



PYTHON
ACADEMY



PADRÕES DE PROJETO EM PYTHON (DESIGN PATTERNS): PROTOTYPE

Conheça o padrão criacional Prototype que possibilita realizar cópias de objetos sem fazer seu código dependente de outras classes.

Gere ebooks como este com



em <https://ebookr.ai>

Crie ebooks profissionais incríveis em minutos com IA



Chega de formatar texto no Google Docs, Word ou ferramentas que só te fazem perder tempo...

E deixe que nossa IA faça o trabalho pesado!



Capas gerados por IA



Infográficos feitos por IA



Edite em Markdown em Tempo Real



Adicione Banners Promocionais

TESTE AGORA

PRIMEIRO CAPÍTULO 100% GRÁTIS

✓ **Atualizado para Python 3.13** (Fevereiro/Março 2025) *Design Patterns modernos: tipo hints, dataclasses, quando usar e não usar.*

Salve salve Pythonista 🙌

No desenvolvimento de software, a eficiência na criação de objetos é essencial.

O **Padrão de Projeto Prototype** é uma solução poderosa para esse desafio em Python.

Neste artigo, exploraremos o que é o Prototype, o problema que ele resolve, sua estrutura, aplicabilidade e como implementá-lo na prática.

Entender este padrão pode melhorar significativamente a performance e a escalabilidade das suas aplicações Python.

O que é o Padrão de Projeto Prototype?

O **Prototype** é um padrão de criação que permite copiar objetos existentes sem depender de suas classes.

Ele facilita a criação de novas instâncias por clonagem, evitando a necessidade de repetição de código.

Essa abordagem é especialmente útil quando a criação de um objeto é complexa ou custosa em termos de recursos.

Problema que o Prototype Resolve

Criar objetos diretamente pode ser ineficiente, especialmente quando envolve processos complexos ou demorados.

Além disso, a criação direta vincula o código à classe concreta, dificultando a manutenção e a expansão.

O **Prototype** resolve esses problemas permitindo a clonagem de objetos existentes, promovendo flexibilidade e reutilização.

Racional por trás do Prototype

A ideia central do **Prototype** é utilizar objetos existentes como “protótipos” para criar novos objetos.

Isso elimina a necessidade de instanciar objetos do zero, reduzindo a sobrecarga de criação.

Além disso, promove o desacoplamento entre o código cliente e as classes concretas, facilitando futuras modificações.

Estrutura do Padrão de Projeto Prototype

A estrutura do **Prototype** consiste em:

1. **Prototype**: Interface que declara o método de clonagem.
2. **ConcretePrototype**: Implementa o método de clonagem definido na interface.

3. **Cliente:** Cria novos objetos clonando os protótipos existentes.

Essa estrutura garante que a clonagem seja realizada de maneira consistente e controlada.

Aplicabilidade do Padrão Prototype

O **Prototype** é aplicável quando:

- A criação de objetos é custosa em termos de recursos.
- É necessário criar cópias de objetos existentes.
- O sistema deve ser independente das classes que precisa para criar novos objetos.
- Os ambientes de runtime do programa devem permitir a criação dinâmica de novos objetos.

Aproveita a pausa: Estou construindo o [Ebookr.ai](https://ebookr.ai), uma plataforma que gera ebooks profissionais sobre qualquer tema usando IA. Imagina transformar seu conhecimento sobre Design Patterns em um ebook com capa, infográficos e tudo mais — em minutos! Vale conferir.

Crie Ebooks profissionais incríveis em minutos com IA



Chega de formatar texto no Google Docs, Word ou ferramentas que só te fazem perder tempo...

... e deixe que nossa IA faça o trabalho pesado!

TESTE AGORA! PRIMEIRO CAPÍTULO 100% GRÁTIS [↗](#)

Capas gerados por IA

Adicione Banners Promocionais

Edite em Markdown em Tempo Real

Infográficos feitos por IA

Implementando o Prototype na Prática

Vamos implementar o **Prototype** em Python.

Primeiro, definimos a interface `Prototype` com o método `clone` :

```
from abc import ABC, abstractmethod
import copy

class Prototype(ABC):
    @abstractmethod
    def clone(self):
        pass
```

Explicação do código:

1. **Importações:** Importamos `ABC` e `abstractmethod` para definir classes abstratas, e `copy` para clonagem.
2. **Classe Abstrata:** `Prototype` é uma classe abstrata que declara o método `clone`.
3. **Método Abstrato:** `clone` será implementado pelas classes concretas.

Exemplos Práticos de Prototype em Python

Agora, criamos uma classe concreta que implementa `Prototype`:

```
class Documento(Prototype):  
    def __init__(self, titulo, conteudo):  
        self.titulo = titulo  
        self.conteudo = conteudo  
  
    def clone(self):  
        return copy.deepcopy(self)  
  
    def __str__(self):  
        return f"Documento: {self.titulo}\nConteúdo: {self.conteudo}"
```

Explicação do código:

1. **Classe Documento:** Herda de `Prototype`.
2. **Construtor:** Recebe `titulo` e `conteudo`.
3. **Método clone:** Retorna uma cópia profunda do objeto.
4. **Método str:** Define a representação em string do objeto.

Vamos testar a clonagem:

```
# Criando o protótipo original
doc_original = Documento("Relatório Anual", "Conteúdo do relatório
                           anual.")

# Clonando o documento
doc_clone = doc_original.clone()

print(doc_original)
print(doc_clone)
```

E a saída será:

```
Documento: Relatório Anual
Conteúdo: Conteúdo do relatório anual.
Documento: Relatório Anual
Conteúdo: Conteúdo do relatório anual.
```

Explicação do código:

1. **Instância Original:** Criamos `doc_original` com título e conteúdo específicos.
2. **Clonagem:** `doc_clone` é uma cópia de `doc_original` criada pelo método `clone`.
3. **Impressão:** Exibimos ambos os documentos, que possuem os mesmos valores.

Conclusão

O **Padrão de Projeto Prototype** é uma ferramenta valiosa para otimizar a criação de objetos em Python.

Ele resolve problemas de eficiência e acoplamento, facilitando a clonagem de objetos existentes.

Com sua estrutura simples e aplicabilidade ampla, o Prototype pode melhorar a flexibilidade e a manutenção das suas aplicações.

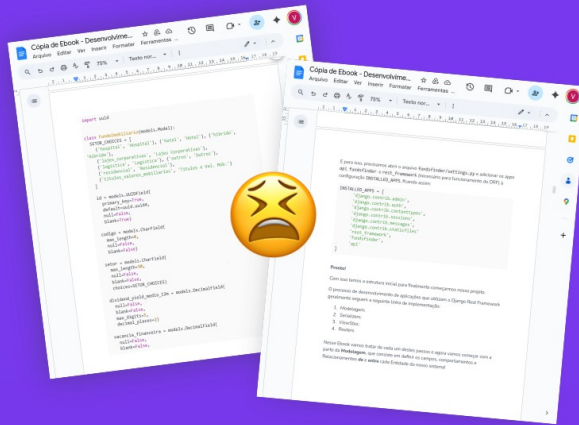
Esperamos que este artigo tenha esclarecido como implementar e utilizar o Prototype em seus projetos Python.

Não se esqueça de conferir!

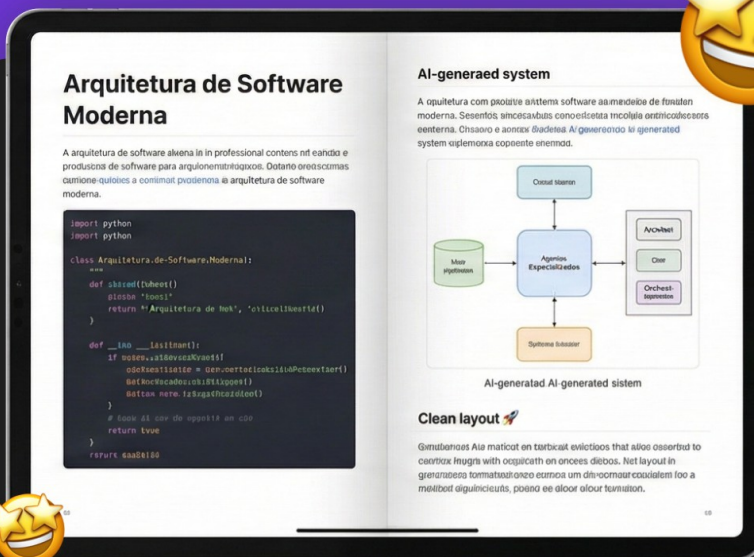


Ebookr

Crie Ebooks profissionais em minutos com IA



Chega de formatar código no Google Docs ou Word



Capas gerados por IA



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