

digite literalmente no seu código-fonte. Em vez disso, o nome da propriedade que você precisa é armazenado em uma variável ou é o valor de retorno de uma função que você invocar. Você não pode usar um objeto básico literal para esse tipo de propriedade. Em vez disso, você precisa criar um objeto e depois adicionar o desejado propriedades como uma etapa extra:

```
const Property_name = "P1";  
function computePropertyName () {return "p" + 2;}  
Seja o = {};  
o [property_name] = 1;  
o [computePropertyName ()] = 2;
```

É muito mais simples configurar um objeto como este com um recurso ES6

Conhecido como propriedades computadas que permitem pegar os suportes quadrados

Do código anterior e mova -os diretamente para o objeto literal:

```
const Property_name = "P1";  
function computePropertyName () {return "p" + 2;}  
Seja p = {  
  [Property_name]: 1,  
  [ComputePropertyName ()]: 2  
};  
p.p1 + p.p2 // => 3
```

Com esta nova sintaxe, os parênteses quadrados delimitam um arbitrário

Expressão de JavaScript. Essa expressão é avaliada e o resultante

O valor (convertido em uma string, se necessário) é usado como o nome da propriedade.

Uma situação em que você pode querer usar propriedades computadas é

Quando você tem uma biblioteca de código JavaScript que espera ser passado objetos com um determinado conjunto de propriedades e os nomes daqueles