

JavaScript

Parte 3

Arrays, Loop e Estilos

Exercícios

1 O IMC de todos os pacientes

Por enquanto nosso código só calcula e valida o IMC de **um** único paciente. Vamos atacar este problema neste capítulo para que o IMC de todos os pacientes sejam carregados:

1- Primeiro passo então é deixar de selecionar apenas um paciente e passar a selecionar **todos** os pacientes. Faremos isto buscando por todas as `<tr>` que tem a classe `paciente`. Claro, devemos também utilizar outra função, a função `querySelectorAll()` que é capaz de retornar mais de um elemento quando faz a busca no HTML. Faça a substituição no seu código para trocar o nome da variável `paciente` para `pacientes` e também faça a substituição da função e do seletor:

```
var pacientes = document.querySelectorAll(".paciente");
console.log(pacientes);
```

2- Recarregando a página agora, veremos que o console dará um erro, mas é por que trocamos o nome da variável e ainda não adaptamos nosso código para trabalhar com um *array* de pacientes em vez de um único. Para validar as informações e calcular o IMC de cada paciente do array precisamos **percorrer** este array utilizando um clássico **for**:

```
for(var i = 0; i < pacientes.length; i++){
  var paciente = pacientes[i];
  console.log(paciente);
}
```

3- Agora que conseguimos percorrer a lista de pacientes, basta mover o nosso código que validava e calculava o IMC do paciente para dentro da tabela. O código final deve ficar assim:

```

var pacientes = document.querySelectorAll(".paciente");

for(var i = 0; i < pacientes.length ; i++){
    var paciente = pacientes[i];
    var tdPeso = paciente.querySelector(".info-peso");
    var tdAltura = paciente.querySelector(".info-altura");

    var tdImc = paciente.querySelector(".info-imc");

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

    var alturaEhValida = true;
    var pesoEhValido = true;

    if(peso <= 0 || peso > 1000){
        console.log("Peso inválido");
        tdImc.textContent = "Peso inválido!";
        pesoEhValido = false;
    }

    if(altura <= 0 || altura >= 1000){
        console.log("Altura inválida");
        tdImc.textContent = "Altura inválida!";
        alturaEhValida = false;
    }

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

```

4- Por último, podemos limitar o número de casas decimais do IMC utilizando a função `.toFixed()`. Faça a substituição para imprimir o IMC na tabela com apenas duas casas decimais:

```

// Restante do código

if(pesoEhValido && alturaEhValida){
    var imc = peso / ( altura * altura);
    //ALTERAÇÃO AQUI...
    tdImc.textContent = imc.toFixed(2);
}

```

Nome	Peso(kg)	Altura(m)	Gordura Corporal(%)	IMC
Paulo	100	2.00	10	25.00
João	80	1.72	40	27.04
Erica	54	1.64	14	20.08
Douglas	85	1.73	24	28.40
Tatiana	46	1.55	19	19.15

Agora cada paciente deve estar com IMC sendo calculado!

2 Melhorando a visualização do erro

Vamos melhorar um pouco a visualização de nossa tabela, para que os erros pintem a linha de vermelho além escrever na coluna de IMC quando ocorrer um erro.

1- Vamos trocar a cor da linha colocando uma nova classe na mesma, que será a classe `paciente-invalido`. Como primeiro passo crie uma nova classe no seu arquivo `index.css` como a classe abaixo:

```
.paciente-invalido{  
  background-color: lightcoral;  
}
```

Nossa classe troca o `background-color` do elemento para a cor `lightcoral`, que é um vermelho claro.

2- Agora que temos uma classe com o estilo que queremos, devemos aplicá-la quando acontecer um erro, e os erros são detectados dentro de cada `if`. No `if` de peso, adicione a classe `paciente-invalido` a linha da tabela que isto ocorreu (ou seja, na variável `paciente`):

```
if(peso <= 0 || peso >= 1000){  
  console.log("Peso inválido");  
  tdImc.textContent = "Peso inválido!";  
  pesoEhValido = false;  
  
  paciente.classList.add("paciente-invalido");  
}
```

Estamos alterando o estilo colocando uma classe para facilitar a manutenção de nosso código, afinal se precisarmos alterar a cor, basta alterar o valor da propriedade `background-color` no CSS.

3- Faça a mesma coisa agora para quando a altura for inválida:

```
if(altura <= 0 || altura >= 3.00){  
  console.log("Altura inválida");  
  tdImc.textContent = "Altura inválida!";  
  alturaEhValida = false;  
  
  paciente.classList.add("paciente-invalido");  
}
```

Pronto, agora temos um visual melhor quando ocorrer um erro em nossa tabela!