**Carpeta: PYx01**

1. **0-positive\_or\_negative.py:**

Este programa asignará un número aleatorio con signo a la variable **número** cada vez que se ejecute. Complete el código fuente para imprimir si el número almacenado en la variable es positivo o negativo.

* Código fuente:

import random

numero = random.randint(-10, 10)

# PONER ACÁ TU CÓDIGO

* La variable **número** almacenará un valor diferente cada vez que ejecutes el programa
* No tienes que tocar ni entender que hace **import, random** ni **randint**
* La salida del programa debe ser:
  + **el número,** seguido de:
    - Si el número es mayor que cero: **es positivo**
    - Si el número es cero: **es cero**
    - Si el número es menor que cero: **es negativo**
  + Seguido de una nueva línea

1. **1-last\_digit.py**

Este programa asignará un número con signo aleatorio a la variable **número** cada vez que se ejecute. Complete el código fuente para imprimir el último dígito del número almacenado en la variable.

* Código fuente

import random

numero = random.randint(-10000, 10000)

# PONER ACÁ TU CÓDIGO

* La variable **número** almacenará un valor diferente cada vez que ejecutes este programa
* La salida del programa debe ser:
  + El texto **el último dígito de**, seguido de:
  + El **número**, seguido de:
  + El texto **es**, seguido de el último dígito del **número**, seguido de:
    - Un texto final que diga si el último dígito es:
    - **y es mayor que 5**
    - **y es cero**
    - **y es menor que 6 y no es 0**

1. **2-print\_alphabet.py**

Escribe un programa que imprima el alfabeto ASCII, en minúsculas.

* Sólo puede utilizar una vez la función **print**
* Sólo puede utilizar un bucle en su código
* No puede almacenar caracteres en una variable
* No puede importar ningún módulo
* La salida no puede terminar en una nueva línea

1. **3-print\_alphabet.py**

Escribe un programa que imprima el alfabeto ASCII, en minúsculas, no seguido de una nueva línea.

* + Imprime todas las letras excepto **q** y **e**
  + Sólo puede utilizar una vez la función **print**
  + Sólo puede utilizar un bucle en su código
  + No puede almacenar caracteres en una variable
  + No puede importar ningún módulo
  + La salida no puede terminar en una nueva línea
  + El código debe tener exactamente 3 líneas

1. **4-print\_hexa.py**

Escribe un programa que imprima todos los números del 0 al 98 en decimal y en hexadecimal (como en el siguiente ejemplo)

0 = 0x0

1 = 0x1

…

8 = 0x8

9 = 0x9

10 = 0xa

...

97 = 0x61

98 = 0x62

* Sólo puede utilizar una vez la función **print**
* Sólo puede utilizar un bucle en su código
* No está permitido almacenar números o cadenas en una variable
* No está permitido importar ningún módulo

1. **5-print\_comb2.py**

Escribe un programa que imprima números del **0** al **99**.

* Los números deben ir separados con una coma, seguido de un espacio
* Los números deben imprimirse en orden ascendente, con dos dígitos
* El último número debe ir seguido de una nueva línea
* Sólo puede utilizar un máximo de 2 funciones **print**
* Sólo puede utilizar un bucle en su código
* No está permitido almacenar números o cadenas en una variable
* No está permitido importar ningún módulo

1. **6-print\_comb3.py**

Escribe un programa que imprima todas las posibles combinaciones diferentes de dos dígitos.

* Los números deben ir separados con una coma, seguido de un espacio
* Las dos cifras deben ser diferentes
* **01** y **10** se consideran la misma combinación de los dos dígitos **0** y **1**
* Imprimir sólo la combinación más pequeña de dos dígitos
* Los números deben imprimirse en orden ascendente, con dos dígitos
* El último número debe ir seguido de una nueva línea
* Sólo puede utilizar un máximo de 3 funciones **print**
* Sólo puede utilizar un máximo de 2 bucles en su código
* No está permitido almacenar números o cadenas en una variable
* No está permitido importar ningún módulo