

# Prototype em JavaScript.

Quando se há a necessidade de criar uma cópia de um objeto, o prototype é totalmente aplicável. As adições de novas subclasses podem ser acionadas a partir de um objeto prototype, assim evitando replicação de métodos e, consequentemente, diminuir o consumo de memória.

É muito lembrado a um sistema de clonagem de uma classe, pois, cópias exatas podem ser feitas a partir dele.



# Vantagens



# Desvantagens



- Clonar objetos sem acoplá-los a suas classes concretas.
- Não fazer códigos de inicialização repetidos em troca de clonar protótipos pré-construídos.
- Produção de objetos complexos mais convenientemente.
- Há a alternativa para herança quando lidar com configurações pré determinadas para objetos complexos.



- Clonar objetos complexos que têm referências circulares pode ser bem complicado.

Fonte:



<https://refactoring.guru/pt-br/design-patterns/prototype>