

Menu Principal

```
import ex1
import ex2
import ex3
import ex4
import ex5
import ex6

def main():
    op = 0
    while op < 1 or op > 7:
        print("""
            1. Mastermind
            2. Llista de compra
            3. Joc
            4. POO
            5. Scrapping
            6. Servei web
            7. Sortir
            """)

        try:
            op = int(input("Tria una opció: "))
            if op < 1 or op > 7:
                print("Ópcio no vàlida")
        except ValueError:
            print("Per favor, introdueix un número vàlid.")

    return op

op = 0
while op != 7:
    op = main()
    if op == 1:
        ex1.pex1()
    elif op == 2:
        ex2.pex2()
    elif op == 3:
        ex3.pex3()
    elif op == 4:
        ex4.pex4()
    elif op == 5:
        ex5.pex5()
    elif op == 6:
        ex6.pex6()
```

```

elif op == 7:
    print("Gracies per utilitzar el joc")

main()

```

Per executar el programa, tindras que executar-lo posant main().

Funcionament:

```

pygame 2.1.2 (SDL 2.0.20, Python 3.10.12)
Hello from the pygame community. https://www.pygame.org

1. Mastermind
2. Llista de compra
3. Joc
4. P00
5. Scrapping
6. Servei web
7. Sortir

Tria una opció:

```

Ex1

```

import random

# Función para generar una lista aleatoria de 3 números entre 1 y 9
def gllistaaleatoris():
    l=[]
    for i in range(3):
        l.append(random.randint(1,9)) # Genera un número aleatorio
entre 1 y 9
    return l

# Función para que el usuario ingrese una lista de 3 números
def llegir_llista():
    l=[]
    for e in range(3):
        a = int(input("Introudueixi el número: ")) # Solicita al
usuario un número
        l.append(a) # Agrega el número a la lista
    return l

```

```

# Función para comparar la lista aleatoria con la lista ingresada por
el usuario
def comparar(l, m):
    a=[0,0,0,0]
    for i in range(3):
        if l[i]==m[i]:
            a[i]=10 # Marca con 10 si el número y la posición son
correctos
    if a[0]==10 and a[1]==10 and a[2]==10:
        print("Enhorabona, ho has encertat tot ") # Mensaje si todas
las posiciones son correctas
        return 0 # Retorna 0 para indicar que el juego ha terminado
    for i in range(3):
        if a[i]==0:
            if m[i] in l:
                a[i]=5 # Marca con 5 si el número está en la lista pero
no en la posición correcta
            for i in range(3):
                if a[i]==10:
                    print("L'element {} és correcte".format(m[i])) # Mensaje
para números en la posición correcta
                elif a[i]==5:
                    print("L'element {} existeix, però no està al seu
lloc".format(m[i])) # Mensaje para números en la lista pero posición
incorrecta
                else:
                    print("L'element {} no existeix".format(m[i])) # Mensaje
para números que no están en la lista

# Función principal que maneja el juego de adivinanza
def pexl():
    op = 1
    while op != 0:
        l = gllistaaleatoris() # Genera la lista aleatoria
        m = llegir_llista() # El usuario ingresa la lista de números
        op = comparar(l, m) # Compara las listas y actualiza op según
el resultado

        if op == 0:
            break

        continuar = input("Vols seguir jugant? (s/n): ").lower()
        if continuar != 's':

```

```

        break
    print("Gràcies per jugar!")

# Entrada principal del programa
if __name__ == "__main__":
    pex1()

```

Llibreries importades; import random

Per executar tindras que posar "if __name__ == "__main__":pex1()"per executar-lo dintre del menu principal, si el volem executar desde dins el exercici, posarem tot sol pex1()

Ex2

```

import json
import os

# Funció per guardar el diccionari en un arxiu JSON abans de sortir
def acabar(diccionari, nom="lista_compra.json"):
    with open(nom, "w") as f:
        json.dump(diccionari, f, indent=4) # Guarda el diccionari en
l'arxiu JSON amb format indentat
    print("Dades guardades en lista_compra.json") # Missatge de
confirmació

# Funció per crear un nou arxiu de llista de compres
def crear_fitxer():
    with open("lista_compra.json", "w") as f:
        print("Fitxer Lista de la Compra creat \n") # Imprimeix un
missatge indicant que l'arxiu ha estat creat

# Funció per inserir un nou element en el diccionari (producte i
quantitat)
def inserir_element(diccionari):
    item = input("Nom del producte: ")
    Unitat = input("Unitats: ")
    diccionari[item] = Unitat # Afegeix el producte i la seva quantitat
al diccionari

```

```

print(f"Producte {item} afegit.") # Missatge de confirmació

# Funció per llistar els elements del diccionari (productes i
quantitats)
def llista_fitxer(diccionari):
    if not diccionari:
        print("No hi ha productes en la llista de la compra.") #
Missatge si el diccionari està buit
    else:
        for item, cantidad in diccionari.items():
            print(f"Producte: {item}, Quantitat: {cantidad}") #
Imprimeix cada producte i la seva quantitat

# Funció per eliminar un element del diccionari pel nom del producte
def eliminar_element(diccionari):
    item = input("Nom del producte a eliminar: ")
    if item in diccionari:
        del diccionari[item] # Elimina el producte del diccionari si
existeix
        print(f"Producte {item} eliminat.") # Missatge de confirmació
    else:
        print("Producte no trobat.") # Missatge si el producte no està
en el diccionari

# Funció per modificar la quantitat d'un producte en el diccionari
def modificar_element(diccionari):
    item = input("Nom del producte a modificar: ")
    if item in diccionari:
        cantidad = input("Nova quantitat: ")
        diccionari[item] = cantidad # Actualitza la quantitat del
producte en el diccionari
        print(f"Producte {item} modificat.") # Missatge de confirmació
    else:
        print("Producte no trobat.")

# Funció per mostrar el menú d'opcions i obtenir la elecció de l'usuari
def menu_fitxer():
    print("""
Menú Llista de la Compra:
1. Afegir producte a la llista
2. Llistar productes de la llista
3. Eliminar un producte de la llista
4. Modificar un producte de la llista
5. Sortir

```

```

    """

    op = int(input("Eligeix una opció: "))
    return op

# Funció principal que maneja la lògica del programa
def pex2():
    nom = "lista_compra.json"
    diccionari = {} # Diccionari que emmagatzemarà els productes i les
seves quantitats

    # Si existeix l'arxiu, el llegim i guardem en el diccionari
    if os.path.isfile(nom):
        with open(nom, "r") as f:
            diccionari = json.load(f)
    else:
        crear_fitxer()

    op = 0
    while op != 5:
        op = menu_fitxer()
        if op == 1:
            inserir_element(diccionari)
        elif op == 2:
            llista_fitxer(diccionari)
        elif op == 3:
            eliminar_element(diccionari)
        elif op == 4:
            modificar_element(diccionari)
        elif op == 5:
            acabar(diccionari, nom)
            print("Surtint del programa...")
        else:
            print("Opció no vàlida. Si us plau, tria novament.")

# Executar la funció principal quan el programa es cridi directament
if __name__ == "__main__":
    pex2()

```

Llibreries importades: import json i import os

Per executar tindras que posar "if __name__ == "__main__":pex2()"per executar-lo dintre del menu principal, si el volem executar desde dins el exercici, posarem tot sol pex2()

Funcionament desde programa principal:

```
Tria una opció: 2
Fitxer Lista de la Compra creat

Menú Llista de la Compra:
1. Afegir producte a la llista
2. Llistar productes de la llista
3. Eliminar un producte de la llista
4. Modificar un producte de la llista
5. Sortir

Eligeix una opció: █
```

Ex3

```
import pygame
import random

# Función principal del juego del dinosaurio
def pex3():
    pygame.init() # Inicializa Pygame

    # Configuración de la pantalla
    SCREEN_WIDTH = 871
    SCREEN_HEIGHT = 489
    screen = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT))
    pygame.display.set_caption("Dinosaur Game") #

    # Cargar imágenes
    ground = pygame.image.load("ground.png").convert()
    DinoS = pygame.image.load("DinoStart.png")
    cactus_img = pygame.image.load("Cactus1.png")

    # Configuración del dinosaurio
    dino_rect = DinoS.get_rect() # Obtiene el rectángulo del dinosaurio
    dino_rect.topleft = (10, SCREEN_HEIGHT - dino_rect.height - 50)
    jumping = False
```

```

jump_velocity = 5
gravity = 1

# Configuración de los cactus
cactus_list = [] # Lista para almacenar los cactus en pantalla
cactus_timer = 0 # Temporizador para controlar la generación de
nuevos cactus
cactus_interval = 60 # Intervalo de tiempo en frames para generar
nuevos cactus

clock = pygame.time.Clock()
done = False # Variable para controlar el fin del juego

# Bucle principal del juego
while not done:
    for event in pygame.event.get(): # Maneja los eventos de Pygame
        if event.type == pygame.QUIT:
            done = True # Sale del bucle si se cierra la ventana

    # Manejo del salto del dinosaurio
    keys = pygame.key.get_pressed() # Obtiene el estado de todas
las teclas
    if keys[pygame.K_SPACE] and not jumping: # Si se presiona
espacio y no está saltando
        jumping = True
        jump_velocity = 20

    if jumping:
        dino_rect.y -= jump_velocity # Actualiza la posición
vertical del dinosaurio
        jump_velocity -= gravity # Aplica la gravedad al salto
        if dino_rect.y >= SCREEN_HEIGHT - dino_rect.height - 50: #
Si toca el suelo
            dino_rect.y = SCREEN_HEIGHT - dino_rect.height - 50
            jumping = False # Termina el salto

    # Generar nuevos cactus
    cactus_timer += 1
    if cactus_timer >= cactus_interval:
        cactus_timer = 0
        new_cactus_rect = cactus_img.get_rect() # Rectángulo del
nuevo cactus

```



```

        new_cactus_rect.topleft = (SCREEN_WIDTH, SCREEN_HEIGHT -
new_cactus_rect.height - 50)

        # Generar velocidad aleatoria para el cactus
        cactus_speed = random.randint(10, 15) # Velocidades entre
10 y 15

        cactus_list.append((new_cactus_rect, cactus_speed)) #
Agrega el nuevo cactus a la lista

        # Mover los cactus y verificar colisiones
        for cactus_rect, cactus_speed in cactus_list:
            cactus_rect.x -= cactus_speed # Mueve el cactus hacia la
izquierda

        # Detección de colisiones
        if dino_rect.colliderect(cactus_rect):
            print("Game over, se ha acabat el joc")
            done = True # Termina el juego si hay colisión

        # Eliminar los cactus que han salido de la pantalla
        cactus_list = [(cactus_rect, cactus_speed) for cactus_rect,
cactus_speed in cactus_list if cactus_rect.x + cactus_rect.width > 0]

        # Dibujar todo en la pantalla
        screen.blit(ground, (0, 0)) # Dibuja el suelo
        screen.blit(DinoS, dino_rect) # Dibuja el dinosaurio
        for cactus_rect, _ in cactus_list:
            screen.blit(cactus_img, cactus_rect) # Dibuja los cactus

        pygame.display.flip() # Actualiza la pantalla
        clock.tick(30)

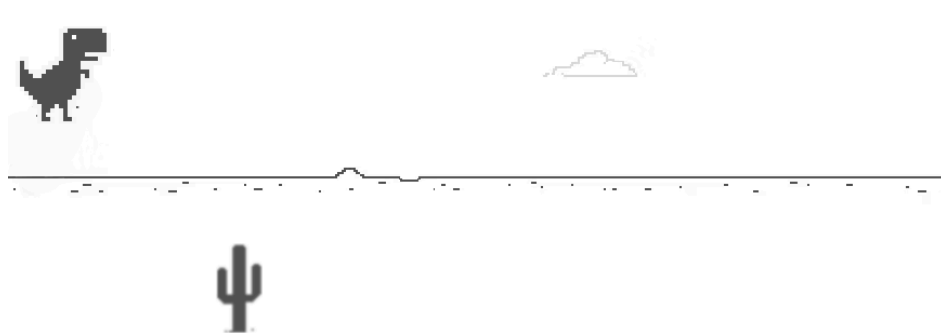
    pygame.quit() # Cierra Pygame

# Esta función se llama desde el programa principal para iniciar el
juego
if __name__ == "__main__":
    pex3()

```

Llibreries importades: import pygame i import radom

Funcionament desde el programa principal:



Ex4

```
# Definición de la clase Animal con métodos básicos
class Animal:
    def __init__(self, nombre, edad):
        self.nombre = nombre
        self.edad = edad

    def hacer_sonido(self):
        pass

    def __str__(self):
        return f"{self.nombre} ({self.__class__.__name__}), edad: {self.edad}"

# Clase Perro que hereda de Animal
class Perro(Animal):
    def hacer_sonido(self):
        return "Guau!" # Poner el ladrido en hacer_sonido específico para Perro

# Clase Gato que hereda de Animal
class Gato(Animal):
    def hacer_sonido(self):
        return "¡Miau!" # Poner el Miau en hacer_sonido específico para Gato

# Clase Vaca que hereda de Animal
class Vaca(Animal):
    def hacer_sonido(self):
        return "¡Muuu!" # Poner el MUU para hacer_sonido específico para Vaca
```

```

# Función principal que crea instancias de diferentes animales y los
utiliza
def pex4():
    perro = Perro("Tobi", 6) # El Perro tiene el nombre "Tobi" y edad
5
    gato = Gato("Michi", 4) # El Gato tiene el nombre "Michi" y edad
4
    vaca = Vaca("Persiana", 5) # La Vaca tiene el nombre "Persiana" y
edad 5

    animales = [perro, gato, vaca] # Lista que los objetos

    for animal in animales:
        if isinstance(animal, Animal): # Verifica si el objeto es de
la clase Animal
            print(f"{animal} dice: {animal.hacer_sonido()}") # Imprime
la representación del animal y el sonido que hace
        else:
            print(f"Objeto {animal} no es una instancia de Animal") #
Mensaje si el objeto no pertenece a Animal

# Programa principal que llama a la función pex4 para ejecutar el
ejemplo
if __name__ == "__main__":
    pex4()

```

Per executar tindras que posar “if __name__ == “__main__”:pex4()”per executar-lo dintre del menu principal, si el volem executar desde dins el exercici, posarem tot sol pex4()

Funcionament amb en programa principal

5. Scrapping
6. Servei web
7. Sortir

Tria una opció: 4

Tobi (Perro), edad: 6 dice: Guau!

Michi (Gato), edad: 4 dice: ¡Miau!

Persiana (Vaca), edad: 5 dice: ¡Muuu!

1. Mastermind
2. Llista de compra
3. Joc

Ln 7, Col 1

Ex6

```
import http.server
import socketserver
import time
from threading import Thread

# Función para iniciar un servidor web en un puerto específico durante
una duración determinada
def start_server(port=5000, duration=5*60):
    # Configurar el manejador de las solicitudes HTTP usando
SimpleHTTPRequestHandler
    handler = http.server.SimpleHTTPRequestHandler

    # Crear el objeto del servidor TCP
    httpd = socketserver.TCPServer("", port), handler)

    print(f"Servidor web funcionando en el puerto {port}")

    try:
        # Iniciar el servidor separado para permitir la espera
        server_thread = Thread(target=httpd.serve_forever)
```

```

server_thread.start()

# Esperar durante el tiempo especificado
time.sleep(duration)
except KeyboardInterrupt:
    pass
finally:
    # Detener el servidor después de que haya pasado el tiempo de
duración
    httpd.shutdown()
    server_thread.join()
    print(f"Servidor web detenido después de {duration} segundos")

# Función pex6 que se puede utilizar para iniciar el servidor web con
parámetros específicos
def pex6():
    start_server(port=5000, duration=5*60) # Llama a start_server con
puerto 5000 y duración de 5 minutos

# Punto de entrada principal del programa
if __name__ == "__main__":
    start_server() # Llama a start_server con los valores
predeterminados

```

Llibreries importades: import http.server, import socketserver, import time i from threading
importy thread

Per executar tindras que posar “if __name__ == “__main__”:pex6()”per executar-lo dintre del menu principal, si el volem executar desde dins el exercici, posarem tot sol start_server()

```

Tria una opció: 6
Servidor web funcionando en el puerto 5000

```

Directory listing for /

- [__pycache__/](#)
 - [Cactus1.png](#)
 - [DinoStart.png](#)
 - [ex1.py](#)
 - [ex2.py](#)
 - [ex3.py](#)
 - [ex4.py](#)
 - [ex5.py](#)
 - [ex6.py](#)
 - [ground.png](#)
 - [principal.py](#)
-