



UNIVERSIDAD  
IBEROAMERICANA  
CIUDAD DE MÉXICO ®

## **P1. Fundamentos de Python**

Alumna: Daniela Mendez Ramirez

Numero de Cuenta: 258331-9

### **Aplicaciones de Redes y Laboratorio**

Profesor: Omar Vázquez González

Fecha de Entrega: 19 de enero de 2025

## **1. Resumen General sobre datos de programación.**

### **1.1.1 ¿Cómo funciona un programa de computadora?**

En pocas palabras lo que se necesita para que una computadora pueda realizar procesos, son los programas, estos nos permiten diseñar distintas operaciones para que la computadora las procese de manera rápida y eficiente.

### **1.1.2 Lenguajes naturales vs lenguajes de programación**

Lenguaje máquina, que suele un poco más rudimentario, es el lenguaje natural de las computadoras.

Un conjunto completo de comandos conocidos se llama lista de instrucciones, a veces abreviada IL (por sus siglas en inglés)

Lenguaje natural, se refiere a los que nosotros los humanos usamos día a día, aquellos que se van transformando constantemente, haciendo uso de nuevas palabras y olvidando otras,

Lenguaje de alto nivel, este se refiere a la forma más "sencilla" con la que podemos interactuar con la computadora, usando comandos más sencillos y entendibles para nosotros.

### **1.1.3 ¿Qué compone a un lenguaje?**

Alfabeto: Símbolos utilizados para formar palabras

Léxico: También conocido como diccionario

Sintaxis: Conjunto de reglas que valida si una oración es válida, es decir, si es gramáticamente correcta

Semántica: Si una frase tiene sentido

### **1.1.4 Lenguaje máquina vs. lenguaje de alto nivel**

El IL (Lista de Instrucciones) es, de hecho, el alfabeto de un lenguaje máquina. Siendo así el conjunto más simple para dar instrucciones a la computadora.

Sin embargo, es necesario comunicarnos de una forma más sencilla, tanto para la computadora como para nosotros, es por eso que existe el lenguaje de alto nivel, pues usan símbolos y palabras convencionales para los humanos, ofreciendo comandos mucho más complejos que las IL.

Un programa escrito en un lenguaje de programación de alto nivel se denomina código fuente (en contraste con el código máquina ejecutado por las computadoras). Del mismo modo, el archivo que contiene el código fuente se denomina archivo fuente.

### 1.1.5 Compilación vs. Interpretación

**Compilación:** El programa fuente se traduce al menos una vez, sin embargo, se tiene que hacer esta "traducción" cada que se hace una modificación en el código fuente. Esto convirtiéndose ahora en un código máquina. A estos programas se les llama compilador o traductor.

**Interpretación:** Se puede traducir el código fuente cada vez que se va a ejecutar. El programa que realiza esta transformación se llama intérprete, ya que interpreta el código cada vez que se va a ejecutar. La diferencia radica en que el código fuente tal cual, no se puede distribuir, pues quien desee hacer uso de él también requerirá un intérprete.

	Compilación	Interpretación
Ventajas	<ul style="list-style-type: none"><li>✓ la ejecución del código traducido suele ser más rápida;</li><li>✓ solo el usuario debe tener el compilador; el usuario final puede usar el código sin él;</li><li>✓ el código traducido se almacena usando lenguaje máquina; como es muy difícil de entender, es probable que tus propios inventos y trucos de programación sigan siendo tu secreto.</li></ul>	<ul style="list-style-type: none"><li>✓ puedes ejecutar el código tan pronto como lo completes; no hay fases adicionales de traducción;</li><li>✓ el código se almacena usando un lenguaje de programación, no un lenguaje máquina; esto significa que se puede ejecutar en computadoras que usan diferentes lenguajes máquina; no compila tu código por separado para cada arquitectura diferente.</li></ul>
Desventajas	<ul style="list-style-type: none"><li>✗ la compilación en sí puede ser un proceso que consume mucho tiempo; es posible que no puedas ejecutar su código inmediatamente después de realizar una modificación;</li><li>✗ debes tener tantos compiladores como plataformas de hardware donde desees que se ejecute tu código.</li></ul>	<ul style="list-style-type: none"><li>✗ no esperes que la interpretación acelere tu código a alta velocidad: tu código compartirá el poder de la computadora con el intérprete, por lo que no puede ser realmente rápido;</li><li>✗ tanto tu como el usuario final deben tener el intérprete para ejecutar tu código.</li></ul>

**Nota: Python es un lenguaje interpretado**

## 2. Comandos básicos

Imprimir en consola:

```
print("Hola Mundo, soy Daniela Mendez")
```

Entrada de usuario:

```
radio = float(input("Ingresa el radio del circulo: "))
```

Comentarios:

```
# Esto es un comentario
```

## 3. Declaración de variables

```
x = 10          # Entero  
y = 3.14        # Flotante  
texto = "Hola Dany" # Cadena
```

## 4. Tipos de datos

- a. Números: int, float, complex.
- b. Texto: str.
- c. Booleanos: True, False.

```
numero = 24  
texto = "Hola Dany"  
lista = [1, 2, 3]
```

## 5. Edición de archivos \*.py

- a. Guarda tu archivo, ya sea en un Notepad o un editor de texto, y guárdalo como .py

## 6. Ejemplos de la condición if.

```
# Ejemplo de un if: Determina si un número es positivo, negativo o cero
numero = int(input("Introduce un número: "))
if numero > 0:
    print("El número es positivo")
elif numero < 0:
    print("El número es negativo")
else:
    print("El número es cero")
```

#### 7. Ejemplo de cómo funciona un for

```
# Ejemplo del for: Imprime los números del 1 al 10
for i in range(1, 11):
    print(i)
```

#### 8. Ejemplo de cómo funciona un while

```
ovejas = 0
despierto = "si"

while (despierto == "si"):
    despierto = input("Estas Despierto?\n")
    ovejas += 1

print("Numero total de ovejas contadas: ", ovejas)
```

#### 9. Ejemplo de una función

```
# Define una función que sume dos números
def suma(a, b):
    return a + b

# Llama a la función
resultado = suma(5, 7)
print(f"La suma es: {resultado}")
```

## 10. Resolver los siguientes problemas:

### a. Metro

```
print('Mayoria de edad')
edad = eval(input('Ingresa tu edad:'))

if edad >= 65:
    print('No pagas metro guapi')
elif edad <= 5:
    print('Estas muy baby, no pagas')
else:
    print('Son $5 pesos')

print('Viva la vida al maximo')
```

### b. Informático con Insomnio

```
ovejas = 0
despierto = "si"

while (despierto == "si"):
    despierto = input("Estas Despierto?\n")
    ovejas += 1

print("Numero total de ovejas contadas: ", ovejas)
```

### c. Area de un circulo

```
import math
def area_circulo(r):
    return (math.pi * r)

radio = float(input("Ingresa el radio del circulo: "))
print("El area del circulo es: ", area_circulo(radio))
```

### d. Area de un Triangulo

```
def area_triangulo(b, h):
    return ((b*h)/2)

base = float(input("Ingresa la base: "))
altura = float(input("Ingresa la altura: "))
print("El area del triangulo es: ", area_triangulo(base, altura))
```

**e. Grados Celsius a Fahrenheit y viceversa (Una función de cada uno)**

```
def cel2faren(c):
    return (((9/5)*c) + 32)

def faren2cel(f):
    return (((5/9)*c) - 32)
print('Menu')
print('a. De Celcius a Farenheit')
print('b. De Farenheit a Celcius')

op = (input('Que deseas convertir?\n'))

if(op == 'a'):
    cel = float(input('Ingresa los grados Celsius: '))
    print('De grados Celsius a Farenheit tu resultado es: ', cel2faren(cel))
else:
    faren = float(input('Ingresa los grados Faren: '))
    print('De grados Farenheit a Celsius tu resultado es: ', faren2cel(faren))
```

**f. Promedio de n números**

```
cant_num = int(input('¿Cuántos números deseas evaluar?: '))

for i in range (0, cant_num):
    num = float(input(f'Ingresa el numero {i + 1} a evaluar:'))
    num += num

print('Promedio es igual: ', (num/cant_num))
```

**g. Factorial**

```
def factorial(n):
    if n == 1 or n==0:
        return 1
    else:
        return n * factorial(n-1)

x = 10
print('El resultado es: ', factorial(x))
```

## h. Serie Fibonacci

```
def fibonacci(n):  
    if(n>2):  
        return fibonacci(n-1) + fibonacci(n-2)  
    elif(n==2):  
        return 1  
    elif(n==1):  
        return 1  
    elif(n==0):  
        return 0  
  
fibo = int(input("Ingresa que numero de fibonacci deseas ver: "))  
  
for i in range(0, fibo):  
    print('Resultado es: ', fibonacci(i))
```

## Referencias.

APA: Cisco Networking Academy. (n.d.). *Python Essentials*. Cisco Systems, Inc. Recuperado de <https://www.netacad.com/content/pe1/1.0/index.html?xAPILaunchKey=ea2e6672-740b-4642-ad0c-dcd0e39c1849&xAPILaunchService=https://www.netacad.com/adl/content/&lang=es-XL&moduleNumber=1#/courses/content/m1/id/9945fdd2a1a14b3ba13d033532266fab>

## Adjuntos.

Diagrama de Flujo, sobre el meme “Insomnio”.



