

Python, avanzado

Curso práctico de 4 días - 28h

Ref.: PYA - Precio 2024: 1 670€ sin IVA

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

OBJETIVOS PEDAGÓGICOS

Al término de la formación, el alumno podrá:

Aplicar de manera rigurosa patrones de diseño reconocidos

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

Optimizar el rendimiento de sus programas mediante la monitorización y el paralelismo

Empaquetar y desplegar sus artefactos de Python

Usar bibliotecas que contribuyen al éxito del lenguaje: computación científica, inteligencia artificial, XML, red

MÉTODOS PEDAGÓGICOS

Aportaciones teóricas ilustradas con ejemplos de código y posteriormente consolidadas con la realización de un miniproyecto (70 % de ejercicios prácticos).

PROGRAMA

última actualización: 01/2023

1) Recordatorios importantes sobre el lenguaje

- Asignación por referencia y tipo de datos modificables, no modificables (mutable).
- Trasladar argumentos, valores predeterminados y variables locales.
- Variables de clase y de instancias.
- Cortes y estructuras de datos avanzadas.
- Introspección.
- Elementos avanzados de las estructuras de control: la cláusula else de las instrucciones for, while, try/except.

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

2) Funciones avanzadas

- Uso avanzado de decoradores (desde la generación hasta el consumo, pasando por la cadena de consumidores).
- Decoradores y patrones de diseño.
- Cierre/closure.

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

3) Programación orientada a objetos avanzada

- Las propiedades (property).
- Los iteradores.
- La herencia múltiple y sus inconvenientes.

PARTICIPANTES

Ingenieros y desarrolladores.

REQUISITOS PREVIOS

Buenos conocimientos de desarrollo Python o conocimientos equivalentes a los aportados por los cursos de THO o PYT. Se requiere experiencia.

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.

- Gestores de contexto (Context Managers).
- Clases y métodos abstractos (ABC).
- Metaclases.

Trabajo práctico : Implementación de una metaclassa para crear clases singleton.

4) Despliegue y calidad

- Instalación de bibliotecas de terceros (pip, easy_install).
- El índice de paquetes de Python (PyPI).
- Empaquetar sus bibliotecas (distutils, setuptools).
- Despliegue de un entorno independiente (virtualenv y buildout).

Trabajo práctico : Empaquetar una biblioteca y colocarla en Pypl.

5) Paralelismo: optimización del rendimiento de sus programas

- Profile sus programas con Timeit y cProfile.
- Paralelismo: evite el multihilo y apueste por el multiprocesamiento.
- Computación distribuida con la biblioteca Celery.

Trabajo práctico : Distribuir y consolidar (Map Reduce) los cálculos con Celery.

6) Las bibliotecas contribuyen al éxito del lenguaje

- 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.
- Redes: retransmisión tcp con Twisted y supervisión SNMP con PySNMP.

Trabajo práctico : Extracción de información en archivos de registro XML, filtros y estadísticas sobre los datos recopilados y representación de las tendencias de la información mediante gráficos.

FECHAS

Contacto