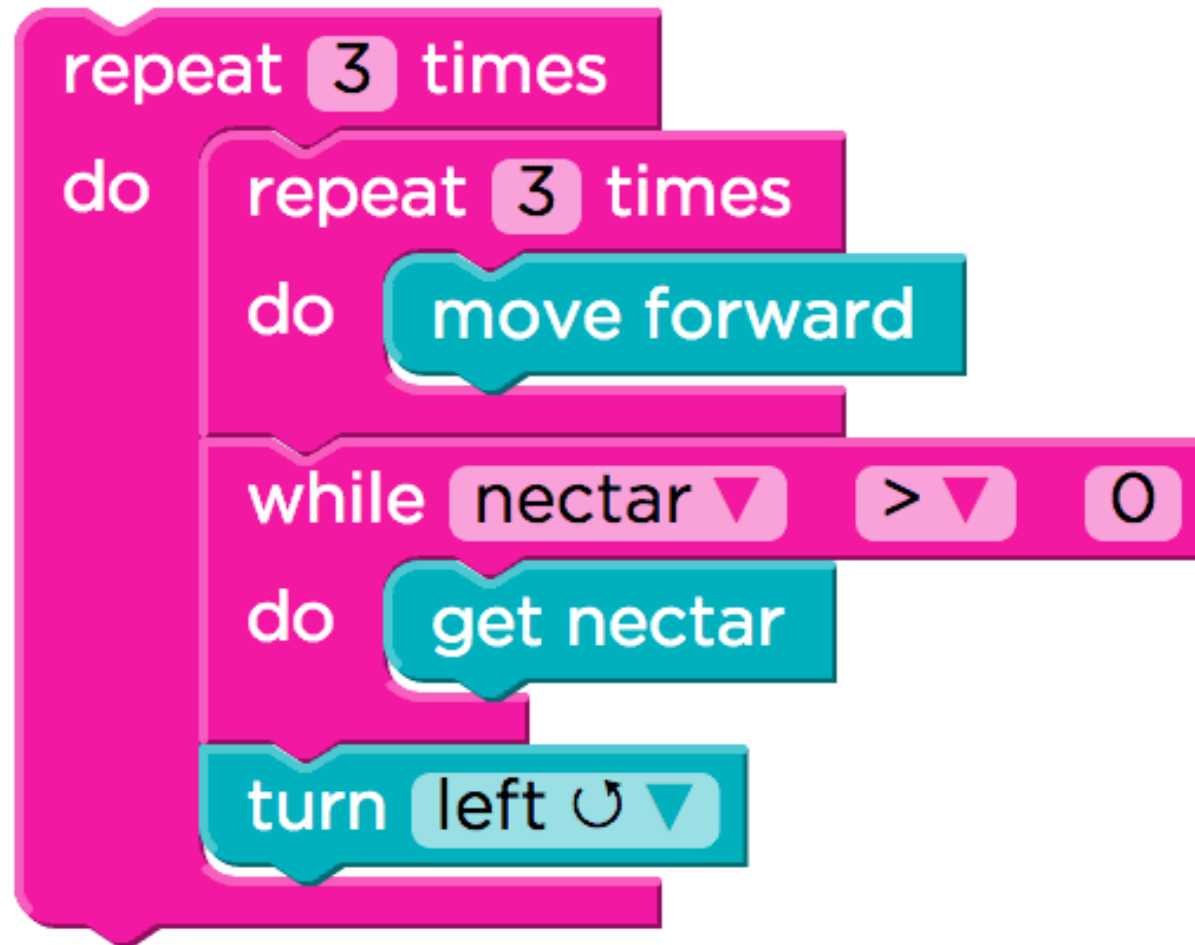


Unidad 6: Estructuras de Control Bucles Anidados

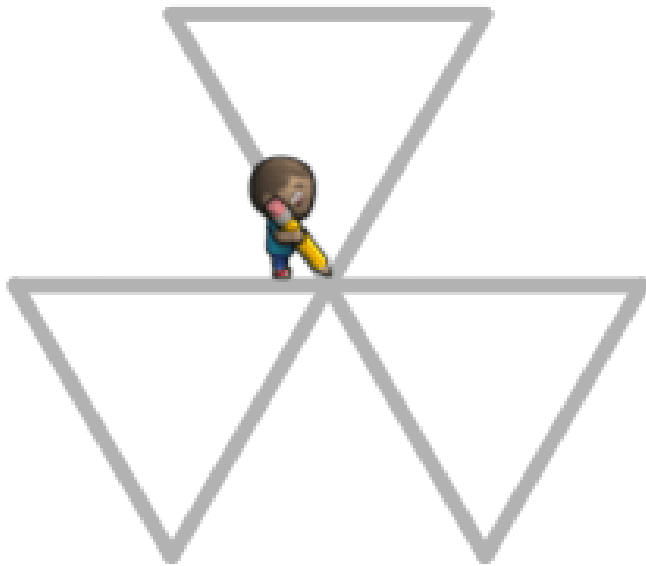
CS1100 - Introducción a Ciencia de la Computación
UTEC

¿Bucles dentro de bucles?



Varias tareas dentro de tareas

Un ejemplo con code.org



<https://studio.code.org/s/course2/stage/19/puzzle/1/>

Logro de la Sesión

Al finalizar esta sesión, estarás en la capacidad de:

- Resolver problemas combinando bucles y condiciones

Logro de la Sesión

Al finalizar esta sesión, estarás en la capacidad de:

- Resolver problemas combinando bucles y condiciones
- Entender bucles interior y exterior.

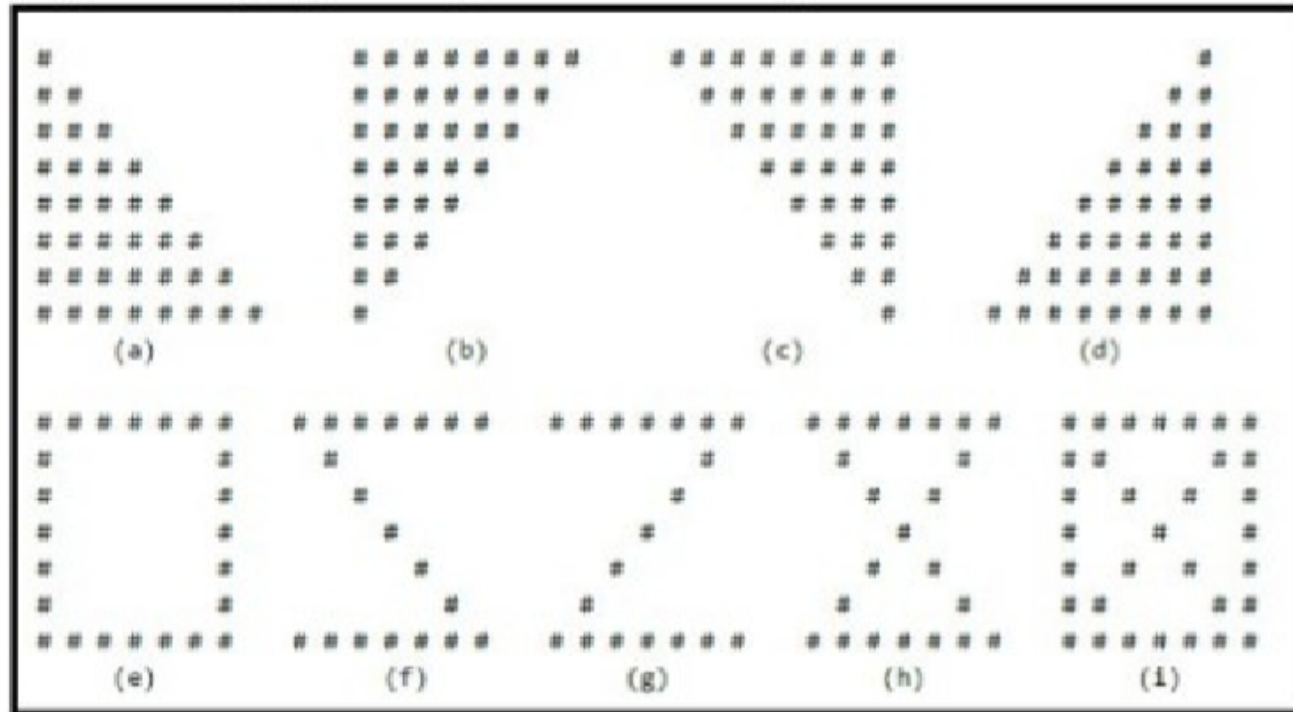
Logro de la Sesión

Al finalizar esta sesión, estarás en la capacidad de:

- Resolver problemas combinando bucles y condiciones
- Entender bucles interior y exterior.
- Identificar la aplicación de bucles anidados en algunos problemas.

Bucles anidados!

Using nested loops, write a program that prints the following patterns to console:



```
1 for i in range(5):
2     for j in range(5):
```

^^I

Bucles anidados (Nested Loop)

Los bucles anidados o nested loop son bucles dentro de otros bucles. Se puede combinar cualquier sentencia condicional o estructura de control dentro de otra. La finalidad es expresar relaciones entre condiciones y/o repeticiones. Veamos este ejemplo con 2 sentencias for anidadas.

```
1 for x in range(1, n):  
2     for y in range(1, m):  
3         print("*", end=" ")  
4     print()
```

¿Cual es el resultado de estos 2 bucles for?.

Bucles anidados (Nested Loop)

Observemos la relación entre ambos bucles. El primer bucle que aparece se le denomina exterior. El segundo bucle que aparece debajo e indentado se le conoce como bucle interior. Durante la ejecución de este algoritmo por cada ejecución del bucle exterior se ejecutará completamente el bucle interior. Es decir si el bucle exterior se ejecuta n veces y el interior m veces. El total de instrucciones será $n*m$ veces. Las sentencias dentro del bucle interior se ejecutará $n*m$ veces.

```
1 for x in range(1, n):  
2     for y in range(1, m):  
3         print("*", end=" ")  
4     print()
```

La cantidad de veces que se ejecuta un bucle dentro de otro tiene implicancias en el tiempo de ejecución de los algoritmos. A mayor nivel de bucles anidados mayor cantidad de operaciones y con ello mayor uso de recursos (tiempo y memoria) del algoritmo.

Bucles anidados (Nested Loop)

Veamos ahora otro ejemplo. La factorización de primos. La factorización de un numero primo es identificar todos los numeros primos que multiplicados forman el número. Por ejemplo $3757208 = 2*2*2*7*13*13*397$. ¿Analicen que hace el bucle interno y que hace el bucle externo?

```
1  n = int(input())
2  x = 2
3  while x <= n//x:
4      while n % x == 0:
5          n //= x
6          print(x, end = " ")
7      x = x + 1
8  if (n>1):
9      print(n)
```

```
1  3757208
2  2 2 2 7 13 13 397
```

Ejercicio 1

Enunciado

Diseñe un algoritmo en Python, que genere una tabla de $n \times n$ de numeros del 1 al 9 inclusive, si n es mayor a 9 se repite la secuencia desde 1:

- imprima una linea de n numeros.
- imprima n filas, use `print()` para generar el salto de linea.

Ejercicio 2

Enunciado

Diseñe un algoritmo en Python que dibuje un triangulo rectangulo tal como aparece en el siguiente resultado para un valor n:

Ejemplo n = 5

```
#  
##  
###  
####  
#####
```

Ejercicio 3

Enunciado

Diseñe un algoritmo que solicite un número N entero y usa dos For anidados para imprimir una tabla de $N \times N$ con * en filas y columnas si es que el número i divide a j o viceversa. Las variables i y j deben ser los índices de los for. En la última columna se imprime el número de la fila.

```
Sample input: 3
* * * 1
* *   2
*    * 3
```

Ejercicio 4

Enunciado

Diseñe un algoritmo en Python que imprima un arbol de los numeros naturales del 1 al N. el programa recibe un numero N con input e imprime una linea conteniendo la secuencia de numeros primero del 1 luego del 1 2 luego del 1 2 3 hasta llegar a la fila hasta el numero N

```
sample input: 5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```


Evaluación

Individual Work

- <https://www.hackerrank.com/contests/cs1100-lab-05a>

Cierre

En esta sesión aprendiste:

- Desarrollar programas con bucles anidados.

Cierre

En esta sesión aprendiste:

- Desarrollar programas con bucles anidados.
- Reconocer bucles exterior e interior.

Cierre

En esta sesión aprendiste:

- Desarrollar programas con bucles anidados.
- Reconocer bucles exterior e interior.
- Imprimir tablas de datos.

Cierre

En esta sesión aprendiste:

- Desarrollar programas con bucles anidados.
- Reconocer bucles exterior e interior.
- Imprimir tablas de datos.
- Usar indistintamente `for` o `while` para anidar bucles.