

# Python Avanzado



40 horas

## OBJETIVOS

---

Conocer todas las posibilidades que te brinda este lenguaje: sintaxis y características del lenguaje Python

- Construir aplicaciones.
- Conocer las funciones avanzadas.
- Adquirir los conocimientos de la programación orientada a objetos.
- Configurar multitareas.

## DESTINATARIOS

---

Programadores en cualquier lenguaje orientado a objetos.

## METODOLOGÍA

---

Formación impartida en modalidad online que combinará la presentación de contenidos teóricos con la realización de ejercicios prácticos.

## PLAZO Y REQUISITOS PARA SUPERAR LA FORMACIÓN

---

En el plazo que se indique en la plataforma (alrededor de 8 **semanas**) se deberán cumplir los siguientes requisitos para superar el curso:

- Visualización del 75% del contenido.
- Conectarse al menos al 30% de las horas del curso.
- Superar el 100% de los controles periódicos.

# CONTENIDOS

## Python vs PHP

- Ventajas y desventajas.
- Python3 vs Python2.
- Facilidad para el desarrollo de “middleware”.
- Comparativa de rendimiento de Python con PHP.

## Documentación y recursos.

- Webs de referencia.
- Lecturas recomendadas.
- Repositorios GitHub Gestor de paquetes de Python.
- Entornos virtuales.

## Orientación a objetos

- Orientación básica a objetos.
- Servidor web/TCP Orientado a objetos vs no orientado.

## Acceso a Bases de Datos desde Python

- Pasando de MongoDB con PHP Doctrine a Python.
- Elastic search.
- Redis.
- MariaDB – PostgreSQL.
- ORMS:
  - SQLAlchemy
  - Django

## Herramientas de Desarrollo

- Netbeans, Eclipse, PyCharm, etc.

## Frameworks

- Django.
- Modelos.
- Vistas.
- Controladores.
- Configuración.
- Tareas de línea de comandos.

## Control de versiones

- Git.
- Migración desde SVN.
- Mercurial.

## Procesos que permanezcan a la escucha en un determinado puerto del servidor

- SupervisorD
- Cron
- Async WebTasks:
  - RabbitMQ
  - Python-RQ

## Documentación del software y tests

- Test de unidad:
  - En los comentarios
  - Módulo unittest
- Mockups:
  - Unittest.mock

## Concurrencia. Funcionamiento y rendimiento

- Diferencias entre librerías y funcionamiento Gevent
- Multiprocesin.

## Penalización de las llamadas vía API

- Llamadas a APIs Python desde PHP vs Llamadas PHP directas.
- Cliente / Servidor REST:
  - Cliente puro.
  - Django REST framework.
- Cliente / Servidor SOAP.

## Expresiones regulares

- Librería RE.
- Extraer datos con PyQuery.

## Librerías de conectividad

- SSH
- Telnet
- FTP
- SMTP
- Scapy
- HTTP Automatizar relleno de formularios y demás Logging y debugg.