



PE1: Módulo 2: Tipos de datos, variables, operaciones básicas de entrada y salida, operadores básicos

Profesor:

Javier Jesús Torres Yañez

Unidad 1:

Principios de Programación de Redes

Alumno:

Juan Miguel Hernández Beltrán 1223100787

GRUPO:

GIR0642

TSU Infraestructura de Redes Digitales

2.1.1.6 LABORATORIO: La función print()

En este laboratorio, me familiaricé con la función `print()`, que permite imprimir texto en la pantalla. Comencé imprimiendo el mensaje "¡Hola, Mundo!" y luego mi nombre, lo que me ayudó a entender la sintaxis básica. Al eliminar las comillas dobles o un paréntesis de una cadena, se generó un `SyntaxError`, ya que Python no pudo interpretar el texto correctamente.

<pre>1 print("¡Hola, Mundo!") 2 print("Juan Miguel Hernández")</pre>	<pre>1 print(¡Hola, Mundo!) 2 print(Juan Miguel Hernández)</pre>
<pre>Console >_ ¡Hola, Mundo! Juan Miguel Hernández</pre>	<pre>Console >_ ¡Hola, Mundo! Juan Miguel Hernández File "main.py", line 1 print(¡Hola, Mundo!) ^ SyntaxError: invalid character '¡' in string literal</pre>

2.1.1.18 LABORATORIO: La función print()

A través de este ejercicio, aprendí a utilizar la función `print()` en Python y a experimentar con sus parámetros `sep` y `end`

« 2.1.1.18 LABORATORIO: La función print() »

```
SECTION (90%)
1 print("Fundamentos", "Programación", "en", sep="***", end="...")
2 print("Python")
3
```

```
Console >_
Fundamentos***Programación***en...Python
```

2.1.1.19 LABORATORIO: Dando formato a la salida

En este laboratorio, me pidieron experimentar con un código de Python, corrigiendo errores de sintaxis y familiarizándome con la función `print()`. Realicé tareas como minimizar invocaciones de `print()`, duplicar flechas y modificar comillas, lo que me ayudó a entender mejor la sintaxis y el funcionamiento del lenguaje.

[illegible]

[illegible]

En este laboratorio, me pidieron que escribiera una línea de código utilizando la función `print()` y caracteres de nueva línea y escape

LABORATORIO

tiempo Estimado

10 minutos

Nivel de Dificultad

Fácil

Objetivos

- Familiarizarse con la función `print()` y sus capacidades de formato.
- Practicar el codificar cadenas.
- Experimentar con el código de Python.

Escenario

Describe 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.

Salida Esperada

```

"Estoy"
"aprendiendo"
"Python"

```

```

1 print("\nEstoy\n\naprendiendo\n\nPython\n\n")

```

Console >_

```

"Estoy"
"aprendiendo"
"Python"

```

2.4.1.7 LABORATORIO: Variables

En este laboratorio, me pidieron familiarizarme con el almacenamiento y manejo de diferentes tipos de datos. Debía crear variables para Juan, María y Adán, asignando el número de manzanas que tenían. Luego, imprimí las variables separadas por comas y calculé el total de manzanas.

Nivel de Dificultad

Fácil

Objetivos

- Familiarizarse con el concepto de almacenar y trabajar con diferentes tipos de datos en Python.
- Experimentar con el código en Python.

Escenario

A continuación una historia:

Érase una vez en la Tierra de las Manzanas, Juan tenía tres manzanas, María tenía cinco manzanas, y Adán tenía seis manzanas. Todos eran muy felices y vivieron por muchísimo tiempo. Fin de la Historia.

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`.

```

1 # Asignar el número
2 juan = 3
3 maria = 5
4 adan = 6
5
6 # Imprimir
7 print(juan, maria, adan, sep=",")
8
9 # total de manzanas
10 total_manzanas = juan + maria + adan
11
12 # Imprimir el total
13 print("Número Total de Manzanas:", total_manzanas)
14
15 manzanas_multiplicadas = total_manzanas * 2
16 division_manzanas = total_manzanas / 3
17 division_entera = total_manzanas // 4
18
19 print("Manzanas multiplicadas:", manzanas_multiplicadas)
20 print("División de manzanas:", division_manzanas)
21 print("División entera de manzanas:", division_entera)
22

```

Console >_

```

3,5,6
Número Total de Manzanas: 14
Manzanas multiplicadas: 28
División de manzanas: 4.666666666666667
División entera de manzanas: 3

```

2.4.1.9 LABORATORIO: Variables, un sencillo convertidor

En este laboratorio, me pidieron familiarizarme con el concepto de variables y realizar operaciones básicas. Debía completar un programa para convertir millas a kilómetros y viceversa, sin modificar el código existente.

LABORATORIO

Tiempo Estimado

10 minutos

Nivel de Dificultad

Fácil

Objetivos

- Familiarizarse con el concepto de variables y trabajar con ellas.
- Realizar operaciones básicas y conversiones.
- Experimentar con el código de Python.

Escenario

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.

Pon mucha atención a lo que esta ocurriendo dentro de la función `print()`. Analiza como es que se proveen múltiples argumentos para la función, y como es que se muestra el resultado.

Nota que algunos de los argumentos dentro de la función `print()` son cadenas (por ejemplo: "millas son"), y otros son variables (por ejemplo: `miles`).

```
1 kilometers = 12.25
2 miles = 7.38
3
4 miles_to_kilometers = miles * 1.61
5
6 kilometers_to_miles = kilometers / 1.61
7
8 print(miles, "millas son", round(miles_to_kilometers, 2), "kilómetros")
9 print(kilometers, "kilómetros son", round(kilometers_to_miles, 2), "millas")
10
```

Console >_

7.38 millas son 11.88 kilómetros
12.25 kilómetros son 7.61 millas

2.4.1.10 LABORATORIO: Operadores y expresiones

En este laboratorio, se me pidió familiarizarme con números, operadores y operaciones aritméticas. Debía completar un código que evaluara la expresión $3x^3 - 2x^2 + 3x - 1$ y asignar el resultado a la variable `y`. Aprendí a incluir explícitamente el operador de multiplicación y a asegurarme de que `x` fuera un flotante.

Escenario

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:

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

El resultado debe ser asignado a `y`.

Recuerda que la notación algebraica clásica muy seguido omite el operador de multiplicación, aquí se debe de incluir de manera explícita. Nota como se cambia el tipo de dato para asegurarnos de que `x` es del tipo `flotante`.

Mantén tu código limpio y legible, y pruébalo utilizando los datos que han sido proporcionados. No te desanimes por no lograrlo en el primer intento. Se persistente y curioso.

Datos de Prueba

Entrada de Muestra

```
x = 0
x = 1
x = -1
```

Salida Esperada

```
y = -1.0
```

```
1 x = float(input("Introduce el valor de x: "))
2 # Escribe tu código aquí
3 y = 3 * x**3 - 2 * x**2 + 3 * x - 1
4 print("y =", y)
5
6
7
```

Console >_

Introduce el valor de x: 1
y = 3.0

2.5.1.2 LABORATORIO: Comentarios

En este laboratorio, aprendí sobre la importancia de los comentarios. Mejoré el código existente al agregar y quitar comentarios, y cambié nombres de variables para aumentar la claridad y legibilidad

Tiempo Estimado

5 minutos

Nivel de Dificultad

Muy Fácil

Objetivos

- Familiarizarse con el concepto de comentarios en Python.
- Utilizar y no utilizar los comentarios.
- Reemplazar los comentarios con código.
- Experimentar con el código de Python.

Escenario

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 mas legible), además, cambia el nombre de las variables donde consideres que esto mejorará la comprensión del código.

NOTA

Los comentarios son muy importantes. No solo hacen que el programa sea **más fácil de entender**, pero también sirven para **deshabilitar aquellas partes de código que no son necesarias** (por ejemplo, cuando se necesita probar cierta parte del código, e ignorar el resto). Los buenos programadores **describen** cada parte importante del código, y dan **nombres significativos** a variables, debido a que en ocasiones es mucho más sencillo dejar el comentario dentro del código mismo.

```
1 # Este programa calcula los segundos en cierto número de horas determinadas
2
3 horas = 2 # Número de horas
4 segundos_por_hora = 3600 # Número de segundos en una hora
5
6 # Imprimir el número de horas
7 print("Horas:", horas)
8
9 # Calcular e imprimir el número de segundos en las horas especificadas
10 print("Segundos en", horas, "horas:", horas * segundos_por_hora)
11
12 # Final del programa
13 print("Adiós") # Mensaje de despedida
14
```

Console >_

Horas: 2
Segundos en 2 horas: 7200
Adiós
Horas: 2

2.6.1.9 LABORATORIO: Entradas y salidas simples

En este laboratorio, me familiaricé con la entrada y salida de datos al completar un código que evalúa y muestra resultados de operaciones aritméticas básicas.

MODULE (92%) SECTION (59%)

LABORATORIO

Tiempo Estimado

5-10 minutos

Nivel de Dificultad

Fácil

Objetivos

- Familiarizarse con la entrada y salida de datos en Python.
- Evaluar expresiones simples.

Escenario

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

El resultado debe ser mostrado en consola.

Quizá no podrás proteger el código de un usuario que intente dividir entre cero. Por ahora, no hay que preocuparse por ello.

Prueba tu código - ¿Produce los resultados esperados?

No te mostraremos ningún dato de prueba, eso sería demasiado sencillo.

```
1 a = float(input("Introduce un valor para a: "))
2 b = float(input("Introduce un valor para b: "))
3
4 print("La suma de a + b es:", a + b)
5 print("La resta de a - b es:", a - b)
6 print("La multiplicación de a * b es:", a * b)
7
8 if b != 0:
9     print("La división de a / b es:", a / b)
10 else:
11     print("No se puede dividir entre cero.")
12
13 print("\n¡Eso es todo, amigos!")
14
15
16
```

Console >_

Introduce un valor para a: 2
Introduce un valor para b: 3
La suma de a + b es: 5.0
La resta de a - b es: -1.0
La multiplicación de a * b es: 6.0
La división de a / b es: 0.6666666666666666

LABORATORIO

tiempo Estimado

5-20 minutos

Nivel de Dificultad

Medio

Objetivos

- Mejorar la habilidad de implementar números, operadores y operaciones aritméticas en Python.
- Utilizar la función `print()` y sus capacidades de formato.
- Aprender a expresar fenómenos del día a día en términos de un lenguaje de programación.

Escenario

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.

Por ejemplo, si el evento comienza a las **12:17** y dura **59 minutos**, terminará a las **13:16**.

O te preocupes si tu código no es perfecto, está bien si acepta una hora inválida, lo más importante es que el código produzca una salida correcta acorde a la entrada dada.

Prueba el código cuidadosamente. Pista: utilizar el operador `%` puede ser clave para el éxito.

Datos de Prueba

Entrada de muestra:

```
12
17
59
```

Salida esperada: `13:16`

Entrada de muestra:

```
13
```

```
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 # Escribe tu código aquí.
5 total_mins = mins + dura
6 new_hour = hour + total_mins // 60
7 new_mins = total_mins % 60
8
9
10 new_hour = new_hour % 24
11
12 print(f"La hora de finalización es: {new_hour}:{new_mins}")
13
14
```

Console >_

```
Hora de inicio (horas): 12
Minuto de inicio (minutos): 17
Duración del evento (minutos): 59
La hora de finalización es: 13:16
```

PrevNext

Pantalla de Quiz

Your score: **10/10**

100%

Congratulations, you've passed the quiz!

SECTION ANALYSIS

PE1 -- Module 2 Quiz 100%

Retake Test

Review Test

We use cookies to improve our service. By continuing to use the site, you agree to our [privacy](#) and [cookie policies](#). X

Restablecer

Pantalla de Examen Módulo 2

Progress (100%)

Your score: 18/20

90%

Congratulations, you've passed the test!

SECTION ANALYSIS

PE1 -- Module 2 Test 90%

Retake Test

Review Test

We use cookies to improve our service. By continuing to use the site, you agree to our [privacy](#) and [cookie policies](#). X