



PYTHON
ACADEMY

DATACLASSES NO PYTHON

Nesse ebook você vai aprender a utilizar dataclasses no Python para criar códigos mais limpos e enxutos.

[PYTHONACADEMY.COM.BR](https://pythonacademy.com.br)

Este ebook foi gerado por



Crie Ebooks técnicos incríveis em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs

Deixe que nossa IA faça o trabalho pesado

 Syntax Highlight

 Adicione Banners Promocionais

 Edite em Markdown em Tempo Real

 Infográficos feitos por IA

TESTE AGORA 

 PRIMEIRO CAPÍTULO 100% GRÁTIS

Salve salve Pythonista!

Neste artigo, vamos falar sobre as **dataclasses** no Python, uma funcionalidade introduzida na versão 3.7 da linguagem.

As dataclasses são uma forma simplificada de criar classes que armazenam dados, proporcionando uma maneira fácil e rápida de implementar classes com atributos e métodos específicos para manipulação de dados.

Vamos ver sua utilidade e como podemos definí-las no Python.

Então vamos nessa!

```
<div class="suggestions-header">
  <h2>Leia também</h2>
</div>
<div class="suggestions-body">
  <ul>
    <li>
      <a href="https://pythonacademy.com.br/blog/introducao-a-programacao-orientada-a-objetos-no-python" target="_blank">
        Introdução à Programação Orientada a Objetos no Python
      </a>
    </li>
    <li>
      <a href="https://pythonacademy.com.br/blog/classes-e-objetos-no-python" target="_blank">
        Classes e Objetos no Python
      </a>
    </li>
  </ul>
</div>
```

Por que utilizar *dataclasses*?

É comum em Python utilizarmos classes para representar estruturas de dados, como por exemplo, um usuário com nome, idade e email.

Para criar essa classe, precisamos definir os atributos, o método construtor, métodos e outras funcionalidades específicas.

Isso pode ser um pouco trabalhoso e propenso a erros.

As **dataclasses** surgem como uma solução para tornar esse processo de criação de classes para armazenamento de dados mais simples, reduzindo a quantidade de código necessário.

Com as dataclasses, podemos criar essas classes de forma mais declarativa e legível, concentrando apenas na definição dos atributos e deixando o Python gerar o código necessário automaticamente.

Como utilizar *dataclasses*

Para usar as dataclasses, precisamos importar o módulo `dataclasses`.

Vamos começar criando uma classe simples que representa um usuário:

```
import dataclasses

@dataclasses.dataclass
class Usuario:
    nome: str
    idade: int
    email: str
```

Essa classe é decorada com o `@dataclasses.dataclass`, indicando que queremos utilizar a funcionalidade de `dataclass` do Python.

Em seguida, declaramos os atributos da classe, junto com os seus tipos.

A partir desse momento, o Python já faz automaticamente algumas coisas para nós:

- Cria um método construtor que recebe os valores dos atributos e atribui esses valores aos atributos correspondentes;
- Cria métodos *getters* e *setters* para cada atributo;
- Implementa os métodos especiais `__repr__` e `__eq__` para exibição amigável do objeto e comparação de igualdade, respectivamente.

Agora, podemos criar objetos dessa classe e acessar seus atributos da seguinte forma, assim como fazemos com classes comuns:

```
usuario1 = Usuario("João", 25, "joao@exemplo.com")

print(usuario1.nome)
print(usuario1.idade)
print(usuario1.email)
```

Ao executar esse código, teremos a seguinte saída:

```
João
25
joao@exemplo.com
```

Podemos também modificar os valores dos atributos:

```
usuario1.nome = "Maria"  
usuario1.idade = 30  
usuario1.email = "maria@exemplo.com"  
  
print(usuario1.nome)  
print(usuario1.idade)  
print(usuario1.email)
```

Agora teremos a seguinte saída:

```
Maria  
30  
maria@exemplo.com
```

Como podemos perceber, o Python nos permite trabalhar facilmente com as data-classes, tanto na criação quanto na modificação dos valores dos atributos.



*Estou desenvolvendo o **DevBook**, uma plataforma que usa IA para gerar ebooks técnicos profissionais. Depois de ler, dá uma passada no site!*

Crie Ebooks técnicos incríveis em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs

Deixe que nossa IA faça o trabalho pesado

Syntax Highlight

Adicione Banners Promocionais

Edite em Markdown em Tempo Real

Infográficos feitos por IA

TESTE AGORA! PRIMEIRO CAPÍTULO 100% GRÁTIS

Atributos opcionais e valores padrão

Em algumas situações, pode ser necessário definir atributos opcionais em nossas dataclasses.

Para isso, basta definir o valor padrão como `None` ou algum outro valor desejado:

```
@dataclasses.dataclass
class Usuario:
    nome: str
    idade: int = 0
    email: str = None
```

Agora podemos criar objetos da classe `Usuario` sem fornecer todos os atributos:

```
usuario1 = Usuario("João")
usuario2 = Usuario("Maria", 30)

print(usuario1.nome)
print(usuario1.idade)
print(usuario1.email)

print(usuario2.nome)
print(usuario2.idade)
print(usuario2.email)
```

A saída será:

```
João
0
None
Maria
30
None
```

Dessa forma, podemos criar objetos da classe `Usuario` com diferentes combinações de atributos obrigatórios e opcionais.

Métodos personalizados

Além de gerar automaticamente os métodos getters e setters, as dataclasses nos permitem **implementar os nossos próprios métodos personalizados**.

Esses métodos podem ser implementados normalmente, como em qualquer outra classe.

Vamos adicionar um método `saudacao` à nossa classe `Usuario`, que retorna uma saudação personalizada com nome e idade do usuário:


```
@dataclasses.dataclass
class Usuario:
    nome: str
    idade: int = 0
    email: str = None

    def saudacao(self):
        return f"Olá, meu nome é {self.nome} e tenho {self.idade} anos."
```

Agora podemos invocar esse método em um objeto da classe `Usuario` :

```
usuario1 = Usuario("João", 25)
usuario2 = Usuario("Maria", 30)

print(usuario1.saudacao())
print(usuario2.saudacao())
```

A saída será:

```
Olá, meu nome é João e tenho 25 anos.
Olá, meu nome é Maria e tenho 30 anos.
```

Podemos adicionar quantos métodos desejarmos às nossas dataclasses, tornando-as mais ricas em funcionalidades de acordo com as necessidades da nossa aplicação.

Comparação de objetos

Ao criar uma dataclass, o Python automaticamente implementa o método especial `__eq__` para realizar a comparação de igualdade entre objetos.

Esse método compara os valores dos atributos das instâncias, garantindo que dois objetos sejam considerados iguais se seus atributos tiverem os mesmos valores.

```
usuario1 = Usuario("João", 25, "joao@exemplo.com")
usuario2 = Usuario("Maria", 30, "maria@exemplo.com")
usuario3 = Usuario("João", 25, "joao@exemplo.com")

print(usuario1 == usuario2)
print(usuario1 == usuario3)
```

A saída será:

```
False
True
```

Como podemos observar, a comparação `usuario1 == usuario2` retorna `False`, pois os valores dos atributos são diferentes, enquanto a comparação `usuario1 == usuario3` retorna `True`, uma vez que os objetos têm os mesmos valores para os atributos.

Conclusão

As dataclasses são uma adição muito útil ao Python para simplificar a criação de classes de armazenamento de dados.

Com elas, podemos definir classes com atributos tipados de forma mais declarativa, e o Python se encarrega de gerar o código necessário para construir e manipular essas classes.

Neste artigo, vimos como utilizar dataclasses no Python, declarando os atributos das classes e aproveitando todas as funcionalidades geradas automaticamente.

Além disso, exploramos a criação de atributos opcionais, valores padrão, métodos personalizados e a comparação de igualdade entre objetos.

As dataclasses são uma ferramenta poderosa e muito útil para desenvolvimento em Python, tornando o código mais limpo, legível e produtivo.

Espero que este artigo tenha sido útil para você entender o que são as dataclasses no Python e como utilizá-las em suas aplicações.

Experimente utilizar dataclasses em seus projetos e desfrute de toda a praticidade e produtividade que elas oferecem.

Nos vemos no próximo Artigo aqui do Blog!

Não se esqueça de conferir!



DevBook

Crie Ebooks técnicos em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs



 Syntax Highlight

 Infográficos feitos por IA

 Adicione Banners Promocionais

Deixe que nossa IA faça o trabalho pesado

 Edite em Markdown em Tempo Real

TESTE AGORA 

 PRIMEIRO CAPÍTULO 100% GRÁTIS