

LEARN2GROW



LEARN2GROW
Build Your own Future

¿Qué es Recursividad?

Concepto básico de Recursividad es el simple hecho de llamarse a asi mismo. El ejemplo clásico es la función que calcula la factorial de un número. Una factorial consiste en multiplicar un número natural por el número anterior, y este a su vez por el anterior, y así sucesivamente hasta llegar al número 1. Por ejemplo, la factorial de 8 sería el resultado de multiplicar 8 por 7, luego por 6 y así sucesivamente hasta llegar a uno.

Link del video

https://youtu.be/RAK-4I3xj_8

Funciones como ciclos

como funciona un ciclo? , hace una serie de tareas dando un numero de vueltas determinado , pero se puede emplear algo así en una función , pues si déjame decirte que si practico quizá no sea ya que el uso de memoria seria mucho mayor , pero se podría usar y te preguntaras entonces porque mencionarlo ? la verdad se menciona para que desde ya tengas en la mente que no existe una forma de cajón para hacer las cosas llevado a la práctica sería un poco costoso porque en la llamada de este tendrías que usar un auto sumable definir cuando termina y hacer los cálculos. **RECUERDA QUE ESTE TEMA ES SOLO PARA QUE TU SEPAS QUE SE PUEDEN REALIZAR MAS NO ES ACONSEJABLE SU USO YA QUE NO SERIAN LOS MAS EFICIENTES**

```

SubProceso val<- tabla ( num , temp )
  definir val como entero;
  val=0;
  si temp>=0
  |   escribir num " * " , temp " = " num*temp;
  |
  |   val= tabla(num , temp-1);
  |   sino
  |   |   val=0;
  |   FinSi
Fin SubProceso

Proceso X

  DEFINIR num,temp, respuesta como entero ;
  |   temp=10;
  |   escribir " ingrese el numero para mostrar la tabla ";
  |   leer num;

  respuesta = tabla(num,temp) ;

FinProceso

```

Funciones con recursividad

Se dice que una función es recursiva cuando el cuerpo de la función utiliza a la propia función. Dentro de una función recursiva suelen distinguirse dos partes: .Los casos base: Son aquellos que para su solución no requieren utilizar la función que se está definiendo. .Los casos recursivos: Son aquellos que sí que requieren utilizar la función que se está definiendo. .Las definiciones recursivas funcionan siempre y cuando las llamadas recursivas se realicen de forma que en algún momento se lleguen a los casos base.

```

funcion factorial (n)
{
  si n == 1:
    return 1

  si no:
    return n*factorial(n-1)
}

```

¿se puede usar un proceso para recursividad?

Pues este es igual que el tema anterior se pueden hacer mas no quiere decir que este sea el mas factible en rederizar , porque para hacer esto se necesitaría hacer dos procesos que se llamen entre si hasta que se cumpla que termina el ciclo lo cual acá seria más difícil de procesar ya que debemos enviar datos entre ambos procesos para así no perder el contra de lo que se está realizando **RECUERDA QUE ESTE TEMA ES SOLO PARA QUE TU SEPAS QUE SE PUEDEN REALIZAR MAS NO ES ACONSEJABLE SU USO YA QUE NO SERIAN LOS MAS EFICIENTES**

Buenas prácticas en recursividad

de las mejores prácticas que puedes usar y no solo en recursividad es el comentar el código y dejar explicito que hace y cuál es la variable que se usa. El hacerlo en papel primero el planteo de cómo se realizaría de esa forma. El manejo de la salida porque al igual que en los ciclos podrías crear un ciclo infinito del cual no podrías salir de no ser porque Pseint te permite el paro total del programa pero imagina que de alguna forma esto llegara a un usuario , esto crearía un bug del cual saldría hasta que se acaba la memoria manejo de las variables con nombres reconocibles.



