

Criando Tipos a Partir de Tipos

O sistema de tipos do TypeScript é muito poderoso porque permite expressar tipos *em termos de outros tipos*.

A forma mais simples dessa ideia são os genéricos. Além disso, temos uma ampla variedade de *operadores de tipo* disponíveis para uso. Também é possível expressar tipos em termos de *valores* que já temos.

Ao combinar vários operadores de tipo, podemos expressar operações e valores complexos de uma forma sucinta e mantível. Nesta seção, abordaremos maneiras de expressar um novo tipo em termos de um tipo ou valor existente.

- * [Genéricos](#) - Tipos que aceitam parâmetros
- * [Operador de Tipo Keyof](#) - Usando o operador `keyof` para criar novos tipos
- * [Operador de Tipo Typeof](#) - Usando o operador `typeof` para criar novos tipos
- * [Tipos de Acesso Indexado](#) - Usando a sintaxe `Type['a']` para acessar um subconjunto de um tipo
- * [Tipos Condicionais](#) - Tipos que agem como instruções if no sistema de tipos
- * [Tipos Mapeados](#) - Criando tipos mapeando cada propriedade em um tipo existente
- * [Tipos de Literal de Template](#) - Tipos mapeados que alteram propriedades via strings literais de template