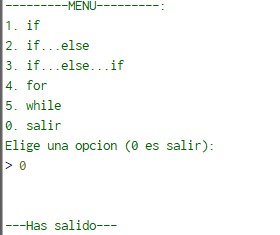
**------------------------------------------------------------------------------------------

6. Switch case--->**Mecanismo de control de selección** (que **permite definir múltiples casos**) utilizado para **cambiar el flujo de la ejecución del programa** mediante el valor de una variable o expresión

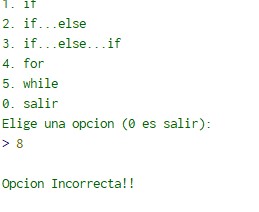


*Según el valor que tenga ‘op’ se producirá un caso u otro si op = 1 se produce el caso 1 y así sucesivamente.*

*El “caso 0” en este ejemplo se ha usado como salida.*

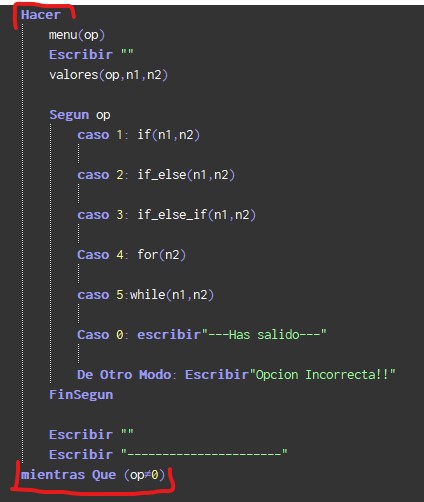


*Si se introduce cualquier otro número que no sean los que están en los casos te salta la opción “De otro modo” mostrándote que la opción es incorrecta.*



------------------------------------------------------------------------------------------

7. Do while--->Se utiliza para especificar un ciclo condicional que **se ejecuta al menos una vez** y se **repetirán** las **instrucciones mientras** se **cumpla** una determinada **condición**.



*En este caso mientras “op” sea diferente de 0 me seguirá preguntando las instrucciones del bloque/cuerpo.*