

Python, avanzado

Curso práctico de 4 días - 28h

Ref.: PYA - Precio 2025: 1 700€ sin IVA

El lenguaje Python se ha convertido en la base tecnológica para el desarrollo de grandes proyectos de software. En este curso aprenderás las técnicas avanzadas del lenguaje Python y sus principales bibliotecas, para que puedas cumplir los requisitos de calidad de estos proyectos.

OBJETIVOS PEDAGÓGICOS

Al término de la formación, el alumno podrá:

Aplicación rigurosa de patrones de diseño reconocidos

Utilizar las técnicas avanzadas del lenguaje Python: gestor de contexto, metaclasses, cierres, funciones avanzadas

Optimice el rendimiento de sus programas mediante la supervisión y el paralelismo

Empaquetado y despliegue de artefactos Python

Utilizar bibliotecas que contribuyan al éxito del lenguaje: computación científica, Inteligencia Artificial, XML, redes, etc.

MÉTODOS PEDAGÓGICOS

Aportación teórica ilustrada mediante ejemplos de código, luego consolidada mediante la realización de un miniproyecto (70% de trabajo práctico).

PROGRAMA

última actualización: 04/2024

1) Recordatorios importantes sobre el lenguaje

- Asignación por referencia y tipos de datos modificables y no modificables (mutables).
- Pasar argumentos, valores por defecto y variables locales.
- Variables de clase y de instancia.
- Rebanadas y estructuras de datos avanzadas.
- Introspección.
- Elementos avanzados de las estructuras de control: la cláusula else en las sentencias for, while y try/except.

Trabajo práctico : Optimización: intersección de listas y cálculo de la complejidad del algoritmo.

2) Funciones avanzadas

- Uso avanzado de decoradores (de generación a consumo, pipeline de consumo).
- Decoradores y patrones de diseño.
- Cierre.

Trabajo práctico : Encadenamiento de consumidores de datos. Suscripción a eventos mediante decoradores.

3) Programación avanzada orientada a objetos

- Propiedades.
- Iteradores.
- El patrimonio múltiple y sus deficiencias.
- Gestores de contexto.

PARTICIPANTES

Ingenieros y desarrolladores.

REQUISITOS PREVIOS

Buenos conocimientos de desarrollo en Python, o conocimientos equivalentes a los adquiridos en el curso PYT. Experiencia requerida.

COMPETENCIAS DEL FORMADOR

Los expertos que imparten la formación son especialistas en las materias tratadas. Han sido validados por nuestros equipos pedagógicos, tanto en el plano de los conocimientos profesionales como en el de la pedagogía, para cada curso que imparten. Cuentan al menos con entre cinco y diez años de experiencia en su área y ocupan o han ocupado puestos de responsabilidad en empresas.

MODALIDADES DE EVALUACIÓN

El formador evalúa los progresos pedagógicos del participante a lo largo de toda la formación mediante preguntas de opción múltiple, escenificaciones de situaciones, trabajos prácticos, etc. El participante también completará una prueba de posicionamiento previo y posterior para validar las competencias adquiridas.

MEDIOS PEDAGÓGICOS Y TÉCNICOS

- Los medios pedagógicos y los métodos de enseñanza utilizados son principalmente: ayudas audiovisuales, documentación y soporte de cursos, ejercicios prácticos de aplicación y ejercicios corregidos para los cursillos prácticos, estudios de casos o presentación de casos reales para los seminarios de formación.
- Al final de cada cursillo o seminario, ORSYS facilita a los participantes un cuestionario de evaluación del curso que analizarán luego nuestros equipos pedagógicos.
- Al final de la formación se entrega una hoja de presencia por cada media jornada de presencia, así como un certificado de fin de formación si el alumno ha asistido a la totalidad de la sesión.

MODALIDADES Y PLAZOS DE ACCESO

La inscripción debe estar finalizada 24 horas antes del inicio de la formación.

ACCESIBILIDAD DE LAS PERSONAS CON DISCAPACIDAD

¿Tiene alguna necesidad específica de accesibilidad? Póngase en contacto con la Sra. FOSSE, interlocutora sobre discapacidad, en la siguiente dirección psh-accueil@orsys.fr para estudiar de la mejor forma posible su solicitud y su viabilidad.

- Clases y métodos abstractos (ABC).

- Metaclasses.

Trabajo práctico : Implementa una metaclasses para crear clases singleton.

4) Despliegue y calidad

- Instalación de bibliotecas de terceros (pip, easy_install).

- El índice de paquetes de Python (PyPI).

- Empaquetando tus librerías (distutils, setuptools).

- Despliegue de un entorno autónomo (virtualenv y buildout).

Trabajo práctico : Empaqueta una biblioteca y súbela a Pypl.

5) Paralelismo: optimice el rendimiento de sus programas

- Perfíle sus programas con Timeit y cProfile.

- Paralelización: evite el multithreading y apueste por el multiprocessing.

- Cálculo distribuido con la biblioteca Celery.

Trabajo práctico : Distribución y consolidación (Map Reduce) de cálculos con Celery.

6) Bibliotecas que contribuyen al éxito de la lengua

- Computación científica y estadística con Numpy, Scipy, Matplotlib y Pandas.

- Inteligencia artificial y algoritmos de aprendizaje con Scikit-Learn.

- Búsqueda de información en archivos XML con ElementTree.

- Red: tcp relay con Twisted y supervisión SNMP con PySNMP.

Trabajo práctico : Extracción de información de archivos de registro XML, filtros y estadísticas de los datos recogidos y, a continuación, representación gráfica de las tendencias de la información.

FECHAS

Contacto