

Pensar paso a paso antes de programar

Fecha: _____ Tiempo de práctica: _____

Reparte 23 fotografías en sobres de 5. Repite con 20, 0 y capacidad 0.

1. Objetivo y resultado esperado

2. Entradas y restricciones

3. Algoritmo numerado

4. Prueba manual y corrección

Observaciones

Preparar Python y tu espacio de trabajo

Fecha: _____ Tiempo de práctica: _____

Guarda comprobacion.py, ejecútalo, cambia el mensaje y vuelve a abrirlo.

1. Sistema y versión comprobada

2. Carpeta y archivo de trabajo

3. Orden de ejecución utilizada

4. Prueba de cierre y reapertura

Observaciones

Escribir y ejecutar tu primer programa

Fecha: _____ Tiempo de práctica: _____

Escribe tres mensajes, predice el orden y provoca un único error de comillas.

1. Predicción de las tres salidas

2. Código del saludo personalizado

3. Error provocado y mensaje recibido

4. Corrección y nueva ejecución

Observaciones

Guardar información con variables y tipos

Fecha: _____ Tiempo de práctica: _____

Usa el código 0024 y 8 unidades. Resta 3, añade 2 y conserva la cantidad inicial.

1. Variables del inventario

2. Tipos elegidos y motivo

3. Valores antes y después

4. Prueba con un dato distinto

Observaciones

Pedir datos y mostrar resultados claros

Fecha: _____ Tiempo de práctica: _____

Pide nombre y paquetes de 3 sobres. Prueba 4 paquetes y una entrada no numérica.

1. Preguntas y entradas de prueba

2. Normalización del nombre

3. Conversión de cantidades

4. Salida prevista y salida real

Observaciones

Calcular y comparar sin perder el sentido

Fecha: _____ Tiempo de práctica: _____

Calcula sobres para 23, 20, 0 y 1 fotografías con capacidad de 5.

1. Fórmula y unidades

2. Reparto exacto y con resto

3. Predicción con paréntesis

4. Pruebas del cálculo de cajas

Observaciones

Tomar decisiones con condiciones

Fecha: _____ Tiempo de práctica: _____

Clasifica las cantidades -1, 0, 1, 5 y 6. Rechaza capacidad cero al repartir.

1. Reglas de clasificación

2. Casos en los límites

3. Código y orden de las ramas

4. Resultado de cada prueba

Observaciones

Repetir tareas con bucles

Fecha: _____ Tiempo de práctica: _____

Suma 2, 4 y 1 con un bucle. Prueba el menú y un recorrido sin repeticiones.

1. Secuencia que debe repetirse

2. Acumulador paso a paso

3. Condición de salida del menú

4. Prueba con cero repeticiones

Observaciones

Organizar colecciones con listas

Fecha: _____ Tiempo de práctica: _____

Crea tres tareas, añade otra, sustituye la segunda y elimina la primera.

1. Lista inicial y posiciones

2. Altas y bajas realizadas

3. Recorrido numerado

4. Pruebas de lista vacía y duplicados

Observaciones

Describir registros con diccionarios

Fecha: _____ Tiempo de práctica: _____

Crea dos registros. Completa solo el primero y consulta un campo inexistente.

1. Campos de una tarea

2. Dos registros completos

3. Cambio del estado pendiente

4. Prueba de independencia entre registros

Observaciones

Crear funciones para reutilizar soluciones

Fecha: _____ Tiempo de práctica: _____

Prueba calcular_cajas y contar_pendientes. Compara devolver y mostrar valores.

1. Contrato de la función

2. Parámetros y resultado

3. Tres llamadas con datos diferentes

4. Comparación entre print y return

Observaciones

Leer errores y validar las entradas

Fecha: _____ Tiempo de práctica: _____

Prueba cinco, -2, 2.5, vacío y 4. Corrige además un NameError provocado.

1. Mensaje y línea del error

2. Hipótesis y cambio realizado

3. Entradas válidas e inválidas

4. Bucle de recuperación comprobado

Observaciones

Comprobar tu código con pruebas

Fecha: _____ Tiempo de práctica: _____

Prepara cuatro aserciones de cajas y comprueba que dos tareas son independientes.

1. Casos y resultados esperados

2. Aserciones ejecutadas

3. Fallo detectado y corrección

4. Pruebas repetidas tras el cambio

Observaciones

Ordenar el proyecto y usar módulos

Fecha: _____ Tiempo de práctica: _____

Crea calculos.py y principal.py. Copia el proyecto y comprueba las importaciones.

1. Mapa de archivos

2. Importaciones y responsabilidades

3. Prueba al importar el módulo

4. Instrucciones para ejecutar el proyecto

Observaciones

Leer y guardar archivos de texto

Fecha: _____ Tiempo de práctica: _____

Guarda texto con tildes, recupéralo y distingue reemplazar, leer y añadir.

1. Ruta del archivo utilizado

2. Texto guardado y recuperado

3. Prueba de archivo inexistente

4. Copia de seguridad y límites

Observaciones

Conservar datos estructurados con JSON

Fecha: _____ Tiempo de práctica: _____

Guarda dos tareas. Prueba una coma inválida, un título vacío y un estado de texto.

1. Estructura de datos elegida

2. Ida y vuelta del archivo

3. Archivo dañado y respuesta esperada

4. Validación de campos y tipos

Observaciones

Diseñar un gestor de tareas útil

Fecha: _____ Tiempo de práctica: _____

Define usuario, operaciones, campos y menú. Redacta seis criterios comprobables.

1. Usuario y alcance del gestor

2. Modelo de datos y operaciones

3. Boceto del menú

4. Criterios de aceptación

Observaciones

Construir el gestor de tareas

Fecha: _____ Tiempo de práctica: _____

Prueba lista vacía, dos altas, un completado y tres rechazos. Cierra y reabre.

1. Archivos y funciones creados

2. Alta y listado comprobados

3. Tarea completada y rechazos

4. Salida del menú sin guardar

Observaciones

Añadir persistencia y revisar la aplicación

Fecha: _____ Tiempo de práctica: _____

Guarda dos tareas, reabre y prueba salir sin guardar. Daña solo una copia.

1. Carga inicial y ruta elegida

2. Guardado y reapertura

3. Prueba con JSON dañado

4. Revisión de mensajes y límites

Observaciones

Terminar el proyecto y seguir aprendiendo

Fecha: _____ Tiempo de práctica: _____

Ejecuta la demostración y pruebas.py. Elige una mejora y planifica cuatro semanas.

1. Demostración de principio a fin

2. Matriz final de comprobaciones

3. Mejora elegida y prueba asociada

4. Plan de práctica de cuatro semanas

Observaciones
