

# Tarea 01: Curso de programación en python

## Evidencias Aula virtual

- Mateo Molina Moises Estrella

Link Repositorio: [https://github.com/R3LM15/MetodosNumericos\\_MateoMolina](https://github.com/R3LM15/MetodosNumericos_MateoMolina)

Esquema de Curso

Recursos

Buscar esquema del curso

2.1.1 Tu primer programa

2.1.2 La función print()

2.1.3 Argumentos de funciones

2.1.4 Invocación de funciones

2.1.5 LAB Trabajando con la función print()

2.1.6 La función print() y su efecto, argumentos, y valores retornados

2.1.7 Instrucciones

2.1.8 Caracteres de escape y nueva línea en Python

2.1.9 Usando múltiples argumentos

2.1.10 Argumentos posicionales

2.1.11 Argumentos de palabra clave

2.1.12 LAB La función print() y sus argumentos

2.1.13 LAB Dando formato a la salida

2.1.14 RESUMEN DE SECCIÓN

2.1.15 CUESTIONARIO DE SECCIÓN

2.1.5 LAB Trabajando con la función print()

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. Usa comillas dobles alrededor de la cadena.
- Habiendo hecho eso, usa la función `print()` nuevamente, pero esta vez imprime tu nombre.
- Elimina las comillas dobles y ejecuta tu código. Mira la reacción de Python. ¿Qué tipo de error arroja?
- Luego, elimina los paréntesis, vuelve a colocar las comillas dobles y ejecuta tu código nuevamente. ¿Qué tipo de error arroja esta vez?
- Experimenta todo lo que puedas. Cambia las comillas dobles a comillas simples, usa múltiples funciones `print()` en la misma línea, y luego en diferentes líneas. Mira qué pasa.

```
1 print("hola mundo, soy Mateo Molina")
```

Console

hola mundo, soy Mateo Molina

Esquema de Curso

Recursos

Buscar esquema del curso

PE1: Módulo 1. Introducción a Python y a la Programación Informática

80%

PE1: Módulo 2. Tipos de datos, Variables, Operaciones Básicas de Entrada y Salida, Operadores Básicos

5%

2.1. Sección 1 – El Programa "¡Hola, Mundo!"

7 / 15

2.1.1 Tu primer programa

2.1.2 La función print()

2.1.3 Argumentos de funciones

2.1.4 Invocación de funciones

2.1.5 LAB Trabajando con la función print()

2.1.6 La función print() y su efecto, argumentos, y valores retornados

2.1.7 Instrucciones

util cuando el código contiene construcciones complejas).

Vamos a expandir el código un poco, puedes verlo en el editor. Ejecútalo y observa lo que aparece en la consola.

```
1 print("La Witsi Witsi Araña subió a su telaraña.")
2 print("Vino la lluvia y se la llevó.")
3
```

Console

La Witsi Witsi Araña subió a su telaraña.
Vino la lluvia y se la llevó.

Tu consola de Python ahora debería verse así:

Esquema de Curso

Recursos

Buscar esquema del curso

2.1.7 Instrucciones

2.1.8 Caracteres de escape y nueva línea en Python

2.1.9 Usando múltiples argumentos

2.1.10 Argumentos posicionales

2.1.11 Argumentos de palabra clave

2.1.12 LAB La función print() y sus argumentos

2.1.13 LAB Dando formato a la salida

2.1.14 RESUMEN DE SECCIÓN

2.1.15 CUESTIONARIO DE SECCIÓN

2.2. Sección 2 – Literales de Python

0 / 8

2.3. Sección 3 – Operadores – herramientas de manipulación de datos

0 / 5

2.4. Sección 4 – Variables

0 / 12

2.5. Sección 5 – Comentarios

0 / 5

2.1.15 CUESTIONARIO DE SECCIÓN

Pregunta 3: ¿Cuál de las siguientes `print()` invocaciones de función generarán un `SyntaxError`?

```
1 print('Greg's book.')
2 print("Greg's book.")
3 print('Greg\'s book.')
4 print("Greg's book.")
5 print('Greg's book.')
6
```

Ocultar

La línea 5 generará un `SyntaxError`, porque el símbolo `'` en la cadena `Greg's book.` requiere un carácter de escape.

Esquema de Curso

Recursos

Buscar esquema del curso

su procesamiento, Operaciones Lógicas y de Bit a Bit

PE1: Módulo 4. Funciones, Tuplas, Diccionarios, Excepciones y Procesamiento de Datos

Fundamentos de Python 1 (PE1) Examen Final del Curso

33%

Proyecto Final

Prueba Final

Encuesta de Fin de Curso

Examen de certificación PCEP – Certified Entry-Level Python Programmer

100%

Proyecto Final

```
20 if not ok:
21     print("Movimiento erróneo, ingrésalo nuevamente") # no, no lo es. ingrésalo nuevamente
22     continue
23 move = int(move) - 1 # número de la celda, del 0 al 8
24 row = move // 3 # fila de la celda
25 col = move % 3 # columna de la celda
26 sign = board[row][col] # marca el cuadro elegido
27 ok = sign not in ['0','X']
28 if not ok: # está ocupado, ingresa una posición nuevamente
29     print("¡Cuadro ocupado, ingresa nuevamente!")
30     continue
31 board[row][col] = '0' # colocar '0' al cuadro seleccionado
32
33 def make_list_of_free_fields(board):
34     free_fields = []
35
36     for i in range(3):
37         for j in range(3):
38             if board[i][j] != 'X' and board[i][j] != '0':
39                 free_fields.append((i,j))
40     return free_fields
41
42 def choose_place_to_move(board, free_fields, player_turn):
43     ok = False
44     while not ok:
45         row = int(input("Ingresa la fila: "))
46         col = int(input("Ingresa la columna: "))
47         sign = board[row][col]
48         if sign == 'X' or sign == '0':
49             print("¡Cuadro ocupado, ingresa nuevamente!")
50             continue
51         board[row][col] = player_turn
52         return (row,col)
53
54 def make_move(board, row, col, player_turn):
55     board[row][col] = player_turn
56
57 def print_board(board):
58     for i in range(3):
59         for j in range(3):
60             print(board[i][j], end=" ")
61         print()
62
63 def play_game(board):
64     player_turn = 'X'
65     while True:
66         print_board(board)
67         row, col = choose_place_to_move(board, free_fields, player_turn)
68         make_move(board, row, col, player_turn)
69         if row == 0 and col == 0:
70             print("¡Ganó X!")
71             return
72         if row == 0 and col == 0:
73             print("¡Ganó O!")
74             return
75         player_turn = 'O' if player_turn == 'X' else 'X'
76
77 def main():
78     board = [['0','X','0'], ['0','0','0'], ['0','0','0']]
79     play_game(board)
80
81 if __name__ == '__main__':
82     main()
```

Console

Ingresa tu movimiento: 1

es erroneo

dará como salida None

## Pregunta 8

¿Qué valor se asignará a la variable `x`?

1 z = 0

2 y = 10

3 x = y < z and z > y or y < z and z < y

4

False

1

True

0

Esquema de Curso

Recursos

Buscar esquema del curso

PE1: Módulo 4. Funciones, Tuplas, Diccionarios, Excepciones y Procesamiento de Datos

Fundamentos de Python 1 (PE1) Examen Final del Curso

66%

Proyecto Final

Prueba Final

Encuesta de Fin de Curso

Examen de certificación PCEP – Certified Entry-Level Python Programmer

100%

Prueba Final

89%

Has obtenido 89%.

Felicidades, has aprobado la evaluación.

NetAcad

Logotipo NetAcad

Explorar

Búsqueda de cursos, artículos y recursos

Estudiante

Mi aprendizaje

ES

?

mateo Estudiante

Perfil > Insignias y certificados >

Este certificado se otorga a

mateo molina

por completar con éxito

Fundamentos de Python 1

ofrecido por Networking Academy

a través del programa Cisco Networking Academy.

Lynn Bloomer

Lynn Bloomer

Directora

Cisco Networking Academy

19 May 2025

Fecha de Finalización

Completado por

mateo molina

Fecha de emisión

May 18, 2025

Total de horas

30

ACERCA DEL CURSO

**Fundamentos de Python 1**

Aprenda los conceptos fundamentales de la programación informática y comience a desarrollar habilidades de codificación con el lenguaje de programación Python.

SKILLS

Fundamentos de Python 1

Descargar