UF3.9 Librerías de Python

Centro Profesional
Universidad Europea de Madrid
LAUREATE INTERNATIONAL UNIVERSITIES

Contenidos

- Librerías de Python
- Modulos de Sistema: modulo os
- Modulos de Sistema: modulo sys
- Otros módulos

Librerías de Python

Python nos provee de un gran abanico de módulos que integran su librería estándar, como bien puede verse en el manual oficial: https://docs.python.org/2/library/index.html

Vamos a ver algunos de ellos que se destacan ya sea por la frecuencia de uso o por sus prestaciones.

Módulos de Sistema: modulo os

El **módulo os** nos permite acceder a funcionalidades dependientes del Sistema Operativo. Sobre todo, aquellas que nos refieren información sobre el entorno del mismo y nos permiten manipular la estructura de directorios.

Métodos destacados:

Descripción	Método
Saber si se puede acceder a un archivo o directorio	os.access(path, modo_de_acceso)
Conocer el directorio actual	**os.getcwd()**
Cambiar de directorio de trabajo	os.chdir(nuevo_path)
Cambiar al directorio de trabajo raíz	os.chroot()
Cambiar los permisos de un archivo o directorio	os.chmod(path, permisos)
Cambiar el propietario de un archivo o directorio	os.chown(path, permisos)

Módulos de Sistema: modulo os

Descripción	Método
Crear un directorio	**os.mkdir(path[, modo])**
Crear directorios recursivamente	os.mkdirs(path[, modo])
Eliminar un archivo	**os.remove(path)**
Eliminar un directorio	**os.rmdir(path)**
Eliminar directorios recursivamente	os.removedirs(path)
Renombrar un archivo	**os.rename(actual, nuevo)**
Crear un enlace simbólico	os.symlink(path, nombre_destino)

Módulos de Sistema: modulo os

El módulo os y las variables de entorno

El módulo os también nos provee de un diccionario con las variables de entorno relativas al sistema. Se trata del diccionario **environ**:

```
import os

for variable, valor in os.environ.iteritems():
     print "%s: %s" % (variable, valor)
```

Módulos de Sistema: modulo os

El módulo os también nos provee del submódulo path (**os.path**) el cual nos permite acceder a ciertas funcionalidades relacionadas con los nombres de las rutas de archivos y directorios.

Métodos destacables:

Descripción	Método
Ruta absoluta	os.path.abspath(path)
Directorio base	os.path.basename(path)
Saber si un directorio existe	**os.path.exists(path)**
Conocer último acceso a un directorio	os.path.getatime(path)
Conocer tamaño del directorio	**os.path.getsize(path)**
Saber si una ruta es absoluta	os.path.isabs(path)
Saber si una ruta es un archivo	os.path.isfile(path)
Saber si una ruta es un directorio	os.path.isdir(path)
Saber si una ruta es un enlace simbólico	os.path.islink(path)
Saber si una ruta es un punto de montaje	os.path.ismount(path)

Módulos de Sistema: modulo sys

El **módulo sys** es el encargado de proveer variables y funcionalidades, directamente relacionadas con el intérprete.

Variables destacadas:

Variable	Descripción
sys.argv	Retorna una lista con todos los argumentos pasados por línea de comandos. Al ejecutar python modulo.py arg1 arg2, retornará una lista: ['modulo.py', 'arg1', 'arg2']
sys.executable	Retorna el path absoluto del binario ejecutable del intérprete de Python
sys.maxint	Retorna el número positivo entero mayor, soportado por Python
sys.platform	Retorna la plataforma sobre la cuál se está ejecutando el intérprete
sys.versión	Retorna el número de versión de Python con información adicional
sys.path	Retorna una lista de cadenas que determinan el camino de búsqueda del intérprete para los módulos.

Módulos de Sistema: modulo sys

Métodos destacados:

Método	Descripción
sys.exit()	Forzar la salida del intérprete. Es la forma más directa de terminar un programa.
sys.getdefaultencoding()	Retorna la codificación de caracteres por defecto
sys.getfilesystemencoding()	Retorna la codificación de caracteres que se utiliza para convertir los nombres de archivos unicode en nombres de archivos del sistema
sys.getsizeof(object[, default])	Retorna el tamaño del objeto pasado como parámetro. El segundo argumento (opcional) es retornado cuando el objeto no devuelve nada.
sys.dir([modulo])	Devuelve una lista ordenada de cadenas que representan los nombres de variables, módulos, funciones, etc, que se definen en un módulo

Módulos de Sistema: modulo sys

El **módulo sys** también tiene atributos para stdin, stdout, y stderr. Este último es útil para emitir mensajes de alerta y error para que se vean incluso cuando se haya redireccionado stdout.

```
sys.stderr.write('Alerta, archivo de log no encontrado\n')
```

Otros módulos

El módulo **glob** provee una función para hacer listas de archivos a partir de búsquedas con comodines en directorios.

```
import glob
print glob.glob('*.py')
```

El módulo **math** permite el acceso a las funciones de la biblioteca C subyacente para la matemática de punto flotante

```
import math
math.cos(math.pi / 4)
```

Otros módulos

El módulo **random** provee herramientas para realizar selecciones al azar:

```
import random
random.choice(['manzana', 'pera', 'banana'])
random.random()    # un float al azar
random.randrange(6)    # un entero al azar tomado de range(6)
```

Otros módulos

El módulo **statistics** calcula propiedades de estadística básica (la media, mediana, varianza, etc) de datos númericos.

```
import statistics

datos = [2.75, 1.75, 1.25, 0.25, 0.5, 1.25, 3.5]

statistics.mean(datos)

statistics.median(datos)

statistics.variance(datos)
```

Otros módulos

El módulo **datetime** ofrece clases para manejar fechas y tiempos tanto de manera simple como compleja. Aunque soporta aritmética sobre fechas y tiempos, el foco de la implementación es en la extracción eficiente de partes para manejarlas o formatear la salida. El módulo también soporta objetos que son conscientes de la zona horaria.

```
from datetime import date
hoy = date.today()
print hoy
fecha = date(2009, 7, 19)
print fecha
print hoy.strftime("%m-%d-%y. %d %b %Y es %A. hoy es %d de %B.")
```

Otros módulos

Las siguientes directivas se pueden utilizar en el formato de cadena:

%a - Nombre del día de la semana
%A - Nombre del día completo
%b - Nombre abreviado del mes
%B - Nombre completo del mes
%c - Fecha y hora actual
%d - Día del mes
%H - Hora (formato 24 horas)
%I - Hora (formato 12 horas)
%j - Día del año
%m - Mes en número
%M- Minutos
%p - Equivalente de AM o PM
%S - Segundos
%U - Semana del año (domingo como primer día de la semana)
%w - Día de la semana
%W - Semana del año (lunes como primer día de la semana)
%x - Fecha actual
%X - Hora actual
%y - Número de año (14)
%Y - Numero de año entero (2014)
%Z - Zona horaria

Otros módulos

```
# las fechas soportan aritmética de calendario
nacimiento = date(2005, 1, 13)
edad = hoy - nacimiento
print edad.days
```

Bibliografía y Webgrafía

http://librosweb.es/libro/python/

http://docs.python.org.ar/tutorial/3/stdlib.html