

## Advanced Python Mastery (PYTHAV)

- **Formato do curso:** Presencial e Live Training
- **Localidade:** Lisboa
- **Data:** 11 Mar. 2019 a 14 Mar. 2019
- **Preço:** 1610€
- **Promoção:** -10%
- **Horário:** Laboral - das 09h30 às 16h30
- **Nível:** Avançado
- **Duração:** 28 horas

Aprende Python num tutorial online, num curso inicial de formação ou num livro e deseja aprender mais? Então este é o curso indicado.

O curso começa com a análise de aspetos subtis do código Python que está habituado a usar, mas rapidamente passa a aprofundar uma ampla variedade de tópicos avançados, incluindo object model, data encapsulation, descriptors, generators, coroutines, context managers, decorators, metaclasses, packages, closures e muito mais.

No final do curso, o formando não só saberá quais são esses recursos, mas como eles podem ser aplicados a uma ampla gama de problemas práticos de programação.

---

### Destinatários

- Desenhado para programadores que desejam elevar seu conhecimento a um nível totalmente novo, este curso destina-se a todos aqueles que querem aprender as potencialidades do Python.

---

### Pré-requisitos

- Conhecimentos de Inglês
  - Conhecimentos de Programação orientada a objetos, de preferência em Python
-

# Objectivos

No final da ação de formação os participantes deverão estar aptos a:

- Aplicar as boas práticas de escrita de código;
  - Criar classes e utilizar objetos de forma efetiva
  - Utilizar as propriedades dos objetos para criar “código dinâmico”.
  - Criar testes unitários para validar o desenvolvimento;
  - Fazer debugging e log de forma correta.
  - Criar pacotes de instalação.
  - Utilizar metaclasses e lambda expressions.
  - Iterar correctamente.
  - Estender funcionalidade de c/c++
- 

## Programa

### **Python Review**

- Revisão de conceitos introdutórios
- Python nos dias de hoje.

### **Idiomatic Data Handling**

- Comparação de operadores em cadeia;
- Identação
- Conceitos de Falsy & Truthy
- O equivalente ao operador ternário
- A keyword “in”
- O uso de return na avaliação de expressões;
- Formatação de strings;
- Comprehension

### **Classes and Objects**

- Criação de Classes;
- Private vs Public;
- Herança;
- Métodos estáticos.

### **Inside Python Objects**

- Serialization;
- Deserializaion.

### **Testing, Logging, and Debugging**

- Criação de testes unitários;

- Log e debugging

## **Packaging and Distributing Python Programs**

- Criação de pacotes de distribuição;
- Criação de exe.

## **Working with Functions**

- Scope in functions

## **Metaprogramming**

- Metaclasses;
- Código a gerar código.

## **Iterators, Generators, and Coroutines**

- O método iter()
- Looping sobre diferentes estruturas.
- Como criar iteradores;
- Geradores
- O método `__next_()`:

## **Extending Python with C and C++**

- Estender o uso de c/c++
- Criar programas que utilizar funcionalidades existentes em c/c++