

## Para saber mais - Validando argumentos do construtor com assert

Considerando o exemplo do `FeatureItem`, esperamos que sempre seja enviado um nome e um ícone válidos. Porém, nada o impede de enviar os seguintes valores:

```
_FeatureItem(  
  null,  
  null,  
  onClick: () => null,  
),
```

Nesta abordagem, o Widget `Text` reclama por possuir um validador em seu construtor, uma abordagem bacana, mas para `Icon`, ou `onTap`, não temos essa mesma sorte. Então, criamos um elemento visual mesmo que seu conteúdo tenha um valor indesejado:

```
_FeatureItem(  
  'test',  
  null,  
  onClick: () => null,  
),
```

Para ter a mesma validação existente no `Text`, o Dart nos oferece o `assert` que apresenta um erro caso a consideração seja falsa. No nosso exemplo, ficaria da seguinte maneira:

```
_FeatureItem(  
  this.name,  
  this.icon, {  
    @required this.onClick,  
  }) : assert(icon != null),  
      assert(onClick != null);
```

Com essa abordagem, valores nulos não são mais aceitos, portanto, mesmo que tenha argumentos obrigatórios ou não, quem usar o Widget `FeatureItem` será responsável por enviar todos os valores diferentes de nulo.