

Buscando dados do paciente

Transcrição

Vamos continuar com o projeto do site da Aparecida! Nós vimos como selecionar um elemento da página com a função `querySelector()`, assim como é possível alterar um texto com o `textContent`. Se selecionarmos um elemento da página, obter e reescrevermos seu conteúdo textual, poderemos realizar o cálculo automático do IMC de cada paciente cadastrado e facilitar a vida da Aparecida.

Para calcularmos o IMC, trabalharemos com valores do **peso** e da **altura** de um paciente da tabela, pois o cálculo é feito desta forma:

$$\text{IMC} = \text{peso} / \text{altura} \times \text{altura};$$

Tendo os dados, precisaremos de uma forma para levá-los ao mundo do JavaScript, calcular o IMC e imprimir o resultado na devida coluna. Esta será nossa missão, vamos começar?

##Extraindo o peso e a altura do paciente

Primeiramente, vamos focar no primeiro paciente, o Paulo. No `index.html`, observaremos os seguintes dados HTML:

```
<tr class="paciente">
  <td class="info-nome">Paulo</td>
  <td class="info-peso">100</td>
  <td class="info-altura">2.00</td>
  <td class="info-gordura">10</td>
  <td class="info-imc">0</td>
</tr>
```

Veremos que cada paciente está contido em uma linha da tabela (`<tr>`) e que em cada coluna (`<td>`) há um dado. Já sabemos como selecionar um paciente, por meio do `querySelector()`:

```
var titulo = document.querySelector(".titulo");
titulo.textContent = "Aparecida Nutricionista";

var paciente = document.querySelector(".paciente");
```

Mas todos os pacientes possuem a classe `paciente`. Se queremos buscar o Paulo, devemos atribuir algo mais específico, para isto, iremos definir um `id` para o primeiro paciente (`primeiro-paciente`):

```
<tr class="paciente" id="primeiro-paciente">
  <td class="info-nome">Paulo</td>
  <td class="info-peso">100</td>
  <td class="info-altura">2.00</td>
  <td class="info-gordura">10</td>
  <td class="info-imc">0</td>
</tr>
```

De volta ao arquivo JavaScript, em `querySelector` utilizaremos o seletor de `id`, a cerquilha (`#`), também conhecida como "tralha" ou "hashtag"):

```
var titulo = document.querySelector(".titulo");
titulo.textContent = "Aparecida Nutricionista";

var paciente = document.querySelector("#primeiro-paciente");

console.log(paciente)
```

Vamos executar o `console.log()` da variável `paciente`, e esperamos conseguir executar a tag `<tr>`, que contém o Paulo. Faremos uma segunda busca no navegador, porém, desta vez vamos buscar em `paciente`, em vez de pesquisarmos no documento inteiro. O retorno será o seguinte:

```
<tr class="paciente" id="primeiro-paciente">
  <td class="info-nome">Paulo</td>
  <td class="info-peso">100</td>
  <td class="info-altura">2.00</td>
  <td class="info-gordura">10</td>
  <td class="info-imc">0</td>
</tr>
```

É retornado um `<tr>`, no entanto, nosso interesse é no peso e na altura do Paulo, que estão dentro de tags `<td>`. Mais especificamente, estamos interessados no conteúdo de texto da `<td>`, e se observarmos o código, veremos que elas podem realizar a busca utilizando as classes `info-peso` e `info-altura`.

Nós já usamos o `querySelector()` para buscar dentro do `document`, será que conseguimos realizá-la também dentro de `paciente`?

```
var titulo = document.querySelector(".titulo");
titulo.textContent = "Aparecida Nutricionista";

var paciente = document.querySelector("#primeiro-paciente");
var tdPeso = paciente.querySelector(".info-peso");

console.log(paciente); // tr
console.log(tdPeso); // td que tem o peso
```

Aqui, buscamos em `paciente` um elemento que tenha a classe `.info-peso`, usando o seletor de classe (`.`). Isso vai nos retornar a `<td>` que contém este dado do paciente. Observe que este dado é salvo na variável `tdPeso`.

No console, veremos o seguinte resultado:

```
<tr class="paciente" id="primeiro-paciente">...</tr>
<td class="info-peso">100</td>
```

Novamente, exibe-se a linha com o `<td>`, e só queremos o conteúdo de texto, ou seja, o valor referente ao peso do paciente. Para termos acesso ao seu conteúdo de texto, adicionaremos o `textContent` no `principal.js`:

```
var paciente = document.querySelector("#primeiro-paciente");
var tdPeso = paciente.querySelector(".info-peso");

var peso = tdPeso.textContent;

console.log(paciente); // tr
console.log(tdPeso); // td que tem o peso
console.log(peso); //Obter 100
```

No console do navegador, será exibido o valor 100 .

```
<tr class="paciente" id="primeiro-paciente">...</tr>
  <td class="info-peso">100</td>
100
```

No entanto, para o cálculo do IMC, além do peso, precisaremos também da altura. Então vamos acessá-la utilizando a mesma lógica e criando a variável `tdAltura` logo abaixo da variável `peso` :

```
var paciente = document.querySelector("#primeiro-paciente");

var tdPeso = paciente.querySelector(".info-peso");
var peso = tdPeso.textContent;

var tdAltura = paciente.querySelector(".info-altura");
var altura = tdAltura.textContent;
```

Fizemos uma busca com o `querySelector()` na variável `paciente` . Nosso objetivo é encontrar a `td` com a classe `.info-altura` . Também criamos a variável `altura` , que é o conteúdo de texto do `td` , e por isso utilizamos o `textContent` .

Para inspecionarmos se o que foi feito deu certo, adicionaremos o `console.log()` para imprimir o `tdAltura` após as variáveis recém adicionadas:

```
var tdAltura = paciente.querySelector(".info-altura");
var altura = tdAltura.textContent;

console.log(paciente); // tr
console.log(tdAltura);
console.log(altura);
```

No console, obteremos:

```
<tr class="paciente" id="primeiro-paciente">...</tr>
  <td class="info-altura">2.00</td>

2.00
```

Veremos o valor 2.00 impresso no console, e obteremos os valores do peso e da altura, então seguiremos para o cálculo do IMC, a partir da seguinte fórmula:

$$\text{IMC} = \text{peso} / \text{altura} \times \text{altura}$$

Isto é, o valor da massa (peso) dividido pela altura multiplicada por ela mesma.

Se a replicarmos no JavaScript utilizando os operadores matemáticos da linguagem de programação, faremos o seguinte cálculo em uma nova variável, chamada de `imc` :

```
var tdAltura = paciente.querySelector(".info-altura");
var altura = tdAltura.textContent;

var imc = peso / (altura * altura);
```

Removeremos o `console.log()` , que se tornou desnecessário no código - observe que usamos o operador `*` (asterisco) na multiplicação.

No caso do paciente Paulo, o resultado do IMC deveria ser:

```
IMC = 100 / (2.00 x 2.00)
IMC = 100 / 4.00
IMC = 25
```

Anotaremos a lógica do cálculo com comentário.

```
var paciente = document.querySelector("#primeiro-paciente");

var tdPeso = paciente.querySelector(".info-peso");
var peso = tdPeso.textContent;

var tdAltura = paciente.querySelector(".info-altura");
var altura = tdAltura.textContent;

var imc = peso / altura * altura; // 100 / 2.0 x 2.0 = 100 / 4 =>>>>>>> 25

console.log(imc);
```

O resultado esperado para a variável `imc` é **25**, porém, o console exibirá `100` como resultado.

Como isso é possível? Será que JavaScript não sabe fazer esta multiplicação básica?

Na verdade, a multiplicação está sendo feita, no entanto precisaremos identificar o problema no código!