**Conhecendo o DOM e manipulando páginas**

Até agora, nós apenas criamos variáveis e atribuímos sempre o mesmo valor a elas, por exemplo, o código var peso = 55.7 sempre criará uma variável com valor 55.7, mas qual a utilidade de uma variável se ela não variar, certo?

A ideia então é que agora nós comecemos a utilizar os valores da nossa página, ou seja, criar variáveis com base nos valores do nosso html, mas no mundo do Javascript, para pegar alguma informação de uma página, temos que, primeiro, entender o que é o tal do DOM.

**O que é DOM e como usá-lo?**

DOM é um acrônimo para *Document Object Model*, basicamente ele é uma representação de uma página html no mundo Javascript e pode ser acessado a partir da variável document, que representa a raiz de nossa página.

Utilizando esta variável, conseguimos, por exemplo, pegar o conteúdo de uma tag html. O que eu quero dizer com isso? Considere a seguinte página html:

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="utf-8">

<title>Javascript, a linguagem</title>

</head>

<body>

<h1 id="titulo">Olá, mundo!</h1>

</body>

</html>

Caso queiramos pegar o conteúdo da tag <h1> que possui o atributo id="titulo", podemos pedir ao documentpara selecionar um elemento que o id seja titulo:

var titulo = document.getElementById("titulo"); // Seleciona o titulo

É importante notar que, caso não exista nenhuma tag com o atributo id="titulo" em nossa página, o conteúdo da variável será undefined. Com o elemento em mãos, podemos pegar o conteúdo dele do seguinte modo:

var titulo = document.getElementById("titulo"); // Seleciona elemento

var conteudo = titulo.textContent; // Pega texto dentro do elemento

console.log(conteudo); // Imprime "Olá, mundo!"

Pronto! Agora já conseguimos pegar o conteúdo de uma tag html e colocar em uma variável, então agora vamos pensar na nossa aplicação do curso, a lista de pacientes.

O que queremos fazer é conseguir pegar a variável altura e peso de nosso primeiro paciente na nossa tabela, para isso vamos analisar a estrutura dela:

<table>

<tr>

<th>Nome</th>

<th>Peso</th>

<th>Altura</th>

<th>IMC</th>

</tr>

<tr>

<td id="nome-1">Leonardo</td>

<td id="peso-1">50.0</td>

<td id="altura-1">1.60</td>

<td id="imc-1"></td>

</tr>

<tr>

<td id="nome-2">Paulo</td>

<td id="peso-2">100</td>

<td id="altura-2">2.00</td>

<td id="imc-2"></td>

</tr>

</table>

Vamos montar um paciente com as informações do primeiro da tabela, ou seja, o "Leonardo". Para isso, basta utilizar o comando que acabamos de aprender, o document.getElementById("id-para-selecionar"):

var peso = document.getElementById("peso-1").textContent; // pega o peso do paciente

var altura = document.getElementById("altura-1").textContent; // pega a altura do paciente

**Modificando sua página**

Agora que nós já conseguimos declarar variáveis com valores que estão em nossas páginas, podemos calcular o IMC a partir desses valores aplicando o nosso código original:

var peso = document.getElementById("peso-1").textContent; // pega o peso do paciente

var altura = document.getElementById("altura-1").textContent; // pega a altura do paciente

if(altura != 0){

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

console.log(imc) // 25

}else{

console("Não posso executar uma divisão por 0!");

}

Bacana! Já calculamos o IMC, mas ainda temos uma coluna vazia na nossa tabela, precisamos preenchê-la com o valor calculado!

Para isso, vamos começar selecionando o <td> que queremos preencher:

var tdDoImc = document.getElementById("imc-1"); // pega o td do imc do paciente 1

Lembra que para pegarmos o conteúdo de uma tag nós utilizamos o textContent? Para modificar o seu conteúdo também usaremos ele, mas atribuindo um valor, como se fosse uma variável!

var tdDoImc = document.getElementById("imc-1"); // pega o td do imc do paciente 1

tdDoImc.textContent = imc; //Altera o conteúdo do td para o valor da variável imc

Prontinho! Então o nosso código final será parecido com o seguinte:

var peso = document.getElementById("peso-1").textContent; // pega o peso do paciente

var altura = document.getElementById("altura-1").textContent; // pega a altura do paciente

if(altura != 0){

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

var tdDoImc = document.getElementById("imc-1"); // pega o td do imc do paciente 1

tdDoImc.textContent = imc; //Altera o conteúdo do td para o valor da variável imc

console.log(imc) // 25

}else{

console("Não posso executar uma divisão por 0!");

}

Agora você já sabe como selecionar elementos do DOM e modificá-los! Durante o curso você aprenderá mais modos de selecionar e mais propriedades que poderá alterar, fique ligado!

**Organizando nosso código com objetos**

Agora nós já conseguimos calcular o IMC de um paciente, mas para isso nós usamos duas variáveis separadas: aaltura e o peso.

Seria legal se tivéssemos um modo de juntar essas duas variáveis para que ficasse claro que elas pertencem a umpaciente. Para isso, vamos aprender como declaramos objetos em Javascript!

**A notação de objetos do Javascript**

Um objeto em Javascript, nada mais é que uma variável que possui, dentro dela, chaves e valores. O que queremos fazer é um objeto chamado paciente que terá as chaves (ou atributos) altura e peso.

Para criarmos esse objeto, utilizaremos os caracteres { para abrir o objeto e } para fechá-lo.

var paciente = {};

Simplesmente assim nós já criamos o nosso objeto paciente, porém, ele ainda não tem nenhum atributo. Nós podemos, então, dentro das chaves, adicionarmos o atributo peso com seu valor separado por dois pontos(:):

var paciente = { peso : 100 };

Agora nós temos, dentro de nosso objeto paciente, a propriedade peso, que pode ser acessada a qualquer momento utilizando a seguinte sintaxe:

paciente.peso

Repare que utilizamos um ponto(.) para dizermos ao interpretador de Javascript que queremos acessar um atributo desse objeto, essa notação é chamada de *dot notation*. Outra forma menos usada de acessar um atributo sería a partir de colchetes com o nome do atributo:

paciente["peso"];

Agora só falta adicionarmos o ultimo atributo, a altura. Para isso, basta adicioná-lo após o peso separado por uma vírgula:

var paciente = {peso : 100, altura : 2.00}

Pronto! Temos um paciente completo, com peso e altura! Mas ele terá sempre o mesmo valor de peso e de altura, seria legal se nós pegássemos essa informação da página, como fazíamos antes:

var peso = document.getElementById("peso-1").textContent; // pega o peso do paciente

var altura = document.getElementById("altura-1").textContent; // pega a altura do paciente

var paciente = {

peso : peso,

altura : altura,

}

Note que quebramos a linha para cada atributo por questões de legibilidade, essa é uma notação bem usada quando o objeto tem muitos atributos.

Mas ainda está faltando um atributo no nosso paciente: o **nome**. Agora o paciente já está criado, mas conseguimos adicionar novos atributos mesmo assim utilizando o *dot notation* e atribuindo um valor a ele:

var peso = document.getElementById("peso-1").textContent; // pega o peso do paciente

var altura = document.getElementById("altura-1").textContent; // pega a altura do paciente

var paciente = {

nome:nome //Para poder chamar no comando abaixo

peso : peso,

altura : altura,

}

paciente.nome = document.getElementById("nome-1").textContent; // pega nome do paciente

Agora que organizamos mais o nosso código juntando as informações do paciente, vamos ver como ficaria o nosso cálculo de IMC utilizando esse novo objeto:

var peso = document.getElementById("peso-1").textContent; // pega o peso do paciente

var altura = document.getElementById("altura-1").textContent; // pega a altura do paciente

var nome = document.getElementById("nome-1").textContent; // pega o nome do paciente

var paciente = {

peso : peso,

altura : altura,

nome : nome

}

if(paciente.altura != 0){

var imc = paciente.peso / (paciente.altura \* paciente.altura);

var tdDoImc = document.getElementById("imc-1"); // pega o td do imc do paciente 1

tdDoImc.textContent = imc; //Altera o conteúdo do td para o valor da variável imc

console.log(imc);

}else{

console("Não posso executar uma divisão por 0!");

}

**Para saber mais**

Repare que os objetos em Javascript são um pouco diferentes do que você pode estar acostumado em outras linguagens orientadas a objetos como o Java. Quando estamos utilizando Javascript, nós não precisamos declarar uma classe para, a partir desta, instanciar um objeto, basta criar um objeto diretamente.