

Conceptos importantes en Python relacionados con procesos, hilos, red y seguridad

Procesos

- `subprocess.Popen`: Clase que permite crear y controlar procesos en Python.

[crayon-6a9d57ca09509888660678/]

- `os.system`: Permite interactuar con el entorno de ejecución y controlar procesos en tiempo de ejecución.

[crayon-6a9d57ca0950f332163624/]

- `subprocess.run`: Representa un proceso en ejecución y proporciona métodos para controlarlo.

[crayon-6a9d57ca09511633068857/]

- `psutil`: Obtener información sobre procesos en ejecución y controlarlos.

[crayon-6a9d57ca09513129057937/]

Hilos

- `threading.Thread`: Clase que permite crear y controlar

hilos en Python.

[crayon-6a9d57ca09515848744106/]

- `threading.Thread` con `target`: Interfaz funcional que define un hilo de ejecución en Python.

[crayon-6a9d57ca09516651014958/]

- `asyncio`: Mecanismo de concurrencia que permite realizar tareas asíncronas y no bloqueantes en Python.

[crayon-6a9d57ca09518346979264/]

- `concurrent.futures.ThreadPoolExecutor`: Esta interfaz permite administrar hilos y ejecutar tareas de manera concurrente.

[crayon-6a9d57ca0951a706552264/]

- `asyncio.Future`: Clase que facilita la programación asíncrona y la combinación de múltiples futuros en Python.

[crayon-6a9d57ca0951b759912401/]

Red

- `urllib.request.urlopen`: Clase que permite trabajar con URL y realizar operaciones de red como abrir conexiones HTTP.

[crayon-6a9d57ca0951d211458145/]

- `http.client.HTTPConnection`: Clase abstracta que establece una conexión a un recurso a través de una URL.

[crayon-6a9d57ca09522908978745/]

- `socket.socket`: Clase que representa un socket de red y permite la comunicación entre procesos a través de la red.

[crayon-6a9d57ca09524536235297/]

- `selectors.DefaultSelector`: Una clase para operaciones de socket no bloqueantes en Python.

[crayon-6a9d57ca09525897541427/]

- `socket.socket` con `SOCK_DGRAM`: Para trabajar con datagramas y sockets UDP en Python.

[crayon-6a9d57ca09527044971255/]

Seguridad

- `cryptography.Fernet`: Clase que proporciona funcionalidad de cifrado y descifrado para encriptar y desencriptar datos.

[crayon-6a9d57ca09529528824318/]

- `secrets`: Clase que genera números aleatorios criptográficamente seguros.

[crayon-6a9d57ca0952b833086386/]

- `cryptography.hazmat.primitives.serialization.load_pem_private_key`: Clase que almacena claves criptográficas y certificados de seguridad.

[crayon-6a9d57ca0952c584348046/]

- `cryptography.hazmat.primitives.asymmetric.rsa.RSAPrivateKey`: Para generar pares de claves criptográficas (pública y privada) en Python.

[crayon-6a9d57ca0952e130456057/]

- `hashlib.sha256`: Clase que proporciona funciones de resumen criptográfico, como SHA-256 o MD5, para procesar datos.

[crayon-6a9d57ca09530910739646/]

- `ssl.create_default_context`: Para configurar conexiones seguras con el protocolo SSL/TLS en Python.

[crayon-6a9d57ca09531417094982/]