

{reprograma}

JavaScript

Helena Strada

Sumário

JavaScript.....	3
Formulário	3
Acessando algumas informações da página	3
Selecionando um aluno	4
E a lógica?	6
Organizando nossos arquivos	7
Aplicando a média para todos os alunos	8
Melhorando a visualização dos alunos	11
Escutando eventos	12
Como evitar o comportamento padrão de um evento.....	13
Adicionando um novo aluno na lista.....	13
Separando nosso código.....	15
Criando um objeto.....	16
Validando um formulário.....	18
Retornando uma mensagem de erro.....	20

JavaScript

Apresentação PDF: 'reprograma-js-jquery.pdf'.

Formulário

Nesta primeira etapa, vou começar a trabalhar com js em um formulário de cadastro.

O objetivo deste formulário é: calcular a média com base em duas notas e colocar uma cor de fundo para saber se o aluno possui uma média maior do que a exigida pelo curso.

Acessando algumas informações da página

Antes de calcularmos a média, vamos ver como buscamos algumas informações de nossa página.

```
<script>
  var tituloDaPagina = document.querySelector("h2");
  console.log(tituloDaPagina);
</script>
```

Nós queremos apenas a informação sobre o conteúdo deste título ("Alun@s"). Assim, podemos alterar o que está escrito.

```
onwebkitanimationiteration: null
onwebkitanimationstart: null
onwebkittransitionend: null
onwheel: null
outerHTML: "<h2>Alun@s</h2>"
▶ ownerDocument: HTMLDocument file:///D:/reprograma/5turra
  javascript/index.html
▶ parentElement: <section class="container"> ⚙
▶ parentNode: <section class="container"> ⚙
  prefix: null
  previousElementSibling: null
▶ previousSibling: #text "
  " ⚙
  scrollHeight: 48
  scrollLeft: 0
  scrollLeftMax: 0
  scrollTop: 0
  scrollTopMax: 0
  scrollWidth: 479
  spellcheck: false
▶ style: CSS2Properties { }
  tabIndex: -1
  tagName: "H2"
  textContent: "Alun@s"
  title: ""
```

```
<script>
  var tituloDaPagina = document.querySelector("h2");
  console.log(tituloDaPagina);
  console.log(tituloDaPagina.textContent);

  tituloDaPagina.textContent = "Novo título";

</script>
```

Novo título

Nome	Nota 1	Nota 2	Média Final
Nathalia	5.75	10	0
João	8	10	0
Mariana	9	6	0
Douglas	4	3	0
Tatiana	3	6	0

Adicionar Alun@

Conseguimos manipular qualquer elemento dentro de uma página html.

Selecionando um aluno

Uma vez que conseguimos manipular os elementos dentro de uma página HTML, o próximo passo, sendo objetivo, é que queremos buscar as notas do nosso primeiro aluno da lista e calcular a média.

Porém, observamos que os dados de acesso para os alunos, são os mesmos. Vamos realizar a seguinte tarefa. Vamos colocar um identificador para este aluno e realizarmos os cálculos que desejamos.

```
<tr class="aluno" id="primeiro-da-lista">
  <td class="info-nome">Nathalia</td>
  <td class="info-notaum">5.75</td>
  <td class="info-notadois">10</td>
  <td class="info-media">0</td>
</tr>
```

Existem algumas maneiras de eu buscar um determinado item (por id ou por classe).

```
var aluno = document.querySelector("#primeiro-da-lista");  
var aluno = document.getElementById("primeiro-da-lista");  
console.log(aluno);
```

Posso utilizar o `querySelector` ou até mesmo o `getElementById`, ou até por classe ou tag.

```
var aluno = document.querySelector("#primeiro-da-lista");  
// var aluno = document.getElementById("primeiro-da-lista");  
var tdPrimeiraNota = aluno.querySelector(".info-notaum");  
var primeiraNota = tdPrimeiraNota.textContent;  
  
console.log(primeiraNota);
```

Dessa maneira, conseguimos buscar as notas do aluno que desejamos e realizar o cálculo. Assim como fizemos com a primeira nota, vamos realizar com a segunda nota.

Nome	Nota 1	Nota 2	Média Final
Nathalia	5.75	10	0
João			
Mariana			
Douglas			

Developer Tools - {Reprograma} - Notas Finais - file:///D:/repro
Inspector Console Debugger Style Edit
Filter output
5.75

Buscando os dois valores das notas que foram inseridas.

```
var tdPrimeiraNota = aluno.querySelector(".info-notaum");  
var primeiraNota = tdPrimeiraNota.textContent;  
var tdSegundaNota = aluno.querySelector(".info-notadois");  
var segundaNota = tdSegundaNota.textContent;  
  
console.log(primeiraNota);  
console.log(segundaNota);
```

Nome	Nota 1	Nota 2	Média Final
Nathalia	5.75	10	0
João			
Mariana			
Douglas			
Tatiana			

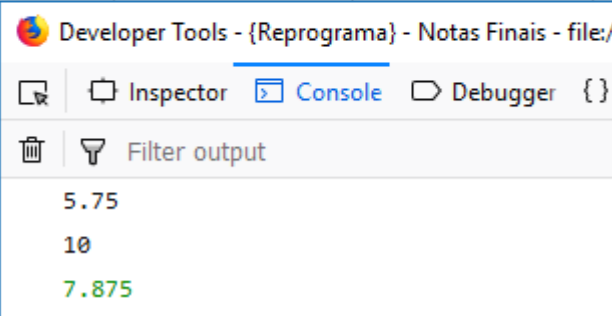
Developer Tools - {Reprograma} - Notas Finais - file:///D:/repro
Inspector Console Debugger Style Edit
Filter output
5.75
10

Uma vez que temos os dados das nossas notas, vamos apenas calcular sua média.

```
var aluno = document.querySelector("#primeiro-da-lista");
var tdPrimeiraNota = aluno.querySelector(".info-notaum");
var primeiraNota = tdPrimeiraNota.textContent;
var tdSegundaNota = aluno.querySelector(".info-notadois");
var segundaNota = tdSegundaNota.textContent;
var mediaFinal = aluno.querySelector(".info-media");

var mediaCalculada = (parseFloat(primeiraNota) + parseFloat(segundaNota)) / 2;
console.log(mediaCalculada);
mediaFinal.textContent = mediaCalculada;
```

Nome	Nota 1	Nota 2	Média Final
Nathalia	5.75	10	7.875
João	8		
Mariana	9		
Douglas	4		
Tatiana	3		



E a lógica?

Vimos até agora, como calcular a média dada duas notas. Mas aonde entra a lógica? Imagina que, por algum motivo, o professor tenha entrado com uma inválida ou até mesmo negativa (inválida). Ainda assim, a média seria calculada? Além disso, uma nota poderia ser maior do que 10 (dez)?

Vamos fazer uma verificação simples: caso a nota seja menor do que zero ou superior a 10, a média final deverá ser mostrada como inválida. Como faríamos?

Vamos alterar o valor da primeira nota para um valor negativo.

```
<tr class="aluno" id="primeiro-da-lista">
  <td class="info-nome">Nathalia</td>
  <td class="info-notaum">-5.75</td>
  <td class="info-notadois">10</td>
  <td class="info-media">0</td>
</tr>
```

Após alterarmos o valor, vamos realizar a nossa lógica.

Primeiro vamos declarar uma variável para saber se o valor inserido está de acordo com os critérios.

```
var notaEhValida = true;
```

Após essa verificação, vamos verificar os valores inseridos até então.

```
var notaEhValida = true;

if (primeiraNota < 0 || primeiraNota > 10) {
  notaEhValida = false;
  mediaFinal.textContent = "Primeira Nota Inválida";
}

if (notaEhValida) {
  var mediaCalculada = (parseFloat(primeiraNota) + parseFloat(segundaNota)) / 2;
  mediaFinal.textContent = mediaCalculada;
}
```

Caso o valor da primeira nota entre em uma das duas primeiras decisões, o valor será considerado inválido.

Nome	Nota 1	Nota 2	Média Final
Nathalia	-5.75	10	Primeira Nota Inválida
João	8	10	9
Mariana	8	10	9
Douglas	8	10	9
Tatiana	8	10	9

Developer Tools - {Reprograma} - Notas Finais - file:///D:/repro...
Inspector Console Debugger Style Edit
Filter output
-5.75
10

Organizando nossos arquivos

Vamos começar a organizar os nossos arquivos do nosso projeto. Vamos criar um novo arquivo chamado index.js (por enquanto) dentro da pasta js.



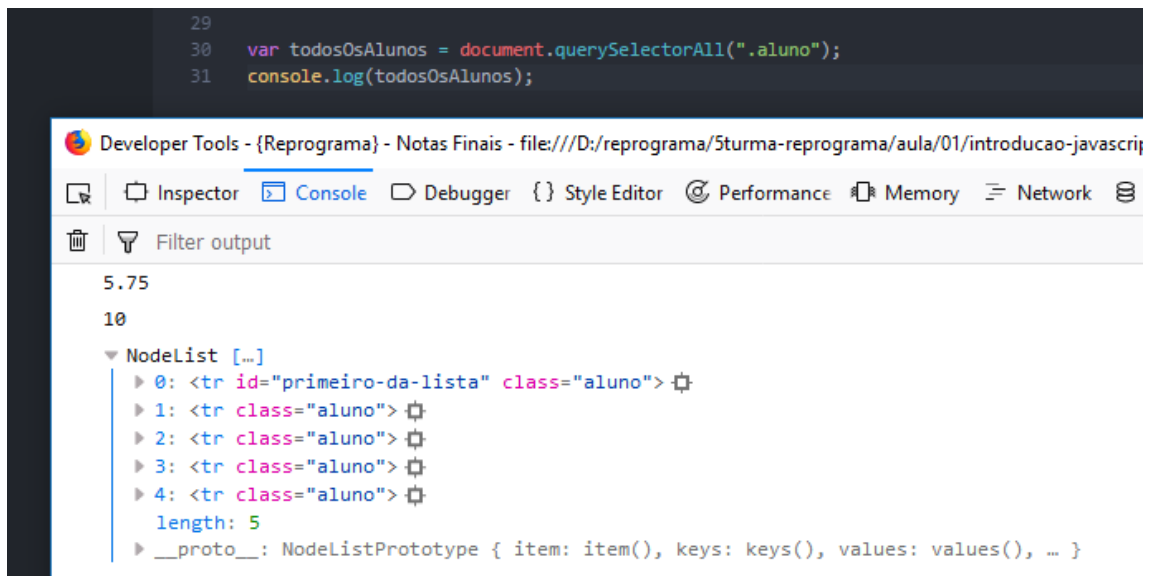
Além disso, vamos colocar o código que escrevemos até agora, dentro do index.js.

```
index_02.html x index.html JS index.js x index.css
1 var tituloDaPagina = document.querySelector("h2");
2 // console.log(tituloDaPagina);
3 // console.log(tituloDaPagina.textContent);
4
5 // tituloDaPagina.textContent = "Novo título";
6
7 var aluno = document.querySelector("#primeiro-da-lista");
8 // var aluno = document.getElementById("primeiro-da-lista");
9 var tdPrimeiraNota = aluno.querySelector(".info-notaum");
10 var primeiraNota = tdPrimeiraNota.textContent;
11 var tdSegundaNota = aluno.querySelector(".info-notadois");
12 var segundaNota = tdSegundaNota.textContent;
13
14 var mediaFinal = aluno.querySelector(".info-media");
15
16 console.log(primeiraNota);
17 console.log(segundaNota);
18
19 var notaEhValida = true;
20
21 if (primeiraNota < 0 || primeiraNota > 10) {
22     notaEhValida = false;
23     mediaFinal.textContent = "Primeira Nota Inválida";
24 }
25
26 if (notaEhValida) {
27     var mediaCalculada = (parseFloat(primeiraNota) + parseFloat(segundaNota)) / 2;
28     mediaFinal.textContent = mediaCalculada;
29 }
30
<script src="./js/index.js" type="text/javascript">
</script>
```

Aplicando a média para todos os alunos

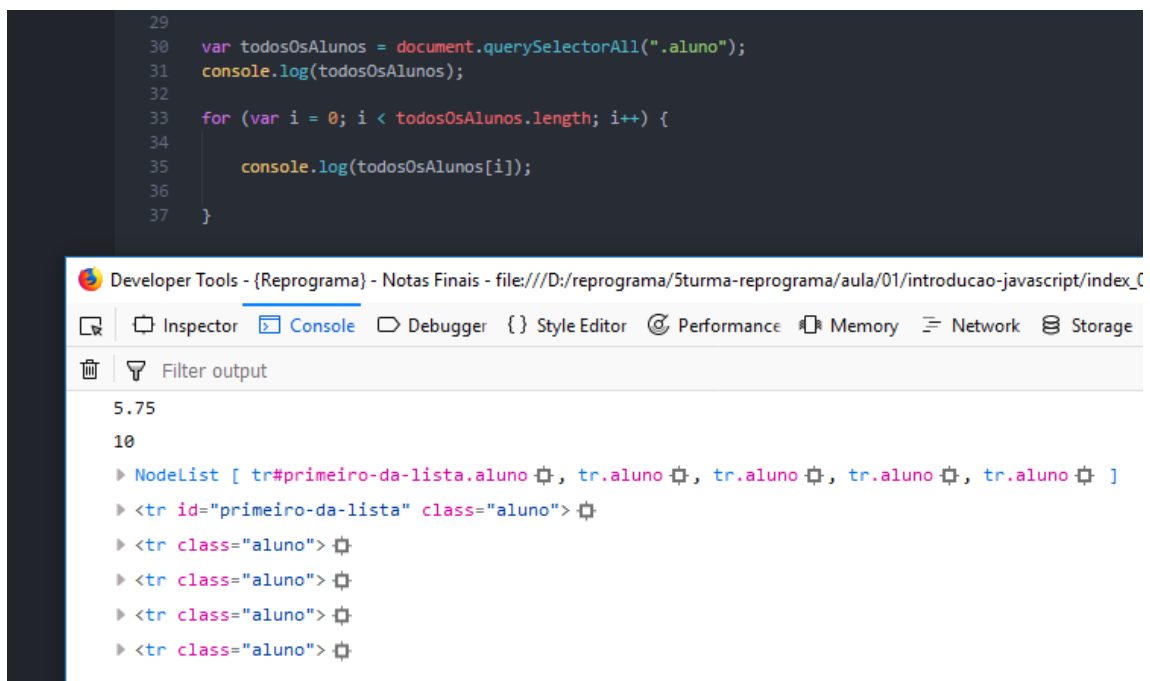
Uma vez que calculamos a média apenas para o primeiro aluno da lista, o que precisamos fazer, é calcular a média para todos os alunos da nossa lista.

O `querySelector`, ele busca apenas uma referência deste item. Já o `querySelectorAll`, nos trará toda a listagem de alunos.



Já temos a nossa lista de alunos. Falta apenas percorrermos essa lista de maneira que: para cada aluno dessa lista, eu consiga realizar o cálculo da média.

Vamos aos passos:



Uma vez que eu consigo percorrer a minha lista de alunos inteira, basta eu copiar o código que nós já fizemos e colocarmos dentro dessa condição. Ou seja, quando a lista for percorrida, ele fará para item da minha lista, sendo assim, para cada que for incluído.

A única diferença será que ao invés de selecionarmos o primeiro aluno pelo id, nós selecionaremos pelo número do índice.

```

var todosOsAlunos = document.querySelectorAll(".aluno");
// console.log(todosOsAlunos);

for (var i = 0; i < todosOsAlunos.length; i++) {

    // console.log(todosOsAlunos[i]);
    var aluno = todosOsAlunos[i];
    // console.log(aluno);

    var tdPrimeiraNota = aluno.querySelector(".info-notaum");
    var primeiraNota = tdPrimeiraNota.textContent;
    var tdSegundaNota = aluno.querySelector(".info-notadois");
    var segundaNota = tdSegundaNota.textContent;
    var mediaFinal = aluno.querySelector(".info-media");

    // console.log(primeiraNota);
    // console.log(segundaNota);

    var notaEhValida = true;

    if (primeiraNota < 0 || primeiraNota > 10) {
        notaEhValida = false;
        mediaFinal.textContent = "Primeira Nota Inválida";
    }

    if (notaEhValida) {
        var mediaCalculada = (parseFloat(primeiraNota) + parseFloat(segundaNota)) / 2;
        mediaFinal.textContent = mediaCalculada;
    }

}

```

Nome	Nota 1	Nota 2	Média Final
Nathalia	5.75	10	7.875
João	8	10	9
Mariana	9	6	7.5
Douglas	4	3	3.5
Tatiana	3	6	4.5

https://www.w3schools.com/js/tryit.asp?filename=tryjs_number_tofixed

Apenas para formatar os números que estamos mostrando em nossa lista, vamos utilizar o toFixed.

```

if (notaEhValida) {
    var mediaCalculada = (parseFloat(primeiraNota) + parseFloat(segundaNota)) / 2;
    mediaFinal.textContent = mediaCalculada.toFixed(2);
}

```

Nome	Nota 1	Nota 2	Média Final
Nathalia	5.75	10	7.88
João	8	10	9.00
Mariana	9	6	7.50
Douglas	4	3	3.50
Tatiana	3	6	4.50

Melhorando a visualização dos alunos

Nome	Nota 1	Nota 2	Média Final
Nathalia	5.75	10	7.88
João	8	10	9.00
Mariana	9	6	7.50
Douglas	4	3	3.50
Tatiana	3	6	4.50

Nós já vimos que sempre que quisermos manipular os elementos dentro de uma árvore de elementos html, nós temos recursos e uma api que faz isso pra nós. Uma vez que queremos alterar o estilo de um determinado item da nossa lista, o que podemos fazer?

Antes de alterarmos o valor correspondente, precisamos realizar uma outra condição simples. Só ficará vermelho caso a média do aluno seja inferior a 5.

```
if (notaEhValida) {
  var mediaCalculada = (parseFloat(primeiraNota) + parseFloat(segundaNota)) / 2;
  if (mediaCalculada < 5) {

  }
  mediaFinal.textContent = mediaCalculada.toFixed(2);
}
```

Caso contrário, a cor permanecerá a mesma.

Google:

Javascript change css style

https://www.w3schools.com/js/js_htmlDOM_css.asp

Nós podemos realizar inicialmente de duas maneiras:

```
if (mediaCalculada < 5) {  
    aluno.style.backgroundColor = "red";  
}
```

Nós podemos adicionar diretamente o estilo em nossa lista para cada item. Porém, imagina que, nós vamos manipular mais de um item por vez, nós teríamos que adicionar muitas propriedades.

Nós podemos criar um estilo em nosso arquivo css, por exemplo:



```
index_02.html JS index.js index.css x  
135 }  
136  
137 .adicionar-aluno {  
138     margin-top: 30px;  
139 }  
140  
141 .campo-invalido {  
142     border: 1px solid red;  
143 }  
144  
145 .abaixo-da-media {  
146     background-color: red;  
147 }
```

Agora, ao invés de adicionarmos diretamente o background em cada linha, adicionamos o estilo como um todo. E caso eu queira adicionar novas propriedades nesse estilo, sua manipulação ficará mais fácil.

```
if (notaEhValida) {  
    var mediaCalculada = (parseFloat(primeiraNota) + parseFloat(segundaNota)) / 2;  
    if (mediaCalculada < 5) {  
        // aluno.style.backgroundColor = "red";  
        aluno.classList.add("abaixo-da-media");  
    }  
    mediaFinal.textContent = mediaCalculada.toFixed(2);  
}
```

Escutando eventos

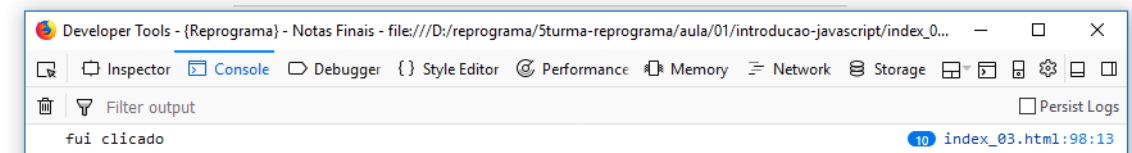
Por quê será que é necessário eu precisar “escutar” um evento da minha página? A ideia é simples. Quando eu clicar no botão para adicionar um novo aluno, eu desejo que alguma ação se realize. Neste caso, eu vou adicionar um novo aluno em nossa tabela.

Mas antes disso, vamos ao básico.

Quando eu quero que algo ocorra quando eu clico, por exemplo, no título da página, eu trabalho com o ‘click’.

```
<script>
  var tituloDaTabela = document.querySelector("h2");
  tituloDaTabela.addEventListener("click", function() {
    console.log("fui clicado");
  });
</script>
```

Alun@s



Fizemos o teste no nosso próprio index.html. E tudo funciona perfeitamente. Vamos então ao que de fato, nós precisamos fazer.

```
var adicionarAluno = document.querySelector("#adicionar-aluno");
adicionarAluno.addEventListener("click", function() {
  console.log("btn adicionar");
});
```

Antes de continuarmos, testem o que ocorre quando eu preencho os dados do formulário e cliço no botão 'Adicionar'.

Como evitar o comportamento padrão de um evento

```
event.preventDefault();
```

```
var adicionarAluno = document.querySelector("#adicionar-aluno");
adicionarAluno.addEventListener("click", function(event) {
  event.preventDefault();
  console.log("btn adicionar");
});
```

Testem novamente o que ocorre agora quando o botão é clicado.

Adicionando um novo aluno na lista

<https://developer.mozilla.org/pt-BR/docs/Traversing an HTML table with JavaScript and DOM Interfaces>

Através do formulário, nós temos acesso aos seus *inputs*. Através da propriedade *name* que colocamos em nosso HTML, nós podemos chama-los diretamente.

```

var adicionarAluno = document.querySelector("#adicionar-aluno");
adicionarAluno.addEventListener("click", function(event) {
    event.preventDefault();
    console.log("btn adicionar");

    var form = document.querySelector("#form-adiciona");

});

```

Como nós queremos o valor de entrada do campo input:

```

var form = document.querySelector("#form-adiciona");

var nome = form.nome.value;

```

Vamos buscar os 3 dados que desejamos de entrada.

```

var nome = form.nome.value;
var notaum = form.notaum.value;
var notadois = form.notadois.value;

```

Uma vez que pegamos os dados de entrada do usuário, nós precisamos criar um novo elemento HTML com JavaScript, ou seja, pegar os dados do usuário e inserir mais uma linha em nossa tabela. Como faremos isso?

```

var form = document.querySelector("#form-adiciona");
var nome = form.nome.value;
var notaum = form.notaum.value;
var notadois = form.notadois.value;

var alunoTr = document.createElement("tr");

var nomeTd = document.createElement("td");
var notaUmTd = document.createElement("td");
var notaDoisTd = document.createElement("td");
var mediaTd = document.createElement("td");

nomeTd.textContent = nome;
notaUmTd.textContent = notaum;
notaDoisTd.textContent = notadois;

alunoTr.appendChild(nomeTd);
alunoTr.appendChild(notaUmTd);
alunoTr.appendChild(notaDoisTd);

var tabela = document.querySelector("#tabela-alunos");
tabela.appendChild(alunoTr);

```

Mariana	9	6	7.50
Douglas	4	3	3.50
Tatiana	3	6	4.50
Helena	10	10	

Separando nosso código

Um dos problemas que nós encaramos agora é que ao adicionarmos um novo aluno, nós precisamos calcular sua média. Seria interessante eu repetir o código no formulário e na lista uma vez que a função que ele irá executar será a mesma?

Sendo assim, vamos criar uma função que faça isso pra nós. O que nós precisamos saber sobre a média para que a função calculaMedia() seja executada?

```
var todosOsAlunos = document.querySelectorAll(".aluno");

for (var i = 0; i < todosOsAlunos.length; i++) {
    var aluno = todosOsAlunos[i];

    var tdPrimeiraNota = aluno.querySelector(".info-notaum");
    var primeiraNota = tdPrimeiraNota.textContent;
    var tdSegundaNota = aluno.querySelector(".info-notadois");
    var segundaNota = tdSegundaNota.textContent;
    var mediaFinal = aluno.querySelector(".info-media");

    var notaEhValida = true;

    if (primeiraNota < 0 || primeiraNota > 10) {
        notaEhValida = false;
        mediaFinal.textContent = "Primeira Nota Inválida";
    }

    if (notaEhValida) {
        // var mediaCalculada = (parseFloat(primeiraNota) + parseFloat(segundaNota)) / 2;
        var mediaCalculada = calculaMedia(primeiraNota, segundaNota);
        if (mediaCalculada < 5) {
            aluno.classList.add("abaixo-da-media");
        }
        mediaFinal.textContent = mediaCalculada.toFixed(2);
    }
}

function calculaMedia(provaUm, provaDois) {
    var mediaDentroDaFuncao = 0;
    mediaDentroDaFuncao = (parseFloat(provaUm) + parseFloat(provaDois)) / 2;
    return mediaDentroDaFuncao;
}
```

Aqui, estamos trabalhando com a nossa lista. Uma vez que desejamos realizar o mesmo cálculo ao inserir um novo aluno, essa função também pode ser chamada.

```
var nomeTd = document.createElement("td");
var notaUmTd = document.createElement("td");
var notaDoisTd = document.createElement("td");
var mediaTd = document.createElement("td");

nomeTd.textContent = nome;
notaUmTd.textContent = notaum;
notaDoisTd.textContent = notadois;
mediaTd.textContent = calculaMedia(notaum, notadois);

alunoTr.appendChild(nomeTd);
alunoTr.appendChild(notaUmTd);
alunoTr.appendChild(notaDoisTd);
alunoTr.appendChild(mediaTd);

var tabela = document.querySelector("#tabela-alunos");
tabela.appendChild(alunoTr);
```

Sendo assim, ao inserir um novo aluno, eu consigo calcular a sua devida média.

Nome	Nota 1	Nota 2	Média Final
Nathalia	5.75	10	7.88
João	8	10	9.00
Mariana	9	6	7.50
Douglas	4	3	3.50
Tatiana	3	6	4.50
Helena	10	10	10

Criando um objeto

Vamos passar os dados que recebermos do formulário para uma função e, além disso, vamos criar um objeto do tipo aluno.


```

var adicionarAluno = document.querySelector("#adicionar-aluno");
adicionarAluno.addEventListener("click", function(event) {
    event.preventDefault();
    console.log("btn adicionar");

    var form = document.querySelector("#form-adiciona");

    // funcao para obter os dados do formulario
    var aluno = obterDadosDoForm(form);
    console.log(aluno);

    var alunoTr = document.createElement("tr");
    var nomeTd = document.createElement("td");
    var notaUmTd = document.createElement("td");
    var notaDoisTd = document.createElement("td");
    var mediaTd = document.createElement("td");

    nomeTd.textContent = aluno.nome;
    notaUmTd.textContent = aluno.notaum;
    notaDoisTd.textContent = aluno.notadois;
    // mediaTd.textContent = calculaMedia(nota1, nota2);
    mediaTd.textContent = aluno.media;

    alunoTr.appendChild(nomeTd);
    alunoTr.appendChild(notaUmTd);
    alunoTr.appendChild(notaDoisTd);
    alunoTr.appendChild(mediaTd);

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

    tabela.appendChild(alunoTr);

});

function obterDadosDoForm(form) {
    var alunoRetorno = {
        nome: form.nome.value,
        notaum: form.notaum.value,
        notadois: form.notadois.value,
        media: calculaMedia(form.notaum.value, form.notadois.value)
    }
    console.log(alunoRetorno);
    return alunoRetorno;
}

```

Alun@s

Nome	Nota 1	Nota 2	Média Final
Nathalia	5.75	10	7.88
João	8	10	9.00
Mariana	9	6	7.50
Douglas	4	3	3.50
Tatiana	3	6	4.50
Helena	10	10	10

Adicionar Alun@

Nome:	Nota 1:	Nota 2:
<input type="text" value="Helena"/>	<input type="text" value="10"/>	<input type="text" value="10"/>

Adicionar

Validando um formulário

Uma vez que conseguimos buscar os dados de nosso formulário, não seria interessante, antes de colocarmos estes dados em uma lista, validarmos os dados de entrada do usuário e inclusive mostrar uma mensagem de erro para o mesmo?

Faremos isso nos passos a seguir.

```
function validaNota(aluno) {  
  if (aluno.notaum > 0 && aluno.notaum <= 10) {  
    return true;  
  }  
}
```

Vamos criar uma função que valida os dados de entrada do usuário.

```

var adicionarAluno = document.querySelector("#adicionar-aluno");
adicionarAluno.addEventListener("click", function(event) {
    event.preventDefault();

    var form = document.querySelector("#form-adiciona");
    var aluno = obterDadosDoForm(form);

    var alunoTr = document.createElement("tr");

    if (!validaNota(aluno)) {
        console.log("nota inválida");
        return;
    }

    var nomeTd = document.createElement("td");
    var notaUmTd = document.createElement("td");
    var notaDoisTd = document.createElement("td");
    var mediaTd = document.createElement("td");

    nomeTd.textContent = aluno.nome;
    notaUmTd.textContent = aluno.notaum;
    notaDoisTd.textContent = aluno.notadois;
    mediaTd.textContent = aluno.media;

    alunoTr.appendChild(nomeTd);
    alunoTr.appendChild(notaUmTd);
    alunoTr.appendChild(notaDoisTd);
    alunoTr.appendChild(mediaTd);

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

    tabela.appendChild(alunoTr);

});

function obterDadosDoForm(form) {
    var alunoRetorno = {
        nome: form.nome.value,
        notaum: form.notaum.value,
        notadois: form.notadois.value,
        media: calculaMedia(form.notaum.value, form.notadois.value)
    }
    console.log(alunoRetorno);
    return alunoRetorno;
}

function validaNota(aluno) {
    if (aluno.notaum > 0 && aluno.notaum <= 10) {
        return true;
    }
}

```

Sempre que eu precisar realizar uma chamada, eu chamo a própria função para realizar o cálculo. Sendo assim, ele só vai inserir um novo item na lista, caso os dados sejam digitados de maneira correta.

Retornando uma mensagem de erro

Vamos adicionar uma mensagem de erro em nosso html.

```
<h2 id="titulo-form">Adicionar Aluno</h2>
<span id="mensagem-erro"></span>
<form id="form-adiciona">
```

```
if (!validaNota(aluno)) {
  var erro = document.querySelector("#mensagem-erro");
  return erro.innerHTML = "A Nota 1 está incorreta.";
}
```

<https://stackoverflow.com/questions/21311299/nodevalue-vs-innerhtml-and-textcontent-how-to-choose>