



PYTHON

Módulo 8 – Uso de módulos estándar de Python

1

USO DE MÓDULOS ESTÁNDAR DE PYTHON

Módulos estándar

USO DE MÓDULOS ESTÁNDAR DE PYTHON

- Módulos estándar:
 - Vienen incluidos con la distribución de Python.
 - <https://docs.python.org/3/library/index.html>

USO DE MÓDULOS ESTÁNDAR DE PYTHON

- Módulos estándar:
 - Numeric and Mathematical Modules.
 - Funciones y tipos de datos numéricos y relacionados con las matemáticas.
 - <https://docs.python.org/3/library/numeric.html>
 - Functional Programming Modules.
 - Funciones y clases que admiten un estilo de programación funcional y operaciones generales invocables (callable).
 - <https://docs.python.org/3/library/functional.html>

USO DE MÓDULOS ESTÁNDAR DE PYTHON

- Módulos estándar:
 - File and Directory Access.
 - Los módulos de esta sección se ocupan de los archivos y directorios del disco.
 - <https://docs.python.org/3/library/filesys.html>.
 - Data persistence.
 - Los módulos de esta categoría hacen referencia al almacenamiento de datos de Python de forma persistente en el disco.
 - <https://docs.python.org/3/library/persistence.html>

USO DE MÓDULOS ESTÁNDAR DE PYTHON

- Módulos estándar:
 - Data Compression and Archiving.
 - Compresión de datos con los algoritmos zlib, gzip, bzip2 e lzma, y la creación de archivos con formato ZIP y tar.
 - <https://docs.python.org/3/library/archiving.html>
 - File Formats.
 - Varios formatos de archivo que no son lenguajes de marcado y no están relacionados con el correo electrónico.
 - <https://docs.python.org/3/library/fileformats.html>

USO DE MÓDULOS ESTÁNDAR DE PYTHON

- Módulos estándar:
 - Cryptographic Services.
 - Implementan varios algoritmos de naturaleza criptográfica.
 - <https://docs.python.org/3/library/crypto.html>
 - Generic Operating System Services.
 - Proporcionan interfaces a las características del sistema operativo, como archivos y un reloj.
 - <https://docs.python.org/3/library/os.html>

USO DE MÓDULOS ESTÁNDAR DE PYTHON

- Módulos estándar:
 - Concurrent Execution.
 - Provee soporte para la ejecución concurrente de código.
 - <https://docs.python.org/3/library/concurrency.html>
 - Networking and Interprocess Communication.
 - Proporcionan mecanismos para la creación de redes y la comunicación entre procesos.
 - <https://docs.python.org/3/library/ipc.html>

USO DE MÓDULOS ESTÁNDAR DE PYTHON

- Módulos estándar:
 - Internet Data Handling.
 - Módulos que soportan el manejo de formatos de datos usados comúnmente en Internet.
 - <https://docs.python.org/3/library/netdata.html>
 - Structured Markup Processing Tools.
 - Módulos para trabajar con SGML, HTML y XML.
 - <https://docs.python.org/3/library/markup.html>

USO DE MÓDULOS ESTÁNDAR DE PYTHON

- Módulos estándar:
 - Internet Protocols and Support.
 - Implementan protocolos de Internet y son compatibles con la tecnología relacionada. La mayoría de estos módulos requieren la presencia del módulo dependiente del sistema socket.
 - <https://docs.python.org/3/library/internet.html>
 - Multimedia Services.
 - Varios algoritmos o interfaces que son útiles principalmente para aplicaciones multimedia.
 - <https://docs.python.org/3/library/mm.html>

USO DE MÓDULOS ESTÁNDAR DE PYTHON

- Módulos estándar:
 - Internationalization.
 - Permite crear programas independientes del idioma y lugar al proporcionar mecanismos para seleccionar un idioma o adaptar la salida para que coincida con las convenciones locales.
 - <https://docs.python.org/3/library/i18n.html>
 - Program Frameworks.
 - Incluye *frameworks* como Turtle (dibujo), Cmd (escribir intérpretes de comando y shlex(Analizador léxico).
 - <https://docs.python.org/3/library/frameworks.html>

USO DE MÓDULOS ESTÁNDAR DE PYTHON

- Módulos estándar:
 - Graphical User Interfaces with Tk.
 - Permite desarrollar interfaces gráficas de usuario (GUI).
 - <https://docs.python.org/3/library/tk.html>
 - Development Tools.
 - Herramientas de ayuda al desarrollo. Generación de documentación (pydoc), pruebas unitarias (unittest y doctest) y traductores de código de versión 2 a versión 3 de Python.
 - <https://docs.python.org/3/library/development.html>

USO DE MÓDULOS ESTÁNDAR DE PYTHON

- Módulos estándar:
 - Debugging and Profiling.
 - Herramientas de depuración y análisis de rendimiento.
 - <https://docs.python.org/3/library/debug.html>
 - Software Packaging and Distribution.
 - Herramientas empaquetado y distribución del software.
Generación de entornos virtuales con venv, instalación de paquetes y generación de versiones compiladas.
 - <https://docs.python.org/3/library/distribution.html>

USO DE MÓDULOS ESTÁNDAR DE PYTHON

- Módulos estándar:
 - Python Runtime Services.
 - Proporcionan servicios relacionados con el intérprete de Python y su interacción con su entorno.
 - <https://docs.python.org/3/library/python.html>
 - Custom Python Interpreters.
 - Los módulos descritos en este grupo permiten escribir interfaces similares al intérprete interactivo de Python.
 - <https://docs.python.org/3/library/custominterp.html>

USO DE MÓDULOS ESTÁNDAR DE PYTHON

- Módulos estándar:
 - Importing Modules.
 - Proporcionan utilidades para la importación de módulos de maneras no habituales, como mediante ficheros zip.
 - <https://docs.python.org/3/library/modules.html>
 - Python Language Services.
 - Módulos de ayuda para a trabajar con el lenguaje Python: tokenización, análisis, análisis sintáctico, desensamblado de código de bytes.
 - <https://docs.python.org/3/library/language.html>

USO DE MÓDULOS ESTÁNDAR DE PYTHON

- Módulos estándar:
 - MS Windows Specific Services.
 - Módulos exclusivos de Windows.
 - <https://docs.python.org/3/library/windows.html>
 - Unix Specific Services.
 - Módulos específicos de Unix.
 - <https://docs.python.org/3/library/unix.html>

USO DE MÓDULOS ESTÁNDAR DE PYTHON

- Módulos datetime, calendar y time
- datetime:
 - <https://docs.python.org/3/library/datetime.html>
 - Clases:
 - `datetime.date` → Fecha (atributos: year, month y day)
 - `datetime.time` → Hora (atributos: hour, minute, second, microsecond)
 - `datetime.datetime` → Fecha y hora
 - **`datetime.timedelta`(days=0, seconds=0, microseconds=0, milliseconds=0, minutes=0, hours=0, weeks=0) → Representa un incremento temporal.**
 - **`datetime.date.today()`→Obtiene la fecha actual (objeto date)**
 - **`datetime.datetime.now()`→Obtiene la fecha y hora actual (objeto datetime)**

USO DE MÓDULOS ESTÁNDAR DE PYTHON

- Módulos datetime, calendar y time

- datetime:

- Operaciones: + y -

- `>>> offset = datetime.timedelta(days=10)`
 - `>>> offset`
 - `datetime.timedelta(days=10)`
 - `>>> fecha_fin = datetime.date.today() + offset`
 - `>>> fecha_fin = datetime.date.today() - offset`

USO DE MÓDULOS ESTÁNDAR DE PYTHON

- Módulos datetime, calendar y time
- datetime:
 - Número de días transcurridos entre dos fechas
 - `>>> nacimiento = datetime.date(1971,9,3)`
 - `>>> hoy = datetime.date.today()`
 - `>>> diferencia = hoy - nacimiento`
 - `>>> print(diferencia.days)`

USO DE MÓDULOS ESTÁNDAR DE PYTHON

- Módulos datetime, calendar y time
- datetime:
 - Fijar un periodos:
 - `>>> periodo = datetime.timedelta(days=15)`
 - `>>> fecha_vencimiento = datetime.datetime.now() + periodo`
 - `>>> "Fecha vencimiento: {}-{}-{}".format(fecha_vencimiento.day, fecha_vencimiento.month, fecha_vencimiento.year)`