

## Validando e exibindo resultado

Aprendemos a obter um elemento do Mundo HTML e trazê-lo para o mundo JavaScript para podermos acessá-lo aplicando as mais diversas lógicas de programação.

Por exemplo, vamos revisar o código que extrai de uma página o peso e a altura para calcular o IMC.

Primeiro, temos um fragmento do HTML:

```
<table>
  <thead>
    <tr>
      <th>Nome</th>
      <th>Peso(kg)</th>
      <th>Altura(m)</th>
      <th>Gordura Corporal(%)</th>
      <th>IMC</th>
    </tr>
  </thead>
  <tbody id="tabela-pacientes">
    <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>
  </tbody>
</table>
```

Segundo, o código JavaScript que manipula esse HTML:

```
var trPaciente = document.querySelector('#primeiro-paciente');
var tdPeso = trPaciente.querySelector('.info-peso');
var tdAltura = trPaciente.querySelector('.info-altura');

var peso = tdPeso.textContent;
var altura = tdAltura.textContent;

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

No entanto, não podemos simplesmente realizar o cálculo às cegas. Precisamos primeiro verificar se o peso e altura trazidos do mundo HTML são válidos. Não queremos magoar ninguém calculando um IMC indevido, certo? Aprendemos em lógica de programação a trabalharmos com a instrução `if` que espera receber `true` ou `false`. Sendo assim, podemos realizar críticas em nosso código da seguinte forma:

```
var trPaciente = document.querySelector('#primeiro-paciente');
var tdPeso = trPaciente.querySelector('.info-peso');
var tdAltura = trPaciente.querySelector('.info-altura');
var tdImc = trPaciente.querySelector('.info-imc');
```

```

var peso = tdPeso.textContent;
var altura = tdAltura.textContent;

var pesoEhValido = true; // assumindo de boa fé que o peso é válido
var alturaEhValida = true; // assumindo de boa fé que a altura é válida

if(peso <=0 || peso >= 1000) {
  pesoEhValido = false;
}

if(altura <= 0 || altura >= 3.00) {
  alturaEhValida = false;
}

if(pesoEhValido && alturaEhValida) {
  var imc = peso / ( altura * altura);
}

```

Apesar do código acima ser funcional, em nenhum momento ele está exibindo o IMC calculado na página, muito menos as mensagens de erro caso peso ou altura sejam inválidos. Qual das opções abaixo exibe o IMC, inclusive as mensagens de erro?



```

if(peso <=0 || peso >= 1000) {
  pesoEhValido = false;
  tdPeso.textContent = 'peso inválido';
}

if(altura <= 0 || altura >= 3.00) {
  alturaEhValida = false;
  tdAltura.textContent = 'altura inválida';
}

if(pesoEhValido && alturaEhValida) {
  var imc = peso / ( altura * altura);
  tdImc.textContent = imc;
}

```

B



```

if(peso <=0 || peso >= 1000) {
  pesoEhValido = false;
  tdPeso.value = 'peso inválido';
}

if(altura <= 0 || altura >= 3.00) {
  alturaEhValida = false;
  tdAltura.value = 'altura inválida';
}

if(pesoEhValido && alturaEhValida) {
  var imc = peso / ( altura * altura);
  tdImc.value = imc;
}

```



```
C if(peso <=0 || peso >= 1000) {  
    pesoEhValido = false;  
    peso.textContent = 'peso inválido';  
}  
  
if(altura <= 0 || altura >= 3.00) {  
    alturaEhValida = false;  
    altura.textContent = 'altura inválida';  
}  
  
if(pesoEhValido && alturaEhValida) {  
    var imc = peso / ( altura * altura);  
    imc.textContent = imc;  
}
```

Veja que já somos capazes de capturar informações na página e ainda executar um monte de críticas para verificar a integridade dos dados antes de executarmos nossa lógica com eles.

PRÓXIMA ATIVIDADE