

CURSO PYTHON PARA EXPERTOS EN SEGURIDAD



DURACIÓN DEL CURSO
40 HRS

Informes: Telf. (01) 640-5805 | Av. Arenales N° 1912 Of. 805 - Lince

AULATEC
FORMACIÓN Y DESARROLLO PROFESIONAL EN T.I.

TEMARIO

SEMANA 01

- Introducción a Python como lenguaje de programación: clasificación, ventajas, desventajas, paradigmas de programación soportados, diferencias entre versiones.
- Configuración del entorno de trabajo de Python en Kali Linux.
- Entendiendo el código binario: sistemas de numeración, cambio de base, bits más significativos y bits menos significativos, diferencias entre sistemas big endian y little endian, representación de enteros en base 2, 8, 10, 16, 58 y 64, conceptos de bit, nibble y byte, múltiplos de los bytes.
- Codificación de textos: diferencias entre archivos de texto y archivos binarios, la tabla ASCII, la tabla UNICODE, juegos de caracteres ISO y codificación UTF8.
- Fundamentos de scripting en Python: codificación del juego de caracteres del código fuente, shebang, configuración del editor de textos, manejo de la sangría (identación) en los programas de Python.
- Introducción a los objetos en Python: funciones vs métodos, atributos ejecutables y no ejecutables
Tipos de datos básicos: enteros, flotantes, cadenas, booleanos y el objeto None.
- Operaciones aritméticas y lógicas: sumas, restas, multiplicación, división entera, división flotante, resto de la división, conjunción lógica (AND), disyunción lógica (OR), disyunción exclusiva (XOR), negación lógica (NOT).
- Formateo e interpolación de cadenas: concatenación de cadenas con el operador +, conversión a cadenas con la función str, interpolación con el operador %, formateo e interpolación con el método format, cadenas para formateo.
- Ejercicios: divisibilidad, detección de años bisiestos, incremento circular, cálculo de probabilidades en función de frecuencias, cálculo y presentación de ratios y porcentajes, presentación de datos en formato binario, octal y hexadecimal.

SEMANA 02

- Control del flujo del programa: programación secuencial vs programación con bucles y condicionales
- Modificando los ciclos de ejecución de bucles con las sentencias break y continue
- Simulación de estructuras de control no disponibles en Python: switch-case, do-while y repeat-until
- Gestión de las excepciones en Python con bloques try-except-finally
- Tipos de datos no-escalares en Python: listas, tuplas, conjuntos y diccionarios
- Métodos para el manejo de cadenas: conversión a mayúsculas, minúsculas, eliminación de espacios en blanco, relleno.
- Generación de datos aleatorios: números enteros aleatorios, elementos de secuencias elegidos aleatoriamente, muestras seleccionadas aleatoriamente y simulación de eventos con tablas de probabilidad acumulada
- Técnicas fundamentales de programación aplicando mutabilidad: contadores, acumuladores y banderas
- Gestión de la entrada de datos por teclado, conversión de tipos y patrones de validación
- Ejercicios: generación aleatoria de direcciones IP, conversión de una dirección IPv4 a formato decimal, aplicación de máscaras de subneteo

TEMARIO

SEMANA 03

- Introducción a la programación procedural: subrutinas, procedimientos y funciones.
- Gestión de argumentos en funciones: argumentos obligatorios vs opcionales, valores por omisión, paso de argumentos por posición y por nombre, errores típicos en la invocación de funciones, absorción de argumentos no esperados por posición y por nombre, expansión de argumentos por posición y por nombre, invocación programática de funciones vs invocación manual.
- Dominando los procesos fundamentales de la programación: codificación y abstracción.
- Introducción a la programación funcional: mutabilidad vs inmutabilidad, recursividad, funciones como objetos de primer orden en Python, funciones anónimas con la sentencia lambda, funciones map, reduce y filter, ámbitos, closures y decoradores.
- Listas por comprensión como una sintaxis más conveniente de expresar las operaciones map y filter
- Ejercicios: validación de tarjetas de crédito, generación de sentencias insert y select en SQL, generación aleatoria de datos falsos, generación de combinaciones y permutaciones para ataques por fuerza bruta.

SEMANA 04

- Introducción a la programación orientada a objetos en Python
- Conceptos de encapsulamiento, herencia y polimorfismo
- Métodos de instancia, métodos de clase y métodos estáticos
- Explorando y modificando objetos en Python: funciones dir, callable, hasattr, getattr y setattr
- Principales protocolos y métodos mágicos en Python: `__str__`, `__repr__`, `__getattr__`, `__setattr__`, `__getitem__`, `__setitem__`, `__len__`, entre otros.
- Protocolo de interacción y creación de objetos iterables: `iter()`, `next()`, `IterIteration`
- Generadores, sentencia `yield` y expresiones de generador
- Gestión de archivos en Python: abriendo archivos de texto y binarios en distintos modos: solo lectura, lectura y escritura, recreando el contenido, agregando contenido adicional.
- Lectura y escritura de datos en archivos con formatos .csv y JSON.
- Operaciones frecuentes con archivos y carpetas: trabajando con rutas, recorriendo recursivamente archivos y carpetas, leyendo atributos de los archivos, copiar, renombrar y eliminar archivos.
- Módulos, importación absoluta y relativa, PYTHONPATH, paquetes, gestión de paquetes con pip y manejo de dependencias separadas
- Ejercicios: validador de contraseñas, calificador de la seguridad de contraseñas, detección de archivos duplicados utilizando hashes, generación aleatoria de contraseñas basada en diccionarios.

SEMANA 05

- Revisión de conceptos de redes TCP/IP y protocolos relacionados
- Programación de sockets TCP/IP en Python
- Manejo de múltiples conexiones en paralelo utilizando hilos
- Construcción e inyección de paquetes a bajo nivel
- Análisis de tráfico de red con Scapy
- Scaneo de puertos con Python