

Python, Programación de objetos

Curso práctico de 5 días - 35h

Ref.: PYT - Precio 2024: 2 070€ sin IVA

Python es un lenguaje de programación multiplataforma que permite el desarrollo de una amplia variedad de aplicaciones. Dominará su sintaxis, sus principales mecanismos y su paradigma Objeto. Descubrirá las funciones de la biblioteca de módulos estándar, implementará interfaces gráficas, accederá a los datos de una base y utilizará herramientas para probar y evaluar la calidad del código de producto.

OBJETIVOS PEDAGÓGICOS

Al término de la formación, el alumno podrá:

Dominar la sintaxis del lenguaje Python

Adquirir los conceptos esenciales de la programación de objetos

Conocer y poner en práctica los diferentes módulos de Python

Diseñar interfaces gráficas

Aplicar las herramientas de prueba y evaluación de la calidad de un programa en Python

CERTIFICACIÓN

Este curso aborda las competencias evaluadas por el sistema de certificación profesional TOSA® Python, que certifica las competencias del alumno en una escala de 1000 puntos durante un periodo de 3 años. La suscripción a la opción de certificación se realiza en el momento de la inscripción y compromete al alumno a realizar el examen en línea en las 4 semanas siguientes al inicio del curso. El examen dura 1 hora y 30 minutos, y consiste en cuestionario de 35 preguntas de respuesta múltiple, cuya dificultad se adapta según las respuestas del alumno. A menos que se solicite expresamente, se imparte por defecto en francés. La supervisión se realiza mediante un programa informático y se registra con fines de control de conformidad. Una vez realizado el examen, el alumno puede consultar sus resultados en línea y recibir un certificado por correo electrónico, un resumen detallado de sus competencias y su diploma.

PROGRAMA

última actualización: 01/2023

1) Sintaxis del lenguaje Python

- Identificadores y referencias. Convenciones de codificación y reglas de denominación.
- Bloques, comentarios.
- Tipos de datos disponibles.
- Variables, visualización con formato, ámbito local y global.
- Manejo de tipos numéricos, manejo de cadenas de caracteres.
- Manejo de cuadros dinámicos (lista), cuadros estáticos (tupla) y diccionarios.
- El uso de archivos.
- La estructura condicional if/elif/else.
- Operadores lógicos y operadores de comparación.
- Los bucles de iteración while y for. Interrupción de iteraciones break/continue.

PARTICIPANTES

Desarrolladores, ingenieros y gestores de proyectos cercanos al desarrollo.

REQUISITOS PREVIOS

Conocimientos básicos de programación.

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.

- La función de range.
- Escritura y documentación de funciones.
- Expresiones lambda.
- Generadores.
- La estructuración del código en módulos.

Trabajo práctico : Instalación y manejo del intérprete de Python.

2) Enfoque orientado a objetos

- Los principios del paradigma Objeto.
- La definición de un objeto (estado, comportamiento, identidad).
- El concepto de clase, atributos y métodos.
- Encapsulación de datos.
- Comunicación entre los objetos.
- Herencia, transmisión de las características de una clase.
- El concepto de polimorfismo.
- Asociación entre clases.
- Interfaces.
- Presentación de UML.
- Diagramas de clases, de secuencias, de actividad...
- Concepto de patrón de diseño (Design Pattern).

Trabajo práctico : Modelización de un caso práctico sencillo en UML.

3) Programación de objetos en Python

- Las particularidades del modelo de objetos de Python.
- La escritura de clases y su instanciación.
- Constructores y destructores.
- Protección del acceso a atributos y métodos.
- La necesidad del parámetro Self.
- Herencia simple, herencia múltiple, polimorfismo.
- Los conceptos de visibilidad.
- Métodos especiales.
- Introspección.
- Implementación de interfaces.
- Buenas prácticas y patrones de diseño comunes.
- El uso del mecanismo de excepción para la gestión de errores.

Trabajo práctico : Práctica de los diferentes conceptos de Objeto a través de la aplicación del caso de estudio.

4) Uso de StdLib

- Los argumentos pasados a la línea de comandos.
- Uso del motor de expresiones regulares Python con el módulo «re», caracteres especiales, cardinalidades.
- Manejo del sistema de archivos.
- Presentación de algunos módulos importantes de la biblioteca estándar: módulo «sys», «os», «os.path».
- Empaquetado e instalación de una biblioteca de Python.
- El acceso a bases de datos relacionales, el funcionamiento de la DB API.

Trabajo práctico : Implementación de módulos Python: expresiones regulares, acceso a una base de datos,

5) Herramientas de control de calidad

- Herramientas de análisis estático de código (Pylint, Pychecker).
- Estudio de los informes de análisis (tipos de mensajes, advertencias, errores).
- Extracción automática de documentación.
- Depurador de Python (ejecución paso a paso y análisis post-mortem).
- Desarrollo basado en pruebas.
- Módulos de pruebas unitarias de Python (Unittest...).

- Automatización de pruebas, agregación de pruebas.
 - Pruebas de cobertura de código y elaboración de perfiles.
- Trabajo práctico : Uso de herramientas pylint y pychecker para verificar un código Python. Realización de pruebas unitarias.*

6) Creación de la GUI TkInter

- Principios de programación de las interfaces gráficas.
- Presentación de la biblioteca TkInter.
- Los contenedores principales.
- Presentación de los widgets disponibles (Button, Radiobutton, Entry, Label, Listbox, Canvas, Menu, Scrollbar, Text...).
- El administrador de ventanas.
- La colocación de los componentes, las diferentes disposiciones.
- Gestión de eventos, el objeto de eventos.
- Aplicaciones multiventana.

Trabajo práctico : Diseño de una interfaz gráfica con la biblioteca Tkinter.

7) Interfaz Python/C

- Presentación del módulo Ctypes.
- Carga de una biblioteca C.
- Llamada a una función.
- Reescribir una función de Python en C con la API de Python/C.
- Creación de módulos C para Python.
- El intérprete de Python en C.
- Uso del perfilador de código.

Trabajo práctico : Llamada a funciones escritas en C desde Python. Creación de módulos C para Python con Pyrex.

8) Conclusión

- Análisis crítico de Python.
- La evolución del lenguaje.
- Elementos webgráficos y bibliográficos.

FECHAS

Contacto