**Metodología de formacion:**

* Se entrega ejemplos de clase
* Bitácora
* Videos
* Presentación usada en la formacion.

Se socializa la guía, las actividades que se van ha estar haciendo

Se les da la fechas de entrega de fase que es la ultima semana de noviembre inicia 27

**Métodos:**

* es una acción que se da en un sistema
* son un conjunto de líneas de código que realiza una acción , operación o evaluación
* pequeño programa que creamos con el fin de una necesidad de no repetir código en nuestro programa.

Método es parte de la programación orientada objetos y en si es una función, el cual son de dos tipos.

**Que son los paradigmas de programación.**

Son estilos y formas de programar, donde le demos una estética al codigo



<https://www.youtube.com/watch?v=IVg3-X8PkY4>



# ejemplos de funciones incorporadas

print("hola")

# convertir  en numero lo digitado por teclado

variable=int(input(" dijite numeros"))

# convertir en titulo

texto=input(" dijite para titulo")

texto=texto.title()

print(texto)

# convertir mayuscula

texto=input(" dijite  texto como sea para ser convertido en mayuscula")

texto=texto.upper()

print(texto)

"""esta es un ejemplo de creacion de funcion

con la palabra reservada def

"""

def ejemplo():

    print("hola mundo")

# esta parte es donde se ejecuta el codigo

ejemplo()

def suma():

    total=6+8+9+3+24+6+784

    print(f" el total es {total}")

suma()

#funciones que retorna un valor enciando los valores

def restar(num1, num2):

    total=num1-num2

    return total

variable=restar(num1=9,num2=6)

print(f" el resultado del valor restado es {variable}")

<https://docs.python.org/es/dev/tutorial/controlflow.html>

# funcion

""" este ejemplo de funciones es para simular la sentesia casos o lo que llaman swiches, consiste que se envia un valor y se espera que devuelva una operacion o ensaje"""

# menu de comidas

print("presiones 1 para pizza")

print("presiones 2 hamburgueza")

print("presiones 3 Chuzo con ensalada")

print("presiones 4 perro caliente ")

print("presiones 5 gratinado de queso")

print("presiones 1 carne asada")

def menu\_comida():

    opcion=int(input("dijite valor del menu"))

    match opcion:

        case 1: print("presiones 100$ para pizza")

        break

        case 2:print("costo 200$ hamburgueza")

        case 3:print("costo 300$ Chuzo con ensalada")

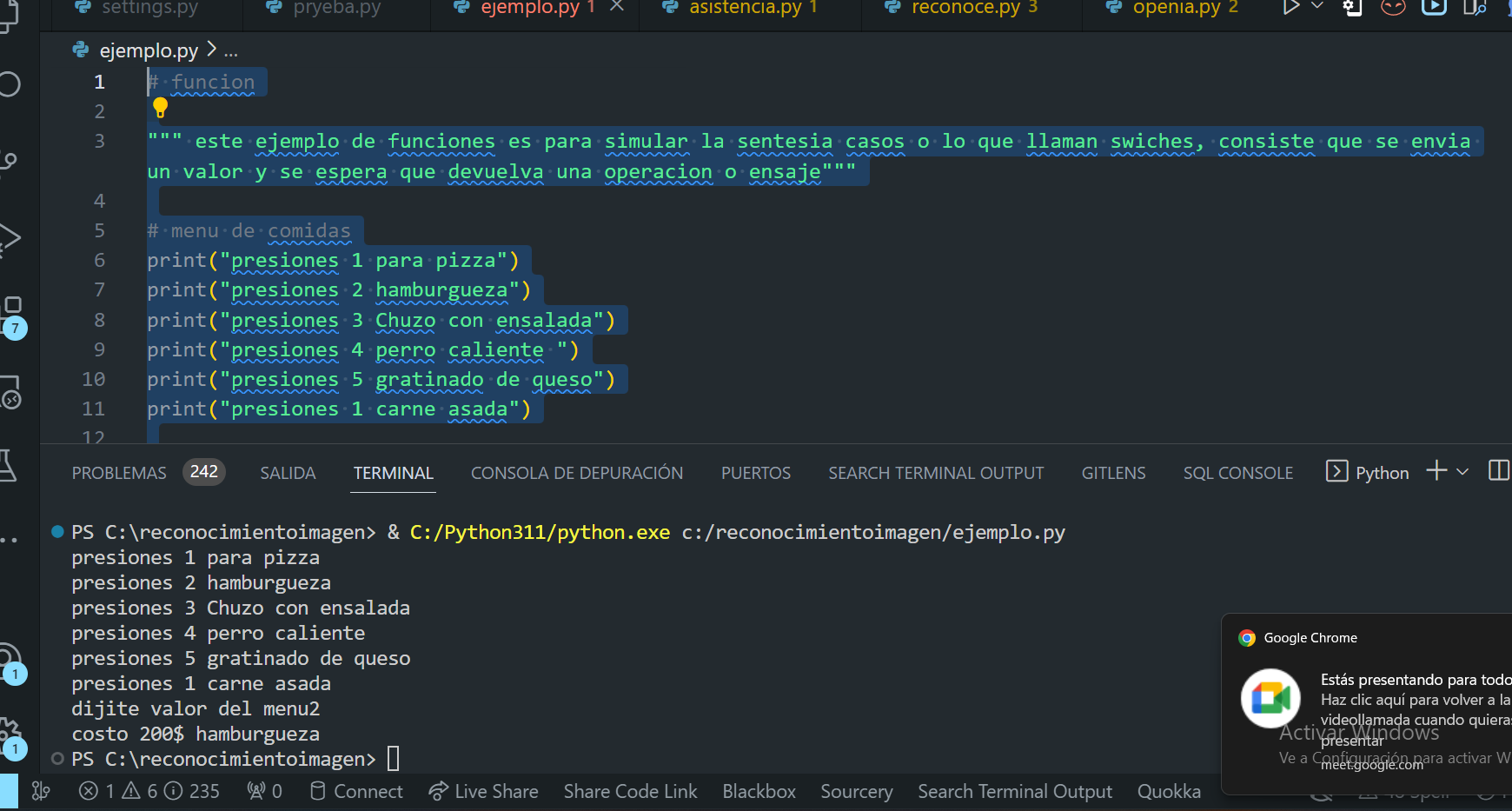
        case 4:print("costo 400$ perro caliente ")

        case 5:print("costo 500$ gratinado de queso")

        case 6:print("costo 100$ carne asada")

        case \_:print("opcion dijitada no coincide")

menu\_comida()

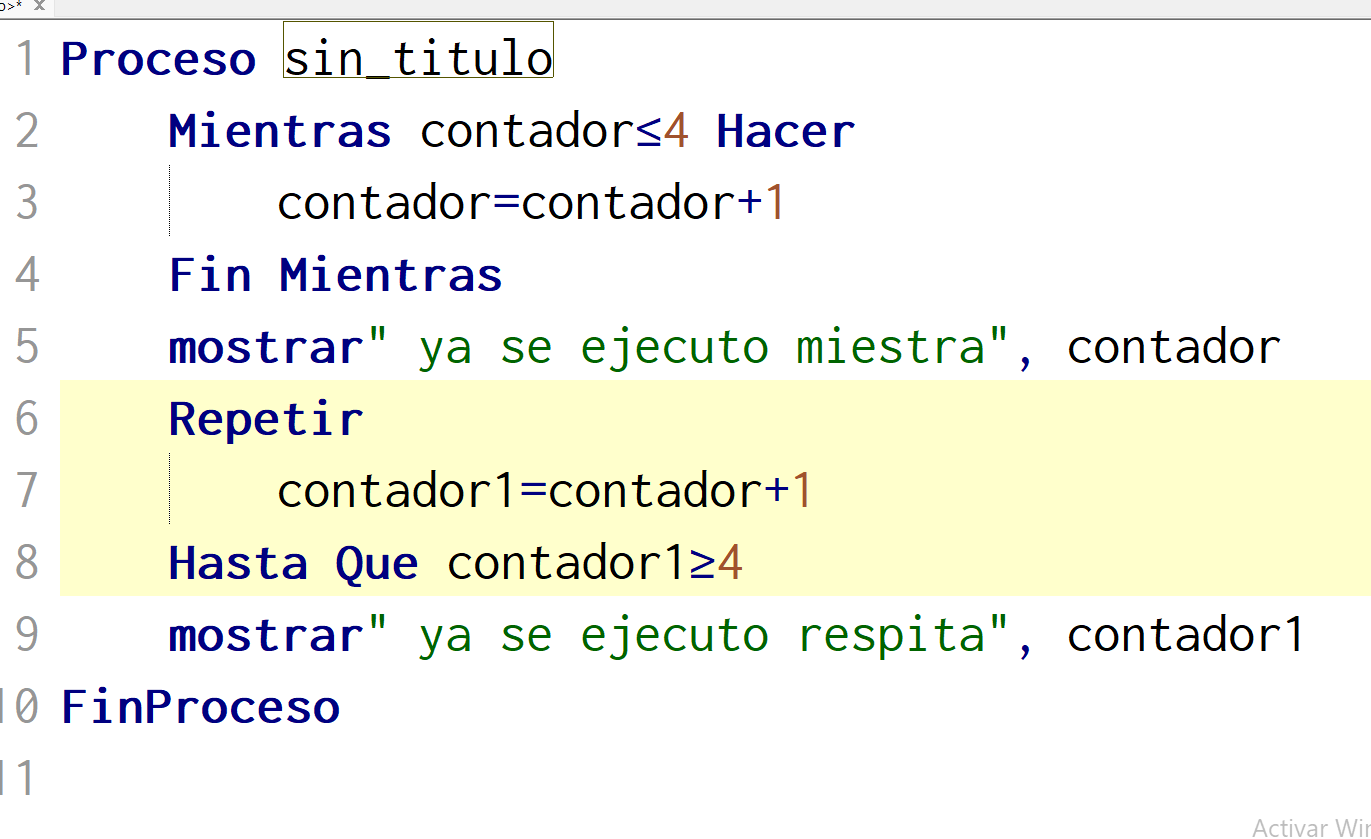
aclaraciones y dudas de aprendices

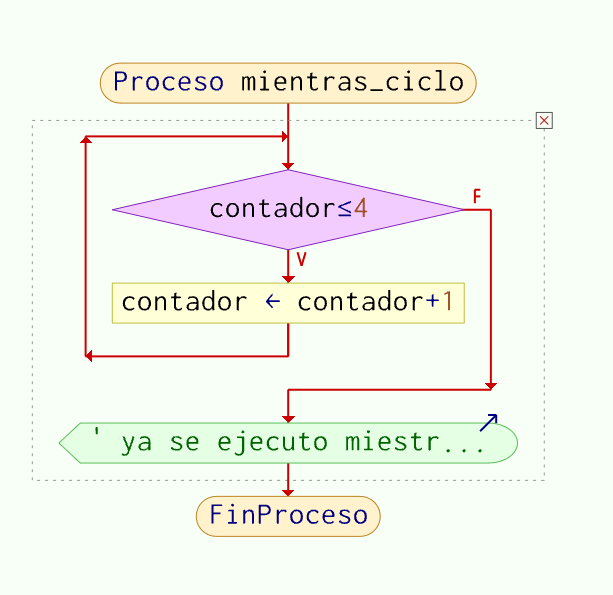
Python en las versiones inferiores de 3.10 no existía los SW o los llamados casos, este se manejafa con if variable= Valor:

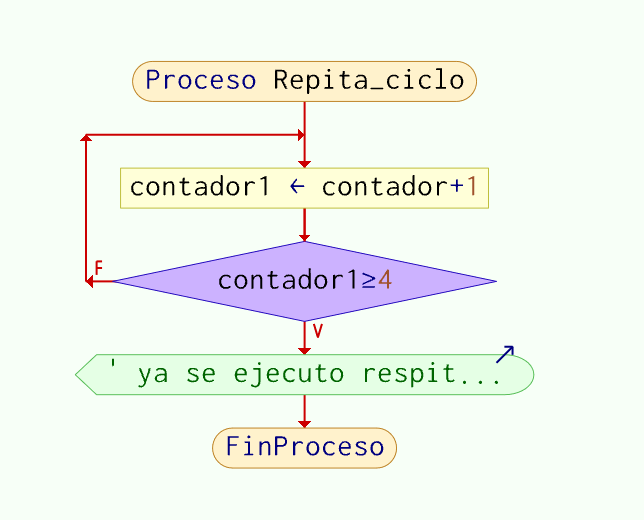
    Elif valor de validación

Python no tiene ciclo repita por que la comunicad no lo consideró importante: Python no tiene un ciclo Repita porque los ciclos while y for pueden utilizarse para realizar las mismas tareas.

**Centinelas=** es un valor que se asigna con el fin que interrumpa o pare un ciclo en ejecución, también una función recursiva, puede ser de tipo numérico, Booleana, cadena, carácter.

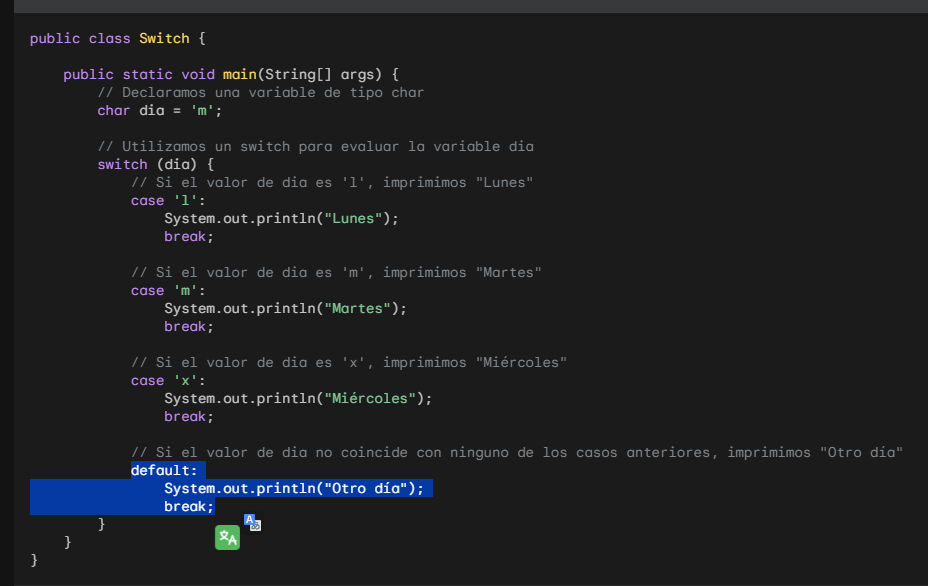
Ejemplo 



****

**Instrucción BREAK en Python**

Bueno muchachos este comando como tal en los casos no se utilizan en Python pero sí en Java y es con el fin de qué cada caso no se esté ejecutando simplemente se ejecuta el que cumpla la condición el cual se ingresó ejemplo.

****

**Para que lo activen y en las subtitulaciones en ingles o español**

Hola a todos, este es un mensaje automático para informarles que mi extensión Tactiq (<https://tactiq.io/r/transcribing>) está transcribiendo esta reunión para que pueda brindarles toda mi atención.

**Mejor redacción:**

**Break se utiliza en los ciclos for y while para detener la ejecución del ciclo cuando se cumple una condición.**

**Explicación:**

**El break es una instrucción que se utiliza para detener la ejecución de un ciclo. Se puede utilizar en cualquier momento, pero es más común utilizarlo en combinación con condicionales.**

**Ejemplos:**

#Breck como se una

"""

este ejemplo lo que hace es un ciclo para es decir un for que se repite 10 veces, pero cuadando llegue a 8 se sale o termina la ejecucion

"""

numero=1

for numero in range(1,11):

    print("El número es: ",numero)

    if numero ==8:

        print("llego a 8 y se terminara la ejecucion")

        break

"""

este ejemplo lo que hace es un ciclo para es decir un for que se repite 10 veces, pero cuadando llegue a 8 se sale o termina la ejecucion

"""

numero=1

while numero<=10:

    print("El número es: ",numero)

    if numero ==8:

        print("llego a 8 y se terminara la ejecucion")

        break

    numero+=1

**Java:**

Java

int numero = 1;

while (numero <= 10) {

System.out.println("El número es: " + numero);

if (numero == 8) {

System.out.println("Llegó a 8 y se terminará la ejecución");

break;

}

numero++;

}

Utiliza el código con precaución. [Más información](https://bard.google.com/faq#coding)

content\_copy

**PHP:**

PHP

$numero = 1;

while ($numero <= 10) {

echo "El número es: " . $numero . PHP\_EOL;

if ($numero == 8) {

echo "Llegó a 8 y se terminará la ejecución" . PHP\_EOL;

break;

}

$numero++;

}

Utiliza el código con precaución. [Más información](https://bard.google.com/faq#coding)

content\_copy

**Kotlin:**

Kotlin

var numero = 1

while (numero <= 10) {

println("El número es: $numero")

if (numero == 8) {

println("Llegó a 8 y se terminará la ejecución")

break

}

numero++

}

Utiliza el código con precaución. [Más información](https://bard.google.com/faq#coding)

content\_copy

**C#:**

C#

int numero = 1;

while (numero <= 10) {

Console.WriteLine("El número es: " + numero);

if (numero == 8) {

Console.WriteLine("Llegó a 8 y se terminará la ejecución");

break;

}

numero++;

}

Utiliza el código con precaución. [Más información](https://bard.google.com/faq#coding)

content\_copy

**Dart:**

Dart

int numero = 1;

while (numero <= 10) {

print("El número es: ${numero}");

if (numero == 8) {

print("Llegó a 8 y se terminará la ejecución");

break;

}

numero++;

}

Utiliza el código con precaución. [Más información](https://bard.google.com/faq#coding)

content\_copy

**Go:**

Go

numero := 1

for numero <= 10 {

fmt.Println("El número es:", numero)

if numero == 8 {

fmt.Println("Llegó a 8 y se terminará la ejecución")

break

}

numero++

}

Utiliza el código con precaución. [Más información](https://bard.google.com/faq#coding)

content\_copy

**Función Lambda:**

Bueno muchachos entonces las funciones lambda en Python también conocidas como funciones anónimas son funciones pequeñas y anónimas que se pueden definir en una sola línea se utilizan comúnmente para crear funciones simples rápidas sin la necesidad de dar un hombre formal la sintaxis es muy fácil es muy básica y vamos a hacer algunos ejemplos:

#ejemplos basicos  de funcion lambda

"""lambda argumento=exprexion"""

resta=lambda numero1,numero2:numero1-numero2

resultado=resta(numero1=7,numero2=9)

print("el resultado es ",resultado)

#ejemplos basicos  de funcion lambda

"""lambda argumento=exprexion"""

# multiplicacion 3 valores

multiplicacion=lambda x,y,z:x\*y\*z

resultado=multiplicacion(x=2,y=3,z=6)

print("resultado de la multimplicacion es   ",resultado)

import math

coseno=lambda x:math.cos(x)

resultado=coseno(x=80)

print("valor del angulo es",resultado)

**Conclusión de la sección de hoy**

En la sección de hoy, pudimos observar las diferencias entre los ciclos, ya que Python no maneja el ciclo repita. Además, vimos parte de los paradigmas de programación, tanto estructurado como funcional. Se definieron los tipos de métodos, con el fin de tener material adelantado para cuando estemos viendo el tema de programación orientada a objetos. Adicionalmente, vimos el comando break, con el fin de que lo sepan interpretar y cuando lo deben de utilizar. Además, nos dimos cuenta de que a partir de la versión de Python 3.10 empezó a salir la función match, que es prácticamente el switch en Python, el cual en versiones inferiores no estaba.

**Conclusión**

Damos como conclusión que las versiones inferiores de 3.x no son muy compatibles en los códigos, por lo cual es muy bueno estar verificando la documentación de Python para hacer certeros con la codificación. También concluimos que no nos podemos negar a decir que no sabemos un lenguaje de programación al momento de ser citados a prácticas, porque vimos homologaciones de diferentes lenguajes de programación, como PHP, C#, Java, JavaScript, y concluimos que lo que necesitamos es buena lógica, dado que tenemos herramientas de inteligencia artificial que nos han facilitado el proceso de aprendizaje en otros lenguajes de programación.