

El módulo `os` en Python proporciona y expone los detalles y la funcionalidad del sistema operativo.

Proporciona funcionalidad independiente del sistema operativo.

Podemos usar el módulo `os` importando a través del siguiente comando.

```
import os
```

Las siguientes son algunas llamadas importantes en el módulo `os`.

```
os.getcwd()
```

La llamada `os.getcwd` obtiene el directorio de trabajo actual.

```
os.getpid()
```

La llamada `os.getpid` obtiene el pid actual.

```
os.getuid()
```

La llamada `os.getuid` obtiene el ID de usuario actual.

```
os.uname()
```

La llamada `os.uname` obtiene el nombre del sistema operativo actual.

```
os.rename(source, destination)
```

La llamada `os.rename` cambia el nombre de un archivo del origen al destino.

```
os.remove(filepath)
```

La función `os.remove` elimina la ruta de archivo dada de la ubicación.

```
os.rmdir(dirpath)
```

La función `os.rmdir` borra el directorio.

```
os.removedirs(dirpath)
```

La función `os.removedirs` elimina el directorio y todos los subdirectorios dentro de él recursivamente.

```
os.listdir(dirpath)
```

La función `os.listdir` recupera los archivos en el directorio.

```
os.mkdir(dirpath)
```

La función `os.mkdir` crea un directorio.

os.makedirs(dirpath)

La función `os.makedirs` crea el directorio y todos los directorios primarios requeridos recursivamente.

Algunos ejemplos en acción:

```
>>> import os
```

```
>>> os.getcwd()
```

```
>>> os.uname()
```

```
>>> os.listdir(os.getcwd())
```

```
>>> os.sys.stdout
```

```
>>> os.path.exists("/Users/mediacenter")
True
```

```
>>> os.path.isdir("/Users/mediacenter")
True
```