

## Faça o que eu fiz na aula

Vamos criar um arquivo chamado `validarDataNascimento.js` que será responsável por receber o valor digitado pelo usuário no campo da data de nascimento e, através deste valor, validar se o usuário é maior ou menor de 18 anos.

Bem, a primeira coisa que fazemos é criar uma função chamada `validarDataNascimento` e fazemos o `export` dela.

```
export const validarDataNascimento = input => {  
  
}
```

Agora vamos criar duas datas do tipo `Date` do Javascript. A primeira data, que vamos chamar de `dataNascimento`, será a data que o usuário digitou. Já a segunda data é a `dataAtual`, que é a data na qual o usuário está preenchendo o formulário.

```
const dataNascimento = new Date(input.value);  
const dataAtual = new Date();
```

Vamos criar uma nova variável que irá segurar a data em que o usuário faz 18 anos e vamos chamar esta variável de `dataFaz18`.

```
const dataFaz18 = new Date(  
  dataNascimento.getUTCFullYear() + 18,  
  dataNascimento.getUTCMonth(),  
  dataNascimento.getUTCDate()  
);
```

Agora, precisamos verificar se a data em que o usuário faz 18 anos é maior do que a data atual. Caso ela seja maior, iremos adicionar um erro ao `input` através do método `setCustomValidity` do próprio `input` da data de nascimento. Caso não seja, vamos remover o erro do `input`, através do mesmo método, só que passando uma `string` vazia.

```
if (dataFaz18 > dataAtual) {  
  input.setCustomValidity("A idade mínima é de 18 anos");  
  return;  
}  
  
input.setCustomValidity("");  
return;
```

O código completo da função está logo abaixo.

```
export const validarDataNascimento = input => {  
  const dataNascimento = new Date(input.value);  
  const dataAtual = new Date();  
  
  const dataFaz18 = new Date(  
    dataNascimento.getUTCFullYear() + 18,
```

```
dataNascimento.getUTCMonth(),
dataNascimento.getUTCDate()
);

if (dataFaz18 > dataAtual) {
  input.setCustomValidity("A idade mínima é de 18 anos");
  return;
}

input.setCustomValidity("");
return;
};
```