**UNIVERSIDAD PONTIFICIA BOLIVARIA SECCIONAL BUCARAMANGA**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**LÓGICA DE PROGRAMACIÓN**

**JUAN JOSÉ DÍAZ PRADA**

# **Práctica For y función Range.**

Lab 2.2 (Segundo laboratorio del segundo corte)

Objetivos:

* El estudiante estará en capacidad aplicar los conceptos estudiados en clase sobre el uso de For y de Range para resolver problemas en el lenguaje Python.
* El estudiante estará en capacidad de subir a su Git el código realizado en el laboratorio y compartir código fuente con sus compañeros de clase.

**Resuelve las siguientes cuestiones y en el espacio de la clase designado para ello, comparte tus respuestas (la participación hace parte de la evaluación).**

1. Responde las siguientes preguntas, puedes consultar en internet los conceptos que creas convenientes, redacta tus respuestas y sube a tu Git el código fuente asociado con la práctica.
   1. Escribe al frente de cada función Range planteada el valor de salida de acuerdo con los parámetros dados:

|  |  |
| --- | --- |
| range(9) | (1,2,3,4,5,6,7,8) |
| range(3, 13) | (3,4,5,6,7,8,9,10,11,12) |
| range(3, 13, 2) | (3,5,7,9,11) |
| range(11, 4) | - |
| range(11, 4, -1) | (11,10,9,8,7,6,5) |
| range(2, 10, 2) | (2,4,6,8) |
| range(2, 10, 3) | (2,5,8) |
| range(10, 5, -1) | (10,9,8,7,6) |
| range(3, -1, -1) | (3,2,1,0) |

* 1. ¿Qué hace el siguiente código? Describe lo que hace cada línea y da ejemplos de la salida de este.

**for** i **in** range(21): Se toman los números del 0 al 21 (sin contar el el 21)

**print** i, i \*\* 3 Se imprime cada número del rango, junto con su potencia a la 3

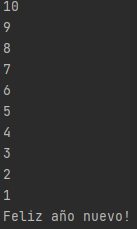
 Etc…

* 1. ¿Qué hace el siguiente código? Describe lo que hace cada línea y muestra la salida de este.

**for** i **in** range(10, 0, -1): Se toman los numeros del 10 al 0, como una cuenta regresiva

**print** i Se imprime el rango

**print** 'Feliz anio nuevo!' Al finalizar el bucle, se imprime “Feliz anio nuevo!”



1.4 ¿Qué hace el siguiente código? Describe lo que hace cada línea y muestra la salida de este.

**for** i **in** [0,1,2,3,4]: Se crea un rango con los números escritos (o sea, del 0 al 4)

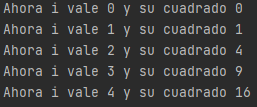
**print** "Hola" Se imprime “Hola” por cada element en el rango



1.5 ¿Qué hace el siguiente código? Describe lo que hace cada línea y muestra la salida de este.

for i in range(5): Se toman los números del 0 al 5 sin contar el 5

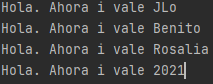
  print "Ahora i vale",i,"y su cuadrado", i\*\*2 Se reemplaza i por su valor dentro del rango y se muestra su cuadrado



1.6 ¿Qué hace el siguiente código? Describe lo que hace cada línea y muestra la salida de este.

for i in ['JLo', 'Benito', 'Rosalia', 2021]: Se crea un rango con las cadenas “JLo”, “Benito”, “Rosalia”, y el entero 2021”

  print "Hola. Ahora i vale", i Se imprime el saludo y se reemplaza i con los valores del rango



1.7 Escribe el código para un bucle tipo for el cual imprime desde número 0 hasta el 7. Utiliza una variable auxiliar llamada n.





1.8 Modifica el rango del bucle anterior para que ahora imprima del número 1 hasta el 12.





1.9 El siguiente programa cuenta cuantos de los números naturales menores que mil tienen un cubo terminado en siete. Explica cada línea de código e impleméntalo:

c = 0 Se define la variable c como 0

**for** i **in** range(1000): Se crea un rango del 0 al 1000 sin tomar el 1000

    ultimo\_digito = (i \*\* 3) % 10 Se define la variable ultimo\_digito como i elevado a la 3, y se saca su modulo en 10

**if** ultimo\_digito == 7: Si la variable ultimo\_digito es igual a 7, se le suma 1 a la variable c

        c = c + 1

**print** c Se imprime c

 etc…

**2. Bucles anidados**

Un bucle anidado es un bucle situado en el cuerpo de otro bucle. Por ejemplo, el programa:

for i in range (3):

  for j in range (2):

    print "i vale", i, "y j vale", j

escribiría:

i vale 0 y j vale 0

i vale 0 y j vale 1

i vale 1 y j vale 0

i vale 1 y j vale 1

i vale 2 y j vale 0

i vale 2 y j vale 1

En este caso, el bucle externo (el controlado por i) se ejecuta 3 veces y el bucle interno (el controlado por j) se ejecuta dos veces por cada valor de i. Por ello la instrucción print se ejecuta en total 6 veces (3 veces que se ejecuta el bucle externo x 2 veces que se ejecuta cada vez el bucle interno = 6 veces).

Se pueden anidar tanto bucles como se desee.

La variable del bucle externo puede controlar el bucle interno. Por ejemplo, el programa:

for i in range (4):

  for j in range(i):

    print ("i vale", i, "y j vale", j)

escribiría:

i vale 1 y j vale 0

i vale 2 y j vale 0

i vale 2 y j vale 1

i vale 3 y j vale 0

i vale 3 y j vale 1

i vale 3 y j vale 2

La variable i toma los valores de 0 a 3 y la variable j toma los valores de 0 a i, por lo que cada vez el bucle interno se ejecuta un número diferente de veces:

• Cuando i vale 0, range(i) devuelve una lista vacía, y por tanto el bucle interno no se ejecuta ninguna vez y el programa no escribe que i vale 0.

• Cuando i vale 1, range(i) devuelve la lista [0] y por tanto el bucle interno se ejecuta una sola vez y el programa escribe una sola línea en la que i vale 1 (y j vale 0).

• Cuando i vale 2, range(i) devuelve la lista [0, 1] y por tanto el bucle interno se ejecuta dos veces y el programa escribe dos líneas en la que i vale 2 (y j vale 0 o 1 en cada una de ellas).

• Cuando i vale 3, range(i) devuelve la lista [0, 1, 2] y por tanto el bucle interno se ejecuta tres veces y el programa escribe tres líneas en la que i vale 3 (y j vale 0, 1 o 2 en cada una de ellas).

2.1 Implementa dos ejemplos en tu Git de bucles anidados.

3. **Jupyter** 

Jupyter es un proyecto heredado de la consola IPython la cual ha evolucionado y ha integrado sus notebooks en él. Así, Jupyter es una aplicación web que aporta una interfaz para el desarrollo de diferentes tipos de códigos de programación (su nombre es resultado de la fusión de los lenguajes de programación Julia, Python y R) y permite su interpretación, así como la visualización de gráficas, mapas, etc.

Un resumen de los beneficios de Jupyter Notebook:

* Código abierto (licencia BSD modificada)
* Gratuito
* Funciona en el navegador
* Código en vivo (Live-Code)
* Diferentes opciones a la hora de exportar y compartir los resultados
* Control de versiones
* Permite colaboración (JupyterHub)
* Soporta más de 50 lenguajes de programación

Tomado de [Jupyter Notebook: la herramienta de Python para procesar datos - IONOS](https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/jupyter-notebook/)

Hay una manera de probar Jupyter en la web sin necesidad de instalar nada más si se entra en la opción JupyterLab:

[Project Jupyter | Try Jupyter](https://jupyter.org/try)

También, es posible instalar Jupyter Notebook en tu computadora:

[Install Python 🐍 with Anaconda: Jupyter Notebook and Spyder - YouTube](https://www.youtube.com/watch?v=LrMOrMb8-3s)

3.1 Prueba Jupyter Notebook on line y si lo crees conveniente, instálalo en tu máquina.

NOTA:

Debes adjuntar una bibliografía donde conste los sitios web, libros o artículos que consultaste para responder el taller.

Debes responder al taller en esta guía y además actualizar tu Git con el respectivo código fuente. Crea una carpeta de **Laboratorios** donde las irás guardando con la fecha de su realización y el respectivo número (la de hoy es Lab22\_2403).