

Evidencias de laboratorios de curso python

Cruz Breña Daniela Janeth

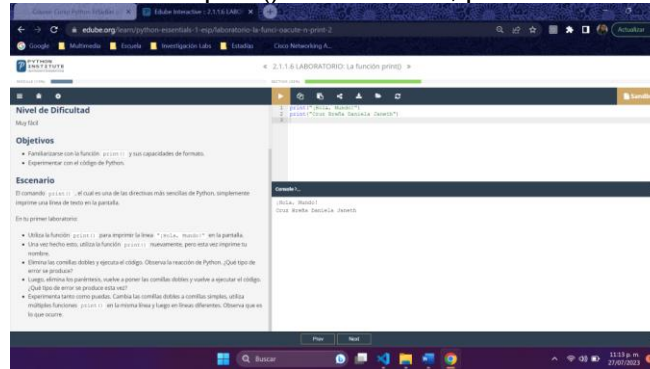
• **Módulo 2: Tipos de datos, variables, operaciones básicas de entrada y salida y operaciones básicas**

1. Laboratorio 1

El comando `print()`, el cual es una de las directivas más sencillas de Python, simplemente imprime una línea de texto en la pantalla.

En tu primer laboratorio:

- Utiliza la función `print()` para imprimir la línea "¡Hola, Mundo!" en la pantalla.
- Una vez hecho esto, utiliza la función `print()` nuevamente, pero esta vez imprime tu nombre.



- Elimina las comillas dobles y ejecuta el código. Observa la reacción de Python. ¿Qué tipo de error se produce?

```
1 print(¡Hola, Mundo!)
2 print(Cruz Breña Daniela Janeth)
3
```

```
Console>_
¡Hola, Mundo!
Cruz Breña Daniela Janeth
File "main.py", line 1
  print(¡Hola, Mundo!)
        ^
SyntaxError: invalid character in identifier
```

Se produce un error de sintaxis, ya que los caracteres no son identificados como una función existente en Python

- Luego, elimina los paréntesis, vuelve a poner las comillas dobles y vuelve a ejecutar el código. ¿Qué tipo de error se produce esta vez?

```
1 print"Cruz Breña Daniela Janeth"
2
```

```
Console>_
¡Hola, Mundo!
Cruz Breña Daniela Janeth
File "main.py", line 1
  print"Cruz Breña Daniela Janeth"
    ^
SyntaxError: invalid character in identifier
File "main.py", line 1
  print"Cruz Breña Daniela Janeth"
    ^
SyntaxError: invalid syntax
```

Se produce un error de sintaxis, ya que los caracteres no son identificados como una función existente en Python

2. Laboratorio 2

- Modifica la primera línea de código en el editor, utilizando las palabras clave `sep` y `end`, para que coincida con el resultado esperado. Recuerda, utilizar dos funciones `print()`.

The screenshot shows the Edube Interactive interface for the lab "2.1.1.18 LABORATORIO: La función print()". The left sidebar contains the following information:

- Tiempo Estimado:** 5-10 minutos
- Nivel de Dificultad:** Muy fácil
- Objetivos:**
 - Familiarizarse con la función de `print()` y sus capacidades de formato.
 - Experimentar con el código de Python.
- Escenario:** Modifica la primera línea de código en el editor, utilizando las palabras clave `sep` y `end`, para que coincida con el resultado esperado. Recuerda, utilizar dos funciones `print()`. No cambies nada en la segunda invocación de `print()`.
- Salida Esperada:** Fundamentos***Programación***en...Python

The main editor area shows the following code:

```
1 print("Fundamentos", "Programación", "en", sep="***", end="...")
2 print("Python")
3
```

The console output shows: Fundamentos***Programación***en...Python.

3. Laboratorio 3

Intenta:

- Minimizar el número de invocaciones de la función `print()` insertando la secuencia `\n` en las cadenas.
- Hacer la flecha dos veces más grande (pero mantener las proporciones).

The screenshot shows the Edube Interactive interface for the lab "2.1.1.19 LABORATORIO: Dando formato a la salida". The left sidebar contains the following information:

- Objetivos:**
 - Experimentar con el código Python existente.
 - Descubrir y solucionar errores básicos de sintaxis.
 - Familiarizarse con la función `print()` y sus capacidades de formato.
- Escenario:** Recomendamos que juegues con el código que hemos escrito para ti y que realices algunas correcciones (quizás incluso destructivas). Siéntete libre de modificar cualquier parte del código, pero hay una condición: aprende de tus errores y saca tus propias conclusiones.

The main editor area shows the following code:

```
1 print(" *")
2 print(" *")
3 print(" *")
4 print(" *")
5 print(" *")
6 print(" *")
7 print(" *")
8 print(" *")
9
10 print(" *", end="\n *")
11 print("\n *", end="\n *")
12 print(" *", end="\n *")
```

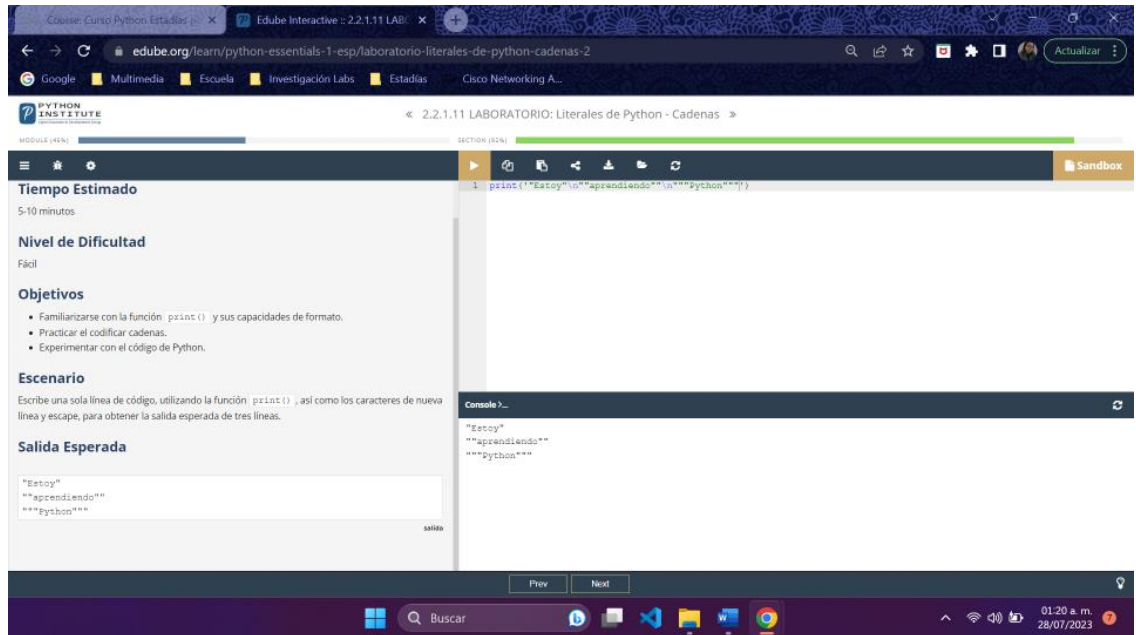
The console output shows a pattern of asterisks forming a shape.

The "Intenta:" section contains the following instructions:

- Minimizar el número de invocaciones de la función `print()` insertando la secuencia `\n` en las cadenas.
- Hacer la flecha dos veces más grande (pero mantener las proporciones).
- Duplicar la flecha, colocando ambas flechas lado a lado; nota: una cadena se puede multiplicar usando el siguiente truco: "cadena" * 2 producirá "cadenacadena" (te contaremos más sobre ello pronto).
- Elimina cualquiera de las comillas y observa detenidamente la respuesta de Python; presta atención a donde Python ve un error; ¿es el lugar en donde realmente existe el error?
- Haz lo mismo con algunos de los paréntesis.
- Cambia cualquiera de las palabras `print` en otra cosa (por ejemplo de minúscula a mayúscula, `Print`); ¿Qué sucede ahora?
- Reemplaza algunas de las comillas por apóstrofes; observa lo que pasa detenidamente.

4. Laboratorio 4

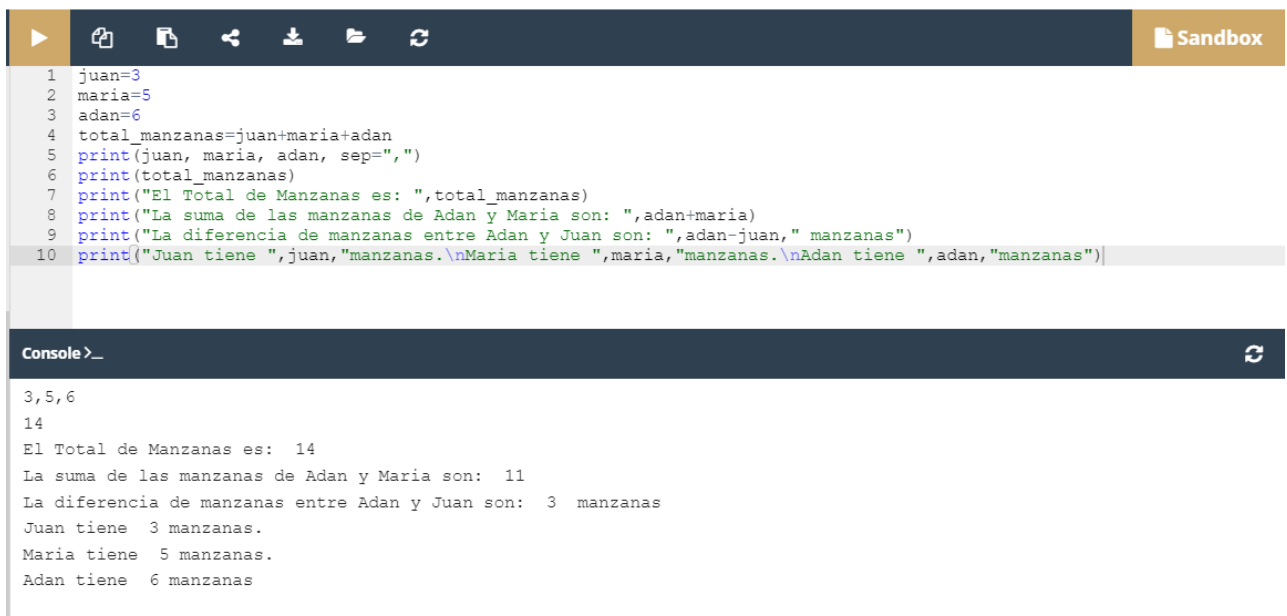
Escribe una sola línea de código, utilizando la función `print()`, así como los caracteres de nueva línea y escape, para obtener la salida esperada de tres líneas.



5. Laboratorio 5

Tu tarea es:

- Crear las variables: `juan`, `maria`, y `adan`.
- Asignar valores a las variables. El valor debe de ser igual al número de manzanas que cada quien tenía.
- Una vez almacenados los números en las variables, imprimir las variables en una línea, y separar cada una de ellas con una coma.
- Después se debe crear una nueva variable llamada `total_manzanas` y se debe igualar a la suma de las tres variables anteriores.
- Imprime el valor almacenado en `total_manzanas` en la consola.
- Experimenta con tu código: crea nuevas variables, asigna diferentes valores a ellas, y realiza varias operaciones aritméticas con ellas (por ejemplo, `+`, `-`, `*`, `/`, `//`, etc.). Intenta poner una cadena con un entero juntos en la misma línea, por ejemplo, "Número Total de Manzanas:" y `total_manzanas`.



6. Laboratorio 6

Millas y kilómetros son unidades de longitud o distancia.

Teniendo en mente que 1 milla equivale aproximadamente a 1.61 kilómetros, complementa el programa en el editor para que convierta de:

- Millas a kilómetros.
- Kilómetros a millas.

No se debe cambiar el código existente. Escribe tu código en los lugares indicados con ###. Prueba tu programa con los datos que han sido provistos en el código fuente

```
1 kilometers = 12.25
2 miles = 7.38
3
4 miles_to_kilometers = miles*1.61
5 kilometers_to_miles =kilometers/ 1.61
6
7 print(miles, "millas son", round(miles_to_kilometers, 2), "kilómetros")
8 print(kilometers, "kilómetros son", round(kilometers_to_miles, 2), "millas")
9
```

```
Console>_
7.38 millas son 11.88 kilómetros
12.25 kilómetros son 7.61 millas
```

7. Laboratorio 7

Observa el código en el editor: lee un valor flotante, lo coloca en una variable llamada x, e imprime el valor de la variable llamada y. Tu tarea es completar el código para evaluar la siguiente expresión:

$$3x^3 - 2x^2 + 3x - 1$$

```
1 print("PRIMER VALOR DE PRUEBA")
2 x1 = 0
3 x1 = float(x1)
4 y1=(3*x1**3)-(2*x1**2)+(3*x1)-1
5 print("y =", y1)
6
7 print("SEGUNDO VALOR DE PRUEBA")
8 x2 = 1
9 x2 = float(x2)
10 y2=(3*x2**3)-(2*x2**2)+(3*x2)-1
11 print("y =", y2)
12
13 print("TECER VALOR DE PRUEBA")
14 x3 = -1
15 x3 = float(x3)
16 y3=(3*x3**3)-(2*x3**2)+(3*x3)-1
17 print("y =", y3)
```

```
Console>_
PRIMER VALOR DE PRUEBA
y = -1.0
SEGUNDO VALOR DE PRUEBA
y = 3.0
TECER VALOR DE PRUEBA
y = -9.0
```

8. Laboratorio 8

El código en el editor contiene comentarios. Intenta mejorarlo: agrega o quita comentarios donde consideres que sea apropiado (en ocasiones el remover un comentario lo hace más legible), además, cambia el nombre de las variables donde consideres que esto mejorará la comprensión del código.

```
1 # este programa calcula los segundos en cierto número de horas determinadas
2
3 horas= 2 # número de horas
4 segundos = 3600 # número de segundos en una hora
5
6 print("Horas: ", horas) #imprime el numero de horas
7 print("Segundos en Horas: ", horas * segundos) # se imprime el numero de segundos en determinado numero de horas
8 print("Adios")
9
```

Console>_

```
Horas: 2
Segundos en Horas: 7200
Adios
```

9. Laboratorio 9

La tarea es completar el código para evaluar y mostrar el resultado de cuatro operaciones aritméticas básicas.

```
1 num1=float(input("Ingresa el primer número: "))
2 num2=float(input("Ingresa el segundo número: "))
3
4 print("La suma de ",num1," y ",num2, "es igual a:", num1+num2)
5 print("La resta de ",num1," y ",num2, "es igual a:", num1-num2)
6 print("La multiplicación de ",num1," y ",num2, "es igual a:", num1*num2)
7 print("La división de ",num1," y ",num2, "es igual a:", num1/num2)
8
9 print("\n;Eso es todo, amigos!")
10
```

Console>_

```
Ingresa el primer número: 2
Ingresa el segundo número: 3
La suma de 2.0 y 3.0 es igual a: 5.0
La resta de 2.0 y 3.0 es igual a: -1.0
La multiplicación de 2.0 y 3.0 es igual a: 6.0
La división de 2.0 y 3.0 es igual a: 0.6666666666666666
;Eso es todo, amigos!
```

10. Laboratorio 10

La tarea es completar el código para poder evaluar la siguiente expresión:

```
1 print("-----")
2 print("Valor Prueba 1")
3 print("-----")
4 x = float(input("Ingresa el valor para x: "))
5 y1=1/((x+(1/(x+(1/(x+(1/x)))))))
6 print("y =", y1)
7 print("-----")
8 print("Valor Prueba 2")
9 print("-----")
10 x = float(input("Ingresa el valor para x: "))
11 y2=1/((x+(1/(x+(1/(x+(1/x)))))))
12 print("y =", y2)
13 print("-----")
14 print("Valor Prueba 3")
15 print("-----")
16 x = float(input("Ingresa el valor para x: "))
17 y3=1/((x+(1/(x+(1/(x+(1/x)))))))
18 print("y =", y3)
19 print("-----")
20 print("Valor Prueba 4")
21 print("-----")
22 x = float(input("Ingresa el valor para x: "))
23 y4=1/((x+(1/(x+(1/(x+(1/x)))))))
24 print("y =", y4)
25
```

Valor Prueba 1

Ingresa el valor para x: 1
y = 0.6000000000000001

Valor Prueba 2

Ingresa el valor para x: 10
y = 0.09901951266867294

Valor Prueba 3

Ingresa el valor para x: 100
y = 0.009999000199950014

Valor Prueba 4

Ingresa el valor para x: -5
y = -0.19258202567760344

11. Laboratorio 11

La tarea es preparar un código simple para evaluar o encontrar el tiempo final de un periodo de tiempo dado, expresándolo en horas y minutos. Las horas van de 0 a 23 y los minutos de 0 a 59. El resultado debe ser mostrado en la consola.

```
1 hour = int(input("Hora de inicio (horas): "))
2 mins = int(input("Minuto de inicio (minutos): "))
3 dura = int(input("Duración del evento (minutos): "))
4
5 # Escribe tu código aquí.
6 mins=mins+dura
7 hour=hour+mins//60
8
9 print(hour%24,":",mins%60)
10
```

- **Módulo 3: Valores booleanos, ejecución condicional, bucles, listas y su procesamiento, operaciones lógicas de bit a bit**

1. Laboratorio: 1

Usando uno de los operadores de comparación en Python, escribe un programa simple de dos líneas que tome el parámetro n como entrada, que es un entero, e imprime False si n es menor que 100, y True si n es mayor o igual que 100.

```
1 #Ingresar el numero
2 num=int(input("Ingresa el Número: "))
3 print(num>100)
4
5
```

```
Console>_
Ingresa el Número: 55
False
Ingresa el Número: 99
False
Ingresa el Número: 100
False
Ingresa el Número: 101
True
Ingresa el Número: -5
False
Ingresa el Número: +123
True
```

2. Laboratorio: 2

Escribe un programa que utilice el concepto de ejecución condicional, tome una cadena como entrada y que:

- Imprima el enunciado "Si, ¡El ESPATIFILIO! es la mejor planta de todos los tiempos!" en la pantalla si la cadena ingresada es "ESPATIFILIO".
- Imprima "No, ¡quiero un gran ESPATIFILIO!" si la cadena ingresada es "espatifilo".
- Imprima "¡ESPATIFILIO!, ¡No [entrada]!" de lo contrario. Nota: [entrada] es la cadena que se toma como entrada.

```
1 entrada=input("Ingresa la Palabra: ")
2 if entrada == "ESPATIFILIO":
3     print("Si, ¡El ESPATIFILIO! es la mejor planta de todos los tiempos!")
4 elif entrada == "espatifilio":
5     print("No, ¡quiero un gran ESPATIFILIO!")
6 if entrada != "ESPATIFILIO":
7     if entrada != "espatifilio":
8         print("¡ESPATIFILIO!, ¡No",entrada,"!")
```

```
Console>_
Ingresa la Palabra: ESPATIFILIO
Si, ¡El ESPATIFILIO! es la mejor planta de todos los tiempos!
Ingresa la Palabra: espatifilio
No, ¡quiero un gran ESPATIFILIO!
Ingresa la Palabra: Daniela
¡ESPATIFILIO!, ¡No Daniela !
```

3. Laboratorio:

El impuesto más importante, denominado Impuesto Personal de Ingresos (IPI, para abreviar) tenía que pagarse una vez al año y se evaluó utilizando la siguiente regla:

- Si el ingreso del ciudadano no era superior a 85,528 pesos, el impuesto era igual al 18% del ingreso menos 556 pesos y 2 centavos (esta fue la llamada exención fiscal).
- Si el ingreso era superior a esta cantidad, el impuesto era igual a 14,839 pesos y 2 centavos, más el 32% del excedente sobre 85,528 pesos.

Tu tarea es escribir una calculadora de impuestos.

- Debe aceptar un valor de punto flotante: el ingreso.
- A continuación, debe imprimir el impuesto calculado, redondeado a pesos totales. Hay una función llamada `round()` que hará el redondeo por ti, la encontrarás en el código de esqueleto del editor.

```
1 income = float(input("Introduce el ingreso anual: "))
2
3 if income < 85528:
4     tax = income * 0.18 - 556.02
5 else:
6     excedente=income-85528
7     tax=excedente*0.32 +14839
8 if tax<0:
9     tax=0.0
10
11 tax = round(tax, 0)
12 print("El impuesto es:", tax, "pesos")
13
```

```
Console >_
Introduce el ingreso anual: 10000
El impuesto es: 1244.0 pesos
Introduce el ingreso anual: 100000
El impuesto es: 19470.0 pesos
Introduce el ingreso anual: 1000
El impuesto es: 0.0 pesos
Introduce el ingreso anual: -100
El impuesto es: 0.0 pesos
```

4. Laboratorio 4

Desde la introducción del calendario Gregoriano (en 1582), se utiliza la siguiente regla para determinar el tipo de año:

- Si el número del año no es divisible entre cuatro, es un año común.
- De lo contrario, si el número del año no es divisible entre 100, es un año bisiesto.
- De lo contrario, si el número del año no es divisible entre 400, es un año común.
- De lo contrario, es un año bisiesto.

```
1
2 year = int(input("Introduce un año:"))
3 if year >1682:
4     if year%4!=0:
5         if year%400!=0:
6             print("El año es Común")
7         elif year%100!=0:
8             print ("El año es Bisiesto")
9         else:
10            print ("El año es Bisiesto")
11 else:
12     print("No esta dentro del periodo del calendario Gregoriano")
13
```

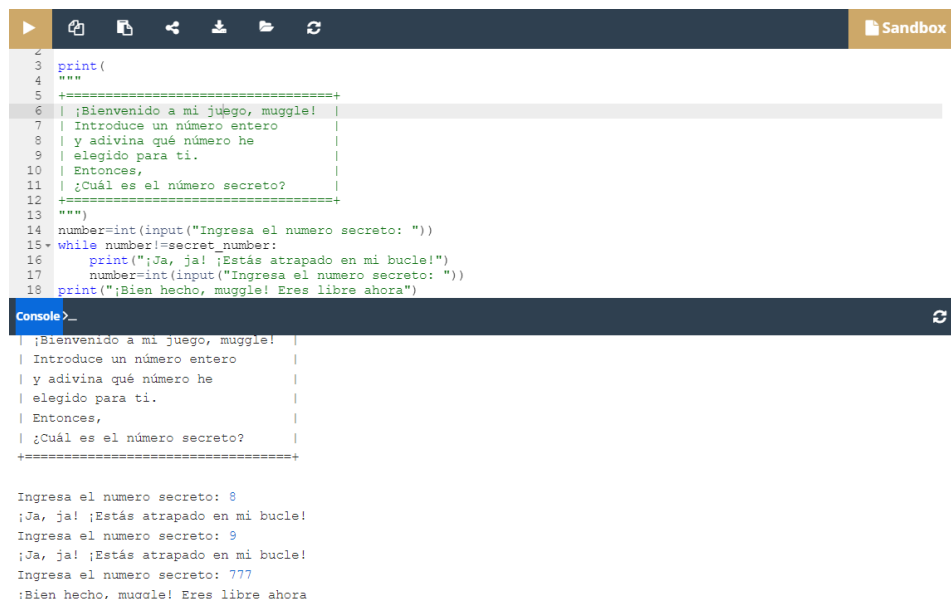
```
Console >_
Introduce un año:2000
El año es Bisiesto
Introduce un año:2015
El año es Común
Introduce un año:1999
El año es Común
Introduce un año:1996
El año es Bisiesto
Introduce un año:1580
El año es Bisiesto
Introduce un año:1580
N esta dentro dle periodo del calendario Gregoriano
```


5. Laboratorio 5

Un mago junior ha elegido un número secreto. Lo ha escondido en una variable llamada `secret_number`. Quiere que todos los que ejecutan su programa jueguen el juego Adivina el número secreto, y adivina qué número ha elegido para ellos. ¡Quiénes no adivinen el número quedarán atrapados en un bucle sin fin para siempre! Desafortunadamente, él no sabe cómo completar el código.

Tu tarea es ayudar al mago a completar el código en el editor de tal manera que el código:

- Pedirá al usuario que ingrese un número entero.
- Utilizará un bucle `while`.
- Comprobará si el número ingresado por el usuario es el mismo que el número escogido por el mago. Si el número elegido por el usuario es diferente al número secreto del mago, el usuario debería ver el mensaje "¡Ja, ja! ¡Estás atrapado en mi bucle!" y se le solicitará que ingrese un número nuevamente. Si el número ingresado por el usuario coincide con el número escogido por el mago, el número debe imprimirse en la pantalla, y el mago debe decir las siguientes palabras: "¡Bien hecho, muggle! Eres libre ahora".



```
2
3 print(
4     """
5     +=====+
6     | ¡Bienvenido a mi juego, muggle!  |
7     | Introduce un número entero     |
8     | y adivina qué número he        |
9     | elegido para ti.                |
10    | Entonces,                        |
11    | ¿Cuál es el número secreto?     |
12    +=====+
13    """
14    number=int(input("Ingresa el numero secreto: "))
15    while number!=secret_number:
16        print("¡Ja, ja! ¡Estás atrapado en mi bucle!")
17        number=int(input("Ingresa el numero secreto: "))
18    print("¡Bien hecho, muggle! Eres libre ahora")

Console>_
| ¡Bienvenido a mi juego, muggle!  |
| Introduce un número entero     |
| y adivina qué número he        |
| elegido para ti.                |
| Entonces,                        |
| ¿Cuál es el número secreto?     |
+=====+

Ingresa el numero secreto: 8
¡Ja, ja! ¡Estás atrapado en mi bucle!
Ingresa el numero secreto: 9
¡Ja, ja! ¡Estás atrapado en mi bucle!
Ingresa el numero secreto: 777
¡Bien hecho, muggle! Eres libre ahora
```

6. Laboratorio 6

Tu tarea es muy simple aquí: escribe un programa que use un bucle `for` para "contar de forma mississippi" hasta cinco. Habiendo contado hasta cinco, el programa debería imprimir en la pantalla el mensaje final "¡Listos o no, ahí voy!"



```
1 import time
2 for i in range(1,6):
3     print(i, " Mississippi")
4     time.sleep(1)
5     i+=1
6 print("¡Listos o no, ahí voy!")
7 # Escribe un bucle for que cuente hasta cinco.
8 # Cuerpo del bucle: imprime el número de iteración del bucle y la palabra "Mississippi".
9 # Cuerpo del bucle - usar: time.sleep (1)
10
11 # Escribe una función de impresión con el mensaje final.

Console>_
1 Mississippi
2 Mississippi
3 Mississippi
4 Mississippi
5 Mississippi
¡Listos o no, ahí voy!
```

7. Laboratorio 7

La instrucción break se implementa para salir/terminar un bucle.

- Diseña un programa que use un bucle while y le pida continuamente al usuario que ingrese una palabra a menos que ingrese "chupacabra" como la palabra de salida secreta, en cuyo caso el mensaje "¡Has dejado el bucle con éxito!" Debe imprimirse en la pantalla y el bucle debe terminar.
 - No imprimas ninguna de las palabras ingresadas por el usuario. Utiliza el concepto de ejecución condicional y la sentencia break.

```
1 palabra=""
2 while palabra!="chupacabra":
3     palabra=input("Ingresa la palabra secreta: ")
4     if palabra=="chupacabra":
5         print("¡Has dejado el bucle con éxito!")
6         break
7
```

Console >_

Ingresa la palabra secreta: mama
Ingresa la palabra secreta: pedito
Ingresa la palabra secreta: danis
Ingresa la palabra secreta: chupacabra
¡Has dejado el bucle con éxito!

8. Laboratorio 8

Tu programa debe:

- Pedir al usuario que ingrese una palabra.
- Utiliza `user_word = user_word.upper()` para convertir la palabra ingresada por el usuario a mayúsculas; hablaremos sobre los llamados métodos de cadena y el `upper()` muy pronto, no te preocupes.
- Utiliza la ejecución condicional y la instrucción continue para "comer" las siguientes vocales A , E , I , O , U de la palabra ingresada.
- Imprime las letras no consumidas en la pantalla, cada una de ellas en una línea separada.

```
1 # Indicar al usuario que ingrese una palabra
2 # y asignarlo a la variable user_word.
3 user_word=input("Ingresa una palabra: ")
4 user_word = user_word.upper()
5 for letter in user_word:
6     # Completa el cuerpo del bucle for.
7     if letter!="A":
8         if letter!="E":
9             if letter!="I":
10                if letter!="O":
11                    if letter!="U":
12                        print(letter)
13 continue
14
```

Sandbox

Console >_

Ingresa una palabra: Gregory
G
R
G
R
Y
Ingresa una palabra: abstemious
B
S
T
M
S
Ingresa una palabra: AEIOU

9. Laboratorio 9

Tu programa debe:

- Pedir al usuario que ingrese una palabra.
- Utilizar `user_word = user_word.upper()` para convertir la palabra ingresada por el usuario a mayúsculas; hablaremos sobre los llamados métodos de cadena y el `upper()` muy pronto, no te preocupes.
- Emplea la ejecución condicional y la instrucción `continue` para "comer" las siguientes vocales A , E , I , O , U de la palabra ingresada.
- Asigne las letras no consumidas a la variable `word_without_vowels` e imprime la variable en la pantalla.

```
1 word_without_vowels = ""
2
3 # Indicar al usuario que ingrese una palabra
4 # y asignarla a la variable user_word.
5 user_word=input("Ingresa una palabra: ")
6 user_word = user_word.upper()
7 for letter in user_word:
8     # Completa el cuerpo del bucle.
9     if(letter!="A"):
10         if(letter!="E"):
11             if(letter!="I"):
12                 if(letter!="O"):
13                     if(letter!="U"):
14                         word_without_vowels+=letter
15 print(word_without_vowels)
16
17 # Imprimir la palabra asignada a word_without_vowels.
18
```

Console >_

Ingresa una palabra: sebastian
SBSTN
Ingresa una palabra: Gregory
GRGRY
Ingresa una palabra: abstemious
BSTMS

10. Laboratorio 10

Tu tarea es escribir un programa que lea la cantidad de bloques que tienen los constructores, y generar la altura de la pirámide que se puede construir utilizando estos bloques.

```
1 blocks = int(input("Ingresa el número de bloques: "))
2
3 height = 0
4 in_layer = 1
5 while in_layer <= blocks:
6     height += 1
7     blocks -= in_layer
8     in_layer += 1
9
10 print("La altura de la pirámide:", height)
11
```

Console >_

Ingresa el número de bloques: 6
La altura de la pirámide: 3
Ingresa el número de bloques: 20
La altura de la pirámide: 5
Ingresa el número de bloques: 1000
La altura de la pirámide: 44
Ingresa el número de bloques: 2
La altura de la pirámide: 1

11. Laboratorio 11

Escribe un programa que lea un número natural y ejecute los pasos anteriores siempre que c_0 sea diferente de 1. También queremos que cuente los pasos necesarios para lograr el objetivo. Tu código también debe mostrar todos los valores intermedios de c_0 .

```
1 c0 = int(input("Ingresa c0: "))
2 if c0 > 1:
3     steps = 0
4     while c0 != 1:
5         if c0 % 2 != 0:
6             cnew = 3 * c0 + 1
7         else:
8             cnew = c0 // 2
9         print(c0)
10        c0 = cnew
11        steps += 1
12    print("pasos =", steps)
13 else:
14    print("Valor de c0 incorrecto")
```

Console>_

```
Ingresa c0: 15
15
46
23
70
35
106
53
160
80
40
20
10
5
16
```

12. Laboratorio

Tu tarea es:

- Escribir una línea de código que solicite al usuario que reemplace el número central en la lista con un número entero ingresado por el usuario (Paso 1).
- Escribir una línea de código que elimine el último elemento de la lista (Paso 2).
- Escribir una línea de código que imprima la longitud de la lista existente (Paso 3).

```
1 hat_list = [1, 2, 3, 4, 5] # Esta es una lista existente de números ocultos en el sombrero.
2
3 # Paso 1: escribe una línea de código que solicite al usuario
4 # reemplazar el número de en medio con un número entero ingresado por el usuario.
5 nuevo=int(input("Ingresa el nuevo valor de la posición central: "))
6 hat_list[2]=nuevo
7 # Paso 2: escribe aquí una línea de código que elimine el último elemento de la lista.
8 del hat_list[-1]
9 # Paso 3: escribe aquí una línea de código que imprima la longitud de la lista existente.
10 print("La longitud de la lista es : ",len(hat_list))
11 print(hat_list)
12
```

Console>_

```
Ingresa el nuevo valor de la posición central: 6
La longitud de la lista es : 4
[1, 2, 6, 4]
```

13. Laboratorio 13

Escribe un programa que refleje estos cambios y le permita practicar con el concepto de listas. Tu tarea es:

- Paso 1: Crea una lista vacía llamada beatles.
- Paso 2: Emplea el método `append()` para agregar los siguientes miembros de la banda a la lista: John Lennon, Paul McCartney y George Harrison.
- Paso 3: Emplea el bucle `for` y el `append()` para pedirle al usuario que agregue los siguientes miembros de la banda a la lista: Stu Sutcliffe, y Pete Best.
- Paso 4: Usa la instrucción `del` para eliminar a Stu Sutcliffe y Pete Best de la lista.
- Paso 5: Usa el método `insert()` para agregar a Ringo Starr al principio de la lista.

```
1 # paso 1:
2 Beatles = []
3 print("Paso 1:", Beatles)
4 # paso 2:
5 Beatles.append("John Lennon")
6 Beatles.append("Paul McCartney")
7 Beatles.append("George Harrison")
8 print("Paso 2:", Beatles)
9 # paso 3:
10 for members in range(2):
11     Beatles.append(input("Nuevo miembro de la banda: "))
12 print("Paso 3:", Beatles)
13 # paso 4:
14 del Beatles[-1]
15 del Beatles[-1]
16 print("Paso 4:", Beatles)
17 # paso 5:
18 Beatles.insert(0, "RingoStarr")
19 print("Paso 5:", Beatles)
20 print("Los Fav:", len(Beatles))
21
```

```
Console>_
Paso 1: []
Paso 2: ['John Lennon', 'Paul McCartney', 'George Harrison']
Nuevo miembro de la banda: stu
Nuevo miembro de la banda: pete
Paso 3: ['John Lennon', 'Paul McCartney', 'George Harrison', 'stu', 'pete']
Paso 4: ['John Lennon', 'Paul McCartney', 'George Harrison']
Paso 5: ['RingoStarr', 'John Lennon', 'Paul McCartney', 'George Harrison']
Los Fav: 4
```

14. Laboratorio 14

Tu tarea es escribir un programa que elimine todas las repeticiones de números de la lista. El objetivo es tener una lista en la que todos los números aparezcan no más de una vez.

```
1 my_list = [1, 2, 4, 4, 1, 4, 2, 6, 2, 9]
2 #
3 # Escribe tu código aquí.
4 #
5 for i in my_list:
6     if(i in my_list):
7         del my_list[i]
8 print("La lista con elementos únicos:")
9 print(my_list)
10
```

```
Console>_
La lista con elementos únicos:
[1, 4, 6, 2, 9]
```