

Fundamentos de Python 2

Alcance y Secuencia

Versión 1.0

Contenido

Audiencia	3
Prerrequisitos	3
Certificación alineada	3
Objetivos del curso	4
Requisitos de equipamiento	4
Requisitos del curso	5

Audiencia

Este es el segundo curso que te preparará para el examen de certificación **PCAP: Certified Associate in Python Programming** a través de Pearson VUE. El curso comienza donde termina Fundamentos de Python 1. El curso se recomienda para los aspirantes a desarrolladores de Software que estén interesados en seguir carreras relacionadas con el tema, la seguridad, las redes y el Internet de las cosas (IoT), así como para los estudiantes que buscan adquirir habilidades más avanzadas en Python y la programación informática: estudiantes en escuelas secundarias, escuelas vocacionales y universidades.

- Profesionales de la industria que deseen explorar tecnologías relacionadas con Python o que utilicen Python como base para proyectos más complejos.
- Líderes de equipo, gerentes de productos y de proyectos que desean comprender la terminología y los procesos en el ciclo de desarrollo de software para administrar y comunicarse con los equipos de TI, pruebas y desarrollo de manera más efectiva.
- Simplemente, cualquier persona interesada en aprender a programar por diversión o con fines relacionados al trabajo.

Prerrequisitos

Curso de Fundamentos de Python 1.

Certificación alineada

El plan de estudios de Fundamentos de Python 2 ayuda a los estudiantes a prepararse para la certificación [PCAP – Certified Associate in Python Programming](#). **PCAP™ – Certified Associate in Python Programming certification (Exam PCAP-31-0x)** es una credencial profesional de alto nivel que mide la capacidad del candidato para realizar tareas de codificación de nivel intermedio en el lenguaje Python, incluida la capacidad de diseñar, desarrollar, depurar, ejecutar y refactorizar programas Python de varios módulos, así como también mide habilidades y conocimientos relacionados con el análisis y modelado de problemas de la vida real en categorías disponibles en el enfoque de programación orientada a objetos (POO), con el uso de las nociones y técnicas fundamentales.

Descripción del curso

Su objetivo principal es enseñar las habilidades relacionadas con los aspectos más avanzados de la programación de Python, incluidos módulos, paquetes, excepciones, procesamiento de archivos, así como técnicas generales de codificación y programación orientada a objetos (POO). El curso familiariza al estudiante con el enfoque del procedimiento de la programación y cubre las siguientes áreas temáticas:

- Técnicas generales de codificación y mejores prácticas.
- Usar la programación orientada a objetos en Python.
- Importar y usar módulos de Python.
- Manejar las excepciones.
- Procesar archivos.

El curso se divide en **4 módulos**. Cada estudiante tiene acceso a materiales prácticos, cuestionarios y pruebas para obtener habilidades, interacción con tareas y situaciones de programación que se presentan en la vida real.

Objetivos del curso

Al finalizar el curso, los estudiantes serán capaces de realizar las siguientes tareas:

- Comprender la función de los módulos en Python y conocer las formas disponibles de importar módulos en su propio código/namespace.
- Adquirir conocimientos sobre módulos útiles y específicos de Python.
- Tener una orientación de los propósitos de los paquetes y ser capaz de crear sus propios paquetes.
- Conocer la función principal de PIP y ser capaz de utilizarlo para instalar y desinstalar paquetes listos para usar de PyPI.
- Saber cómo se codifican y almacenan los caracteres dentro de la memoria de la computadora, y distinguir los estándares de codificación más conocidos.
- Obtener conocimiento de las secuencias de Python y conocer las diferencias entre cadena y listas.
- Ser capaz de utilizar de forma efectiva, listas y métodos de cadenas.
- Tener una orientación de la forma en que Python identifica y maneja los errores de tiempo de ejecución; comprender el propósito de las sentencias de control try, except y raise; comprender las jerarquías de excepción en Python.
- Comprender los conceptos fundamentales de la programación de objetos como clase, objeto, propiedad, método, herencia y polimorfismo.
- Tener una orientación en las diferencias entre los enfoques procedimentales y de objetos, así como estar orientado sobre ambas técnicas con sus pros y contras.
- Ser capaz de construir sus propias clases, objetos, propiedades y métodos; ser capaz de usar herencia y polimorfismo en su ruta de herencia; comprender la naturaleza objetiva de las excepciones de Python.
- Obtener la capacidad de comprender los conceptos de generadores, iterados y cierres, así como poder usarlos en aplicaciones adecuadas.
- Saber cómo Python accede a los recursos del sistema de archivos físico, comprender los modos de apertura de archivos y realizar operaciones básicas de entrada/salida en relación con archivos de texto y binarios.
- Adquirir la capacidad de manipular la fecha y la hora, trabajar con un calendario y crear estructuras de directorios utilizando Python.
- Estar listo para tomar la prueba final del curso.
- Estar preparado para intentar obtener la certificación **PCAP- Certified Associate in Python Programming de OpenEDG Python Institute**.

Requisitos de equipamiento

Se puede acceder al curso en línea a través de cualquier navegador de Internet. El único requisito previo es una conexión a Internet activa, una computadora de escritorio o portátil/ tableta/ dispositivo móvil.

Para obtener la mayor experiencia de aprendizaje, recomendamos utilizar las versiones más recientes de Mozilla Firefox, Internet Explorer/ Microsoft Edge o Google Chrome.

Requisitos del curso

La siguiente tabla detalla los módulos del curso. Cada módulo es una unidad integrada de aprendizaje que consta de contenido, actividades y evaluaciones que apuntan a un conjunto específico de competencias. El tamaño del módulo depende de la profundidad del conocimiento y la habilidad necesaria para dominar la competencia.

Módulo	Descripción
Módulo 1 Módulos, Paquetes y PIP	<ul style="list-style-type: none"> • ¿Qué es un módulo y para qué lo necesitamos? • Importación y uso de módulos. • Revisión de algunos módulos útiles de Python. • ¿Qué es un paquete y en qué se diferencia de un módulo? • Construcción y uso de paquetes. • PIP: La navaja suiza y el mantenimiento de paquetes.
Módulo 2 Caracteres, cadenas y excepciones avanzadas	<ul style="list-style-type: none"> • Caracteres, cadenas y estándares de codificación. • Cadenas y listas: similitudes y diferencias. • Métodos de listas. • Métodos de cadenas. • La forma en que Python maneja los errores en tiempo de ejecución • Controlar el flujo de errores usando try y except. • La jerarquía de excepciones.
Módulo 3 Programación orientada a objetos en Python	<ul style="list-style-type: none"> • Conceptos básicos de programación orientada a objetos. • Del enfoque procedimental al de objetos: motivaciones y beneficios. • Clases, objetos, propiedades, y métodos. • Herencia y polimorfismo. • Excepción como objeto.
Módulo 4 Trabajar con archivos y utilidades	<ul style="list-style-type: none"> • Generadores, iteradores y cierres. • Trabajar con sistemas de archivos, árboles de directorios y archivos. • Módulos seleccionados de la biblioteca estándar de Python (os, date, datetime, calendar).