

 15

"Isso" não encaixa em um "Date"? Se vira!

JavaScript possui um objeto especial para representar datas, o objeto `Date`. Sua criação é direta:

```
let hoje = new Date();
```

Quando instanciamos `Date` sem passar qualquer parâmetro para o seu construtor, é criado um objeto com a data naquele momento. Contudo, podemos criar uma data seguindo o seguinte formato:

```
let outraData = new Date('2016/05/17'); // formato ano/mês/dia
```

Internamente, o construtor de `Date` está preparado para lidar com uma string neste formato e criar uma instância correspondente. Podemos até obter o dia, o mês e o ano de uma data usando seus métodos correspondentes:

```
let outraData = new Date('2016/05/17');
console.log(outraData.getDate()); // imprime 17
console.log(outraData.getMonth()); // imprime 4
console.log(outraData.getFullYear()); // imprime 2016
```

Não podemos esquecer que um `Date` guarda internamente os meses de 0 a 11, é por isso que `outraData.getMonth()` retorna 4 e não 5.

Podemos construir uma nova data também dessa maneira:

```
let outraDataOutraManeira = new Date(2016, 4, 17);
```

Veja que o construtor recebe o ano, mês e dia, contudo esta forma precisa receber o mês menos um e foi exatamente o que fizemos. Para o mês 5, passamos o valor 4. Aliás, estamos passando valores que são `Numbers`, mas poderíamos ter passado esses valores como string que tudo continuaria funcionando do mesmo jeito.

Parta do princípio que você tem a seguinte variável:

```
let dataString = '17-05-2016';
```

E que usará o construtor de `Date`, que está preparado para receber uma data no formato `ano/mes/dia`.

Escreva um código que altere `dataString` para que seu conteúdo fique compatível com o formato `ano/mes/dia`. No final, instancie uma `Date`, exibindo no console seu valor.