

## Animando a remoção do paciente

### Transcrição

Ao removermos um paciente, a linha some instantaneamente, abruptamente. Para melhorarmos a experiência do usuário, seria interessante que a linha fosse sumindo gradualmente, até desaparecer totalmente da tabela.

Então vamos realizar essa implementação, animando a remoção do paciente!

Sabemos que, se queremos incluir um efeito visual, devemos fazer as alterações no arquivo CSS. Em `index.css`, adicionaremos a classe `fadeOut`, que definirá a opacidade do elemento com `0`, fazendo-o sumir:

```
.fadeOut {  
  opacity: 0;  
  transition: 0.5s;  
}
```

Para que o processo ocorra de maneira gradual, adicionaremos uma transição de meio segundo, e incluiremos esta classe no paciente utilizando o JavaScript.

Comentaremos a linha que remove o elemento com a função `remove` em `remover-paciente.js`, e depois adicionaremos a classe `fadeOut` ao elemento:

```
var tabela = document.querySelector("table");  
  
tabela.addEventListener("dblclick", function(event) {  
  event.target.parentNode.classList.add("fadeOut");  
  //event.target.parentNode.remove();  
});
```

Vamos testar com um duplo clique no elemento e vê-lo sumindo aos poucos... No entanto, a linha continuará na tabela.



Nome	Peso(kg)	Altura(m)	Gordura Corporal(%)	IMC
Paulo	100	-2.00	10	Altura inválida!
João	80	1.72	40	27.04

Isso ocorre porque só estamos mexendo na parte visual do elemento, o HTML continuará da mesma forma. Se "descomentarmos" a linha, removeremos o elemento logo em seguida à adição da classe:

```
var tabela = document.querySelector("#tabela-pacientes");

tabela.addEventListener("dblclick", function(event) {
  event.target.parentNode.classList.add("fadeOut");
  event.target.parentNode.remove();
});
```

Após o duplo clique em algum elemento da tabela, a linha desaparecerá instantaneamente. Por quê?

Como o computador é muito rápido, o browser processa as linhas do script instantaneamente. A classe é adicionada e logo em seguida o elemento é removido, por isso não conseguimos ver a transição acontecendo. Porém, devemos remover a linha da tabela só após o término da transição. Devemos pedir para o JavaScript aguardar meio segundo (tempo que colocamos para a transição ocorrer) para remover a linha.

Quando queremos aguardar um tempo, devemos usar a função `setTimeout`. Será passada como parâmetro uma função anônima com quanto tempo deve ser aguardado:

```
var tabela = document.querySelector("#tabela-pacientes");

tabela.addEventListener("dblclick", function(event) {
  event.target.parentNode.classList.add("fadeOut");

  setTimeout(function() {
    event.target.parentNode.remove();
  }, 500);
});
```

Com a função acima iremos assegurar a execução da função passada por parâmetro para ela, com o tempo de 500 milissegundos, equivalente a meio segundo.

Fazendo o teste novamente com um duplo clique em uma linha, a transição ocorre e, após meio segundo, a linha é removida.

Espero que você tenha gostado dessa aula! Aprendemos sobre a delegação, processo possível pelo borbulhamento, com *event bubbling*. Independentemente de qual elemento clicarmos, o evento sempre vai subindo para a tag pai. Conhecemos também a propriedade do `event.target`, que possui o `parentNode`. Eles foram usados no nosso código para a remoção de um paciente na nossa tabela.

Faça os exercícios para praticar o que foi feito aqui!