



## Lección 01

### Print01

```
print01.py X
parcial1 > leccion01 > print01.py > ...
1 # Imprimir variables simples, una cadena, un numero entero
2
3 nombre = "Mario"
4 edad = 43
5 pi = 3.1416
6 vivo = True
7 frutas = ['Manzana', 'Fresa', 'Papaya', 'Uva']
8
9 print("Nombre: ", nombre)
10 print("Edad: ", edad)
11 print("PI: ", pi)
12 print("Vivo: ", vivo)
13 print("Frutas: ", frutas)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print01.py
Nombre: Mario
Edad: 43
PI: 3.1416
Vivo: True
Frutas: ['Manzana', 'Fresa', 'Papaya', 'Uva']
```

### Print02

```
print02.py M
parcial1 > leccion01 > print02.py > ...
1 # Intercalando cadenas y variables para imprimir
2
3 nombre = "Pineda"
4 edad = 43
5
6 print("Hola, yo me llamo", nombre, "y tengo", edad, "años.")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print02.py
Hola, yo me llamo Pineda y tengo 19 años.
```



## Print03

```
print03.py X
parcial1 > leccion01 > print03.py
1 # Separando dos lineas
2
3 print("Esto es una linea \n y esto es otra linea.")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print03.py
Esto es una linea
y esto es otra linea.
```

## Print04

```
print04.py X
parcial1 > leccion01 > print04.py
1 # Aquí vamos a imprimir una linea
2 print ("El mejor ron del mundo es Zacapa XO")
3
4 # Aquí vamos a substituir el caracter de fin de linea por ::
5 print ("Sin embargo este ron tambien es muy bueno", end= "::")
6 print("Diplomatico")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print04.py
El mejor ron del mundo es Zacapa XO
Sin embargo este ron tambien es muy bueno::Diplomatico
```



## Print05

```
print05.py x
parcial1 > leccion01 > print05.py
1 #Concatenando cadenas para imprimir
2
3 print('Amor' + ' & ' + 'Paz')
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print05.py
Amor & Paz
```

## Print06

```
print06.py x
parcial1 > leccion01 > print06.py > ...
1 # Dando formato a una cadena con variables
2
3 import math
4 a,b=3,4
5 c = math.sqrt(a**2 + b**2)
6 print('Cateto a: {} y cateto b: {} igual a hipotenuza c: {}'.format(a,b, c))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print06.py
Cateto a: 3 y cateto b: 4 igual a hipotenuza c: 5.0
```

## Print07

```
print07.py x
parcial1 > leccion01 > print07.py > ...
1 # Leyendo una cadena desde el teclado
2 # Imprimiendo el dato introducido y su tipo
3
4 n = input('Dame un numero: ')
5
6 print('El numero tecleado es:', n)
7
8 print('y el tipo de dato es: ', type(n))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print07.py
Dame un numero: 11
El numero tecleado es: 11
y el tipo de dato es: <class 'str'>
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print07.py
Dame un numero: a
El numero tecleado es: a
y el tipo de dato es: <class 'str'>
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print07.py
Dame un numero: pineda
El numero tecleado es: pineda
y el tipo de dato es: <class 'str'>
```



## Print08

```
print08.py M X
parcial1 > leccion01 > print08.py > ...
4
5 count_seconds = 5
6 for i in reversed(range(count_seconds + 1)):
7     if i > 0:
8         #Intenta primero con esta linea
9         print(i, end='>')          You, 1 second ago * Uncommitted changes
10        #Despues comenta la linea anterior y descomenta la siguiente
11        print(i, end='>>>', flush = True)
12        time.sleep(1)
13    else:
14        print('Inicio')
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1\_2024\parcial1\leccion01> py print08.py  
5->4->3->2->1->Inicio

Al imprimirse sin especificar el `flush = True`, se contabilizarán los 5 segundos antes de imprimir el Inicio.

```
print08.py M X
parcial1 > leccion01 > print08.py > ...
6 for i in reversed(range(count_seconds + 1)):
7     if i > 0:
8         #Intenta primero con esta linea
9         #print(i, end='>')
10        #Despues comenta la linea anterior y descomenta la siguiente
11        print(i, end='>', flush = True)          You, 1 second ago * Uncommite
12        time.sleep(1)
13    else:
14        print('Inicio')
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1\_2024\parcial1\leccion01> py print08.py  
5->4->3->2->1->Inicio

En este caso, al estar activa la opción `flush = True`, se irán imprimiendo con forme pasen los segundos.



## Print09

```
print09.py X
parcial1 > leccion01 > print09.py > ...
  augustoramirez, 2 weeks ago | 1 author (augustoramirez)
1 # Separando datos con un separador especial
2
3 d=27
4 m=9
5 a=2024
6 print(d,m,a,sep="-")  augustoramirez, 2 weeks ago • Leccion 1

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print09.py
27-9-2024
```

## Print10

```
print10.py X
parcial1 > leccion01 > print10.py
  augustoramirez, 2 weeks ago | 1 author (augustoramirez)
1 # Escribiendo un archivo desde el print  augustoramirez, 2 weeks ago • L
2
3 print('Bienvenidos a PCD 2024.!!', file=open('pcd.txt', 'w'))
4

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print10.py
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> 
```

```
pcd.txt U X
parcial1 > leccion01 > pcd.txt
1 Bienvenidos a PCD 2024.!!
2
```



## Print11

```
print11.py M X
parcial1 > leccion01 > print11.py > ...
1 # FILE CONTENTS
2
3 val = 'PINEDA'
4 print(f"Cuando cuentas {val}, cuenta cuántos {val} cuentas, porque si no cuentas
  sabrás cuántos {val} cuentas tú.")
5
6 name = 'PINEDA'
7 age = 19
8 print(f"Hello, My name is {name} and I'm {age} years old.")
9

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print11.py
Cuando cuentas PINEDA, cuenta cuántos PINEDA cuentas, porque si no cuentas cuántos PINEDA cues
A cuentas tú.
Hello, My name is PINEDA and I'm 19 years old.
```

## Print12

```
print12.py X
parcial1 > leccion01 > print12.py > ...
augustoramirez, 2 weeks ago | 1 author (augustoramirez)
1 # Como imprimir la fecha de hoy
2
3 import datetime
4
5 hoy = datetime.datetime.today()
6 print(f"{hoy: %B %d, %Y}")
7 print(f"{hoy: %m %d, %Y}")
8

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print12.py
October 08, 2024
10 08, 2024
```

(Existen muchos otros formatos % para imprimir fechas)



## Print13

```
print13.py M X
parcial1 > leccion01 > print13.py
You, 4 seconds ago | 2 authors (augustoramirez and one other)
1 # Como imprimir comillas
2
3 print(f'Imprimir comillas')
4
5 print(f"""Imprime "dobles" comillas""")
6
7 print(f'Imprime comillas 'simples'.')
8 #usando caracter de escape \ You, 1 second ago • Uncommitted
9 print('Imprime comillas \'simples\'.')
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print13.py
'Imprimir comillas'
Imprime "dobles" comillas
Imprime comillas 'simples'.
Imprime comillas 'simples'.
```

## Print14

```
print14.py X
parcial1 > leccion01 > print14.py > ...
augustoramirez, 2 weeks ago | 1 author (augustoramirez)
1 # Operaciones con f-string augustoramirez, 2 weeks ago • Leccion
2 examen = 60
3 libro = 10
4 practicas = 20
5
6 print(f"Calificacion total: {examen + libro + practicas} de 100")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print14.py
Calificacion total: 90 de 100
```



## Print15

```
print15.py M X
parcial1 > leccion01 > print15.py > ...
You, 6 seconds ago | 2 authors (augustoramirez and one other)
1 # Dando formato a pi
2
3 pi = 3.14159265358979323846264338327950288419716939937510
4 formateado = f"{pi:.4f}"
5 formateado2 = f"{pi:.10f}"
6 print(formateado, formateado2) You, 1 second ago • Uncommitted changes
7

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print15.p
3.1416 3.1415926536
```

## Print16

```
print16.py M X
parcial1 > leccion01 > print16.py > ...
1 # Imprimiendo una lista
2 palabras = ["Hello", "World", "!"]
3
4 print("_ _".join(palabras)) You, 1 second ago • Uncommitted changes

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print16
Hello_ _World_ _!
```





## Print17

```
print17.py M X
parcial1 > leccion01 > print17.py > ...

3 lista = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
4
5 print(*lista)
6 print("imprime la lista con todo y corchetes y comas separadoras",lista)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print17.py
1 2 3 4 5 6 7 8 9 10
imprime la lista con todo y corchetes y comas separadoras [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01>
```

## Print18

```
print18.py X
parcial1 > leccion01 > print18.py
augustoramirez, 2 weeks ago | 1 author (augustoramirez)
1 # Correo electronico augustoramirez, 2 weeks ago • Leccion 1
2
3 print("augusto.ramirez", end='@')
4 print("gmail.com")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print18.py
augusto.ramirez@gmail.com
```



## Print19

```
print19.py X
parcial1 > leccion01 > print19.py
augustoramirez, 2 weeks ago | 1 author (augustoramirez)
1 # Experimentos con el fin de linea
2
3 print('C','D','M', sep='', end='')
4 print('X')
5
6 # Despues de un print sin el delimitador end, se restablece el fin de linea
7 print('27','09','2024', sep='-', end='\n')
8
9 augustoramirez, 2 weeks ago • Leccion 1
10 # Otro fin de linea
11 print('VERDE','BLANCO','ROJO', sep='|', end='@')
12 print('mexico')
13

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print19.py
CDMX
27-09-2024
VERDE|BLANCO|ROJO@mexico
```

## Print20

```
print20.py X
parcial1 > leccion01 > print20.py > ...
augustoramirez, 2 weeks ago | 1 author (augustoramirez)
1 # Mas complicado augustoramirez, 2 weeks ago • Leccion 1
2
3 nombre = "Augusto"
4 edad = 42
5 print("Mi nombre es", nombre, "y tengo", edad, "años.", end=" ")
6 print("Mucho gusto!")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print20.py
Mi nombre es Augusto y tengo 42 años. Mucho gusto!
```



## Print21

```
print21.py X
parcial1 > leccion01 > print21.py > ...
augustoramirez, 2 weeks ago | 1 author (augustoramirez)
1 # Formato con el caracter (%) augustoramirez, 2 weeks ago • Leccion 1
2 pi = 3.14159265358979323846264338327950288419716939937510
3
4 print("Estudiantes : %2d, Edad promedio : %5.2f" % (35, 019.333))
5
6 print("Hombres : %3d, Mujeres : %2d" % (20, 15))
7
8 print("Octal: %7.3o" % (25)) # Imprimir en Octal
9
10 print("Pi: %10.4E" % (pi)) # Notacion exponencial
11

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print21.py
Estudiantes : 35, Edad promedio : 19.33
Hombres : 20, Mujeres : 15
Octal: 031
Pi: 3.1416E+00
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01>
```

## Print22

```
print22.py X
parcial1 > leccion01 > print22.py
augustoramirez, 2 weeks ago | 1 author (augustoramirez)
1 # Diferentes formas de imprimir con formato augustoramirez, 2 weeks ago
2
3 print('I love {}. "{}!{}".format('this game!', 'Just do it!'))
4
5 print('{0} and {1}'.format('I love this game!', 'Just do it!'))
6
7 print('{1} and {0}'.format('I love this game!', 'Just do it!'))
8
9 print(f"I love {'this game'}! and \"{'Just do it'}!\")
10
11 print(f"{'I love this game!'} and {'Just do it!'}")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print22.py
I love this game!. "Just do it!!"
I love this game! and Just do it!
Just do it! and I love this game!
I love this game! and "Just do it!"
I love this game! and Just do it!
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01>
```



## Print23

```
print23.py x
parcial1 > leccion01 > print23.py
augustoramirez, 2 weeks ago | 1 author (augustoramirez)
1 # argumentos por posicion y por nombre augustoramirez, 2 weeks ago • Leccion 1
2 print('El mejor equipo {0}, el segundo {1}, y el tercero {otro}.')
3     .format('CELTICS', 'NUGGETS', otro = 'BULLS'))
4
5 # con format, posiciones y formato
6 print("Primera posicion, entero de un digito:>>{0:3d}<<, segunda posicion flotante:>>{1:8.2f}<<".
7     format(12, 00.546))
8
9 # Cambiando posiciones
10 print("segundo argumento flotante:>>{1:8.2f}<< primer argumento entero:>>{0:3d}<<, ".
11     format(12, 00.546))
12
13 # Argumentos por nombre
14 print("a: {a:5d}, Portbal: {b:8.2f}").
15     format(a = 1234, b = 19.123456789))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print23.py
El mejor equipo CELTICS, el segundo NUGGETS, y el tercero BULLS.
Primera posicion, entero de un digito:>> 12<<, segunda posicion flotante:>> 0.55<<
segundo argumento flotante:>> 0.55<< primer argumento entero:>> 12<<,
a: 1234, Portbal: 19.12
```

## Print24

```
print24.py x
parcial1 > leccion01 > print24.py > [?] texto
augustoramirez, 2 weeks ago | 1 author (augustoramirez)
1 texto = "BOSTON Celtics" augustoramirez, 2 weeks ago • Leccion 1
2
3 # Centrado
4 print("Texto centrado y lleno con #: ")
5 print(texto.center(40, '#'))
6
7 # Alineacion a la izquierda
8 print("Alineado a la izquierda : ")
9 print(texto.ljust(40, '-'))
10
11 # Alineacion a la derecha
12 print("Alineado a la derecha : ")
13 print(texto.rjust(40, '*'))
14

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion01> py print24.py
Texto centrado y lleno con #:
#####BOSTON Celtics#####
Alineado a la izquierda :
BOSTON Celtics-----
Alineado a la derecha :
*****BOSTON Celtics
```