

MENU

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

def menu():
    op = 0
    while op<1 or op>7:
        print("""
            Menú principal:
            1. Mastermind
            2. Llista de compra
            3. Joc
            4. POO
            5. Scrapping
            6. Servidor Web
            7. Sortir
            """)

        op = int(input("Eligeix una opció: "))
        if op<1 or op>8:
            print("Opció no vàlida!")
        else:
            return op

op = 0
while op!=7:
    op = menu()
    match(op):
        case 1:
            ex1.pex1()
        case 2:
            ex2.pex2()
        case 3:
            ex3.pex3()
        case 4:
            ex4.pex4()
        case 5:
            ex5.pex5()
```

```
case 6:
    ex6.pex6()
case 7:
    print("Gràcies per emprar aquest programa!")
```

Per executar aquest programa individualment, és tan simple com executar-lo sense més

```
PS D:\Projecte> & C:/Users/usuario/AppData/Local/Microsoft/WindowsApps/python3.11.exe d:/Projecte/menu.py
pygame 2.5.2 (SDL 2.28.3, Python 3.11.9)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

```
Menú principal:
1. Mastermind
2. Llista de compra
3. Joc
4. POO
5. Scrapping
6. Servidor Web
7. Sortir
```

```
Eligeix una opció:
```

EX1

```
import random

#eligeix 3 nombre aleatoris entre l'1 i el 9
def gllistaaleatoris():
    l = []
    for i in range(3):
        l.append(random.randint(1,9))
    return l

#et demana introduir 3 nombres
def llegir_nombres():
    l = []
    for i in range(3):
        l.append(int(input("Introdueix un nombre: ")))
    return l

#compara tots els nombres amb la resposta correcta
def comparar(l,m):
    a = [0, 0, 0]
    for i in range(3):
        if l[i] == m[i]:
            a[i] = 10
    if a[0] == 10 and a[1] == 10 and a[2] == 10:
        print("Enhorabona, ho has encertat tot!")
        return 0

    for i in range(3):
        if a[i] == 0:
            if m[i] in l:
                a[i] = 5
    for i in range(3):
        if a[i] == 10:
            #et diu els correctes, els incorrectes o els que no estan
            #en la seva posició
            print("L'element {} no és correcte".format(m[i]))
        elif a[i] == 5:
            print("L'element {} existeix, però no està al seu
lloc".format(m[i]))
        else:
            print("L'element {} no existex".format(m[i]))

#bucle de les funcions
```

```
def pex1():  
    op = 1  
    l = gllistaaleatoris()  
    while op!=0:  
        m = llegir_nombres()  
        op = comparar(l,m)
```

Per executar aquest programa individualment, és tan simple com després de la funció pex1() escriure a una linea apart fora de la funció= pex1()

```
PS D:\Projecte> & C:/Users/usuario/AppData/Local/Microsoft/WindowsApps/python3.11.exe d:/Projecte/ex1.py  
Introdueix un nombre: 1  
Introdueix un nombre: 3  
Introdueix un nombre: 2  
L'element 1 no existex  
L'element 3 no és correcte  
L'element 2 no existex  
Introdueix un nombre: █
```

EX2

```
import json
import os

def menu():
    op = -1
    while op < 0 or op > 6:
        print("""
            Llista de compra:
            0. Crear llista
            1. Afegir element
            2. Editar element
            3. Eliminar element
            4. Llegir llista
            5. Guardar abans de sortir
            6. Sortir
            """)
        op = int(input("Introdueix una opció: "))
        if op < 0 or op > 6:
            print("Opció no vàlida!")
        else:
            return op

#Si el fitxer no està creat el crea
def crear_fitxer(nom):
    with open(nom, "w") as f:
        print("El fitxer {} s'ha creat".format(nom))

#Carrega el diccionari al fitxer
def carregar_fitxer(nom, dic):
    with open(nom, "r") as f:
        dic = json.load(f)
    return dic

#Diu que a la clau li sumi 1 cada vegada que hi hagi un element nou
#i li afegeix el seu valor
def afegir(dic):
    n = len(dic) + 1
    dic[str(n)] = input("Introdueix un producte: ")
    return dic

#Apareix un "index" amb tots els productes afegits
#fiques el número del que vols i el pots modificar
```

```

def editar_element(dic, it):
    print("editar")
    for x,y in it:
        print("{}: {}".format(x,y))
    n = input("Indiqui el número del producte a modificar: ")
    dic[n] = input("Introdueix el nou producte: ")
    for x,y in it:
        print("{}: {}".format(x,y))

#El mateix que l'anterior però s'elimina
def eliminar_producte(it, n, pu):
    for x,y in it:
        print("{}: {}".format(x,y))
    n
    pu
    for x,y in it:
        print("{}: {}".format(x,y))

#Llistar tots els elements del fitxer
def llegir_un_per_un(it):
    for x,y in it:
        print("{}: {}".format(x,y))

#Guarda les modificacions que hem fet
def guardar(dic, nom):
    with open(nom, "w") as f:
        json.dump(dic, f, indent=4)
    with open(nom, "r") as f:
        dic = json.load(f)

def pex2():
    #Comprova si el fitxer està creat
    nom = "llista_compra.json"
    dic = {}

    it = dic.items()
    if os.path.isfile(nom):
        dic = carregar_fitxer(nom, dic)
    else:
        crear_fitxer(nom)

    op = -1
    while op!=6:

```

```

    op = menu()
    match(op):
        case 0:
            crear_fixter(nom)
        case 1:
            dic = afegir(dic)
        case 2:
            editar_element(dic, it)
        case 3:
            n = input("Indiqui el número del producte a eliminar:
")
            pu = dic.pop(n)
            dic = eliminar_producte(it, pu, n)
        case 4:
            llegir_un_per_un(it)
        case 5:
            guardar(dic, nom)
        case 6:
            print("Gràcies per emprar aquesta llista!")

```

Per executar aquest programa individualment, és tan simple com després de la funció pex2() escriure a una linea apart fora de la funció= pex2()

```

    4. Llegir llista
    5. Guardar abans de sortir
    6. Sortir

Introdueix una opció: 1
Introdueix un producte: cacao

Llista de compra:
0. Crear llista
1. Afegir element
2. Editar element
3. Eliminar element
4. Llegir llista
5. Guardar abans de sortir
6. Sortir

Introdueix una opció: 4
1: xocolati

2: cacao

Llista de compra:
0. Crear llista
1. Afegir element
2. Editar element
3. Eliminar element
4. Llegir llista
5. Guardar abans de sortir
6. Sortir

Introdueix una opció: █

```

EX3

```
import pygame
from pygame.locals import *
import random

def pex3():
    pygame.init()

    # crear la pestanya
    ancho = 1300
    altura = 1100
    screen_size = (ancho, altura)
    screen = pygame.display.set_mode(screen_size)
    pygame.display.set_caption('Car Game')

    # colors
    gris = (100, 100, 100)
    verde = (76, 208, 56)
    rojo = (200, 0, 0)
    blanco = (255, 255, 255)
    amarillo = (255, 232, 0)

    #fondo
    fondo_imagen =
pygame.image.load("images/editarfondo.png").convert()
    fondo_rect = fondo_imagen.get_rect()

    # tamany carretera i voreres
    road_width = 765
    marker_width = 10
    marker_height = 50

    # cordenades linies blanques
    left_lane = 400
    center_lane = 660
    right_lane = 930
    lanes = [left_lane, center_lane, right_lane]

    #posició carretera i voreres
    road = (265, 0, road_width, altura)
    left_edge_marker = (265, 0, marker_width, altura)
    right_edge_marker = (1030, 0, marker_width, altura)

    #animació linies blanques
```



```

lane_marker_move_y = 0

#coordenades on comença el jugador
player_x = 680
player_y = 700

#configuració FPS
clock = pygame.time.Clock()
fps = 120

#Configuració joc
gameover = False
speed = 2
score = 0

class Vehicle(pygame.sprite.Sprite):

    def __init__(self, image, x, y):
        pygame.sprite.Sprite.__init__(self)

        #escala de les imatges
        image_scale = 100 / image.get_rect().width
        new_width = image.get_rect().width * image_scale
        new_height = image.get_rect().height * image_scale
        self.image = pygame.transform.scale(image, (new_width,
new_height))

        self.rect = self.image.get_rect()
        self.rect.center = [x, y]

class PlayerVehicle(Vehicle):
    #dibuixar el cotxe
    def __init__(self, x, y):
        image = pygame.image.load('images/coche_principal.png')
        super().__init__(image, x, y)

#grups de sprites
player_group = pygame.sprite.Group()
vehicle_group = pygame.sprite.Group()

#cream el cotxu principal i li deim on aparèixer
player = PlayerVehicle(player_x, player_y)
player_group.add(player)

```

```

#carregem les imatges dels enemics
image_filenames = ['coche_lila.png', 'coche_rojo.png',
'coche_verde.png']
vehicle_images = []
#afegim les imatges a la llista buida
for image_filename in image_filenames:
    image = pygame.image.load('images/' + image_filename)
    vehicle_images.append(image)

#carregem la imatge de colisió
crash = pygame.image.load('images/boom.png')
crash_rect = crash.get_rect()

#bucle del joc
running = True
while running:

    clock.tick(fps)

    for event in pygame.event.get():
        if event.type == QUIT:
            running = False
        #establim el fons
        screen.blit(fondo_imagen, fondo_rect)
        #moviment cotxe principal
        if event.type == KEYDOWN:

            if event.key == K_LEFT and player.rect.center[0] >
left_lane:
                player.rect.x -= 250
            elif event.key == K_RIGHT and player.rect.center[0] <
right_lane:
                player.rect.x += 250

        #per detectar si hi ha una colisió
        for vehicle in vehicle_group:
            if pygame.sprite.collide_rect(player, vehicle):

                gameover = True

                #per saber en quin lloc ha estat la colisió

```

```

        if event.key == K_LEFT:
            player.rect.left = vehicle.rect.right
            crash_rect.center = [player.rect.left,
(player.rect.center[1] + vehicle.rect.center[1]) / 2]
        elif event.key == K_RIGHT:
            player.rect.right = vehicle.rect.left
            crash_rect.center = [player.rect.right,
(player.rect.center[1] + vehicle.rect.center[1]) / 2]

#dibuixar la carretera
pygame.draw.rect(screen, gris, road)

#dibuixar les voreres
pygame.draw.rect(screen, amarillo, left_edge_marker)
pygame.draw.rect(screen, amarillo, right_edge_marker)

#dibuixar les linies blanques
lane_marker_move_y += speed * 2
if lane_marker_move_y >= marker_height * 2 :
    lane_marker_move_y = 0
for y in range(marker_height * -2, altura, marker_height * 2):
    pygame.draw.rect(screen, blanco, (left_lane + 400, y +
lane_marker_move_y, marker_width, marker_height))
    pygame.draw.rect(screen, blanco, (center_lane + -140, y +
lane_marker_move_y, marker_width, marker_height))

#dibuixar el cotxe principal
player_group.draw(screen)

#afegir un cotxe
if len(vehicle_group) < 2:

    #mirar que hi hagi prou espai entre cotxe i cotxe
    add_vehicle = True
    for vehicle in vehicle_group:
        if vehicle.rect.top < vehicle.rect.height * 1.5:
            add_vehicle = False

    if add_vehicle:

```

```

        #seleccionar un carril aleatori
        lane = random.choice(lanes)

        #eligeix spawnetjar un coxtu aleatori
        image = random.choice(vehicle_images)
        vehicle = Vehicle(image, lane, altura / -2)
        vehicle_group.add(vehicle)

#velocitat del cotxe
for vehicle in vehicle_group:
    vehicle.rect.y += speed

    #quan es cotxu surti de sa pantalla que desapareixi
    if vehicle.rect.top >= altura:
        vehicle.kill()

    #i sumar 1 punt a la puntuació
    score += 1

    #quan es passin 5 cotxus augmentar la velocitat
    if score > 0 and score % 5 == 0:
        speed += 1

#dibuixar cotxus
vehicle_group.draw(screen)

#escriure la puntuació
font = pygame.font.Font(pygame.font.get_default_font(), 16)
text = font.render('Puntuación: ' + str(score), True, blanco)
text_rect = text.get_rect()
text_rect.center = (60, 30)
screen.blit(text, text_rect)

#vigilar quan hi hagi una colisió
if pygame.sprite.spritecollide(player, vehicle_group, True):
    gameover = True
    crash_rect.center = [player.rect.center[0],
player.rect.top]

#cartell game over
if gameover:
    screen.blit(crash, crash_rect)

```

```

pygame.draw.rect(screen, rojo, (0, 50, ancho, 100))

font = pygame.font.Font(pygame.font.get_default_font(), 16)
text = font.render('Game over. Jugar otra vez? (Presione S
o N)', True, blanco)
text_rect = text.get_rect()
text_rect.center = (ancho / 2, 100)
screen.blit(text, text_rect)

pygame.display.update()

#esperar a q el jugador eligeixi entre tornar a jugar o tancar
la pestanya
while gameover:

    clock.tick(fps)

    for event in pygame.event.get():

        if event.type == QUIT:
            gameover = False
            running = False

        #si es "s" restablir tot
        if event.type == KEYDOWN:
            if event.key == K_s:
                gameover = False
                speed = 2
                score = 0
                vehicle_group.empty()
                player.rect.center = [player_x, player_y]
            elif event.key == K_n:
                #sinos sortir
                gameover = False
                running = False

pygame.quit()

```

Per executar aquest programa individualment, és tan simple com després de la funció pex3() escriure a una línia apart fora de la funció= pex3()

[VIDEO](#)

EX4

```
class Coche:
    def __init__(self, marca, llantas, color, precio):
        #característiques que demanen
        self.marca = marca
        self.llantas = llantas
        self.color = color
        self.precio = precio

        #com s'escruiuran
    def marca_coche(self):
        print(f"Coche: {self.marca}.")

    def tipo_llantas(self):
        print(f"\nEl coche tiene las llantas de {self.llantas}.")

    def color_coche(self):
        print(f"\nEl coche es de color {self.color}.")

    def precio_coche(self):
        print(f"\nEl coche cuesta {self.precio}.\n\n")

#cream els objectes amb classes
class Renault(Coche):
    def __init__(self, marca, llantas, color, precio):
        super().__init__(marca, llantas, color, precio)

class BMW(Coche):
    def __init__(self, marca, llantas, color, precio):
        super().__init__(marca, llantas, color, precio)

class Toyota(Coche):
    def __init__(self, marca, llantas, color, precio):
        super().__init__(marca, llantas, color, precio)

class Opel(Coche):
    def __init__(self, marca, llantas, color, precio):
        super().__init__(marca, llantas, color, precio)

def pex4():
    #li pasem les dades de cada objecte a la llista
    l =[Renault("Renault Clio", "18 pulgadas", "verde metalizado",
"25.000€"), BMW("BMW M2", "20 pulgadas", "azul mate", "50000€"),
```

```

Toyota("Toyota Supra", "18 pulgadas", "rojo", "35000€"), Opel("Opel
Corsa", "14 pulgadas", "negro", "20000€")]
    for e in l:
        #li aplica les següents funcions als elements de la llista
        e.marca_coche()
        e.tipo_llantas()
        e.color_coche()
        e.precio_coche()

```

Per executar aquest programa individualment, és tan simple com després de la funció pex4() escriure a una línia apart fora de la funció= pex4()

```

PS D:\Projecte> & C:/Users/usuario/AppData/Local/Microsoft/WindowsApps/python3.11.exe d:/Projecte/ex4.py
Coche: Renault Clio.

El coche tiene las llantas de 18 pulgadas.

El coche es de color verde metalizado.

El coche cuesta 25.000€.

Coche: BMW M2.

El coche tiene las llantas de 20 pulgadas.

El coche es de color azul mate.

El coche cuesta 50000€.

Coche: Toyota Supra.

El coche tiene las llantas de 18 pulgadas.

El coche es de color rojo.

El coche cuesta 35000€.

Coche: Opel Corsa.

El coche tiene las llantas de 14 pulgadas.

El coche es de color negro.

El coche cuesta 20000€.

PS D:\Projecte>

```

EX5

```
import re
import requests

#web la qual agafem la informació
website =
"https://www.mercadolibre.com.ar/c/computacion#menu=categories"
#agafem el script
resultado = requests.get(website)
#pasem tot a text
content = resultado.text
patron = r"https://listado.mercadolibre.com.ar/computacion/[\w-]*"
#cerca dins dels resultats de text tot el que contingui el patró
maquines_repetides = re.findall(patron, str(content))

#evitem que apareguin apartats duplicats
no_duplicats = list(set(maquines_repetides))

maquines_finals = []

def pex5():
    print("Apartats de informàtica de mercado libre: ")
    for i in no_duplicats:
        #substitueix tota la ruta per res
        nom_maquines =
i.replace("https://listado.mercadolibre.com.ar/computacion/", "")
        maquines_repetides.append(maquines_finals)
        print(nom_maquines)
```

Per executar aquest programa individualment, és tan simple com després de la funció pex5() escriure a una línia apart fora de la funció= pex5()

```
PS D:\Projecte> & C:/Users/usuario/AppData/Local/Microsoft/WindowsApps/python3.11.exe d:/Projecte/ex5.py
```

```
Apartats de informàtica de mercado libre:
monitores-accesorios
laptops-accesorios
almacenamiento
perifericos-pc
impresion
accesorios-pc-gaming
lectores-scanners
tablets-accesorios
software
accesorios-antiestatica
_Envio_Full_Deal_electronica-24-q2
notebooks-accesorios
cables-hubs-usb
otros
conectividad-redes
limpieza-cuidado-pcs
_Deal_electronica-24-q2
componentes-pc
estabilizadores-ups
pc-escritorio
_Deal_electronica-24-q2_Installments_NoInterest
palms-handhelds
proyectores-pantallas
PS D:\Projecte>
```


EX6

```
import http.server
import socketserver
import time

def pex6():
    #defineix el port en el que s'executarà
    PORT = 9000

    #configura l'administrador de sollicitut HTTP
    request = http.server.SimpleHTTPRequestHandler

    #crear objecte de servidor
    httpd = socketserver.TCPServer(("", PORT), request)

    print(f"El servidor està funcionant en el port {PORT}")

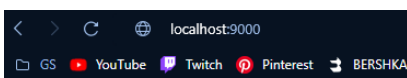
    #temps que durarà el servidor obert(2 min)
    temps = 2*60

    #iniciar el servidor i esperar 2 min
    try:
        httpd.serve_forever()
        time.sleep(temps)
    except KeyboardInterrupt:
        pass

    #després de 2 min tancar
    http.shutdown()
    print(f"El servidor ara està aturat")
```

Per executar aquest programa individualment, és tan simple com després de la funció pex6() escriure a una línia apart fora de la funció= pex6()

```
PS D:\Projecte> & C:/Users/usuario/AppData/Local/Microsoft/WindowsApps/python3.11.exe d:/Projecte/ex6.py
El servidor està funcionant en el port 9000
```



Directory listing for /

- [__pycache__/](#)
- [ex1.py](#)
- [ex2.py](#)
- [ex3.py](#)
- [ex4.py](#)
- [ex5.py](#)
- [ex6.py](#)
- [images/](#)
- [llista_compra.json](#)
- [menu.py](#)