

Tu calificación: 100 %

Próximo artículo →

Tu calificación más reciente: 100 % • Tu calificación más alta: 100 % • Para aprobar necesitas al menos un 80 %. Guardamos tu puntaje más alto.

Estás viendo una versión traducida automáticamente de esta evaluación

Puedes volver a ver el contenido en su idioma original si lo prefieres. No perderás el progreso que hayas conseguido si cambias el idioma.

Mostrar la versión en Inglés

Desestimar

1.

La función create_python_script crea un nuevo script de Python en el directorio de trabajo actual, le añade la línea de comentarios declarada por la variable 'comments' y devuelve el tamaño del nuevo archivo. Rellena los huecos para crear un script llamado "programa.py".

1 / 1 punto

```
1 def create_python_script(filename):
2     comments = "# Start of a new Python program"
3     with open(filename, 'w') as file: # Abre el archivo en modo escritura
4         file.write(comments) # Escribe el comentario en el archivo
5         filesize = file.tell() # Obtiene el tamaño del archivo en el punto actual (después de escribir)
6     return(filesize)
7
8 print(create_python_script("program.py"))
```

Ejecutar

Restablecer

Correcto

¡Buen trabajo! Tu nuevo script de Python está listo para algo de código real!

2.

La función nuevo_directorio crea un nuevo directorio dentro del directorio de trabajo actual, luego crea un nuevo archivo vacío dentro del nuevo directorio, y devuelve la lista de archivos en ese directorio. Rellena los huecos para crear un archivo "script.py" en el directorio "PythonPrograms".

1 / 1 punto

```
1 import os
2
3 def new_directory(directory, filename):
4     # Before creating a new directory, check to see if it already exists
5     if not os.path.exists(directory): # Verifica si el directorio no existe
6         os.mkdir(directory) # Crea el directorio si no existe
7
8     # Create the new file inside of the new directory
9     file_path = os.path.join(directory, filename) # Define la ruta completa del archivo
10    with open(file_path, "w") as file: # Crea un archivo vacío
11        pass
12
13    # Return the list of files in the new directory
14    return os.listdir(directory) # Devuelve la lista de archivos en el directorio
15
16 print(new_directory("PythonPrograms", "script.py"))
17
```

Ejecutar

Restablecer

Correcto

¡Bien hecho! Trabajar con archivos y directorios puede ser un poco complicado, ¡y ya le estás cogiendo el truco!

3.

¿Cuál de los siguientes métodos del módulo OS creará un nuevo directorio?

1 / 1 punto

- ☐ ruta.isdir()
- ☐ listdir()
- ☒ mkdir()
- ☐ chdir()

Correcto

RIGHT on! os.mkdir() creará un nuevo directorio con el nombre proporcionado como parámetro de cadena.

4.

La función fecha_archivo crea un nuevo archivo en el directorio de trabajo actual, comprueba la fecha en que se modificó el archivo y devuelve sólo la parte de fecha de la marca de tiempo en el formato aaaa-mm-dd. Rellene los campos para crear un archivo llamado "archivo_nuevo.txt" y compruebe la fecha en que se modificó.

1 / 1 punto

```
1 import os
2 import datetime
3
4 def file_date(filename):
5     # Create the file in the current directory
6     with open(filename, 'w') as file:
7         pass # Create an empty file
8
9     # Get the timestamp of the file
10    timestamp = os.path.getmtime(filename)
11
12    # Convert the timestamp into a readable format, then into a string
13    readable_time = datetime.datetime.fromtimestamp(timestamp).strftime('%Y-%m-%d')
14
15    # Return just the date portion
16    return readable_time
17
18 print(file_date("newfile.txt"))
19 # Should be today's date in the format of yyyy-mm-dd
20
```

Ejecutar

Restablecer

Correcto

Bien hecho Recordaste los comandos para convertir marcas de tiempo y formatear cadenas, para obtener los resultados solicitados.

5.

La función directorio_padre devuelve el nombre del directorio que se encuentra justo encima del directorio de trabajo actual. Recuerde que '..' es un alias de ruta de acceso relativa que significa "subir al directorio principal". Rellena los huecos para completar esta función.

1 / 1 punto

```
1 import os
2
3 def parent_directory():
4     # Create a relative path to the parent
5     # of the current working directory
6     relative_parent = os.path.join(os.getcwd(), '..')
7
8     # Return the absolute path of the parent directory
9     return os.path.abspath(relative_parent)
10
11 print(parent_directory())
12
```

Ejecutar

Restablecer

Correcto

¡Excelente! Usted hizo todos los movimientos correctos para imprimir la ruta de directorio principal