

# Curso Introdutório de Python

Ementa desenvolvida por Álvaro Justen <[alvaro@justen.eng.br](mailto:alvaro@justen.eng.br)>

## 1 Introdução

Nesse curso serão abordados profundamente os fundamentos da linguagem de programação Python, tais como:

- Sintaxe
- Principais tipos e estruturas de dados
- Funções *built-in*
- Módulos da biblioteca padrão
- Acesso a arquivos e banco de dados
- Orientação a objetos
- Python trabalhando com GUI, Web e gráficos
- Qualidade em desenvolvimento Python

## 2 Metodologia

A metodologia de ensino será baseada em exemplos reais, com exercícios práticos (durante as aulas) guiados pelos novos conceitos. Além disso, serão enviados exercícios para que os alunos resolvam em casa, será feita a correção desses exercícios (explicitando pontos onde o aluno poderia melhorar o código) e serão sanadas dúvidas via lista de discussão a ser criada para a turma.

A lista de discussão também será utilizada para compilação dos conceitos abordados em sala de aula, com envio de links que permitam aos alunos aprofundarem os conceitos e conhecerem assuntos correlatos.

Ao final do curso serão apresentadas aos alunos as comunidades e eventos relacionados a Python no estado, país e no mundo, ressaltando as principais conferências e sites que podem ser úteis para sanar dúvidas e estender o conhecimento nesse domínio.

## 3 Conteúdo Programático

**Carga horária total: 21 horas.**

### 3.1 Conhecendo a Linguagem

#### 3.1.1 Introdução

**Carga horária: 1 hora.**

- O que é Python?
  - Linguagem interpretada
  - Tipagem dinâmica e forte
  - Blocos de código delimitados por indentação
  - Paradigmas: procedural, funcional e orientada a objetos
  - Propósito geral: scripts, jogos, sites, interfaces
  - Exemplos de uso (softwares e empresas)
  - The Zen of Python e as Python Enhancement Proposals
- Instalando o interpretador

- Scripts *versus* interpretador interativo
  - Python hashbang
  - python, ipython, bpython

### 3.1.2 Tipos de Dados e Estruturas de Dados

**Carga horária: 3 horas.**

- int
- float
- str
  - Formatação
- list
  - *List comprehension* e *generators*
- tuple
- dict
- *Slicing* de strings, tuplas e listas

### 3.1.3 Controle de Fluxo

**Carga horária: 3 horas.**

- if
- elif
- else
- for (iterativo)
- while
- Tratamento de exceções

### 3.1.4 Funções

**Carga horária: 3 horas.**

- O que são e como defini-las
- Retornando valores
- Argumentos
  - Nomeados
  - Valores pré-definidos
  - Indefinidos
- Escopo de variáveis
- Algumas funções *built-in* úteis
  - `help()` e `dir()`
  - `range()` e `enumerate()`
  - `len()` e `type()`
  - `exit()`
  - `min()` e `max()`
  - `all()` e `any()`
  - `map()`, `reduce()`, `zip()` e `join()`
- Criando funções anônimas com `lambda`
- Transformando funções com *decorators*

## 3.2 A Biblioteca Padrão e seus Módulos

**Carga horária: 2 horas.**

- Importando módulos
- *Paths* e *namespaces*
- Estrutura e criação
- Gerenciamento de pacotes

## 3.3 Orientação a Objetos

**Carga horária: 3 horas.**

- Classes e instâncias
- Atributos
- Métodos
- Herança e Introspecção

## 3.4 Trabalhando com Arquivos

**Carga horária: 2 horas.**

- A função `open()`
- Utilizando o protocolo HTTP com a biblioteca `urllib`
- Conectando a servidores FTP com a biblioteca `ftplib`
- Enviando e recebendo emails com as bibliotecas `smtplib` e `imaplib`

## 3.5 Utilizando Python no dia-a-dia

**Carga horária: 3 horas.**

- Frameworks Web
- Bibliotecas para criação de GUI
- Ferramentas e bibliotecas para trabalhar com gráficos
- Os módulos `pickle` e `cPickle`
- Conectando a bancos de dados com *Object-Relational Mappers*

## 3.6 Escrevendo Software de Qualidade com Python

**Carga horária: 1 hora.**

- O que são testes de software
  - *Test-Driven Development*
  - Os módulos `doctest` e `unittest`
- Documentação com *docstrings*